

Evolution of a private sector spyware for state actors

FinFisher FinSpy for Android 2012-2019

Thorsten Schröder, Linus Neumann

Chaos Computer Club

28. Dezember 2019 (v1.0)

Abstract

The Chaos Computer Club analyzed several malware samples regarding a) origin and b) creation date.

Within the scope of this report, a total of 28 copies ("samples") of spyware from the years 2012 to 2019 were examined. The result of the analysis is that these samples can be attributed to the same manufacturer:

- A) All samples use the same proprietary mechanism for the provision of malware by the end customer: The case-specific configuration is hidden via the identical covert channel in the Android APK.
- B) All configurations that can be extracted from A) are in a binary format whose structures have the same patterns among each other.
- C) Besides the identical provisioning mechanism of the samples over all generations there are heavy similarities in the Java program code between the samples from 2014 and 2016.
- D) One sample from the year 2012 and several samples from the year 2014 can be clearly assigned to the company Gamma International Deutschland or FinFisher.
- E) Due to the syntax and choice of variable and function names in the Java program code it can be assumed that the software was produced by German-speaking developers.
- F) Based on metadata in the shared-object files of a sample submitted by the Gesellschaft für Freiheitsrechte, it can be proven beyond doubt that this sample was produced in 2016 at the earliest.

Table of Contents

Abstract	2
Table of Contents	3
Introduction	4
Object of investigation.....	4
Central questions.....	5
Method.....	7
A. Determination of the production date	7
1. Versions of software and libraries used during creation	7
2. Timestamps in certificates	7
3. Timestamps in configurations and log files	8
4. Public documentation	8
B. Determination of the origin.....	8
1. Certificates used.....	8
2. Conformity of proprietary routines.....	8
3. Naming of functions and variables.....	10
4. References to samples of known origin	10
Results	13
A. Production time of the "adalet" sample.....	13
Versions of software and libraries used in the creation	13
Timestamps in Certificates.....	14
Timestamps in configurations and log files.....	15
B. Sample origin	16
Used Certificates	16
Matching proprietary routines.....	18
References to samples of known origin	28
Conclusion	32
A. Determination of the date of creation	32
1) When was the "adalet" sample produced and used?.....	32
2) Is the date or period [of creation] before or after 18 July 2015?	32
B. Determination of origin	32
1) Do the samples derive from different sources, or is there clear indication of co-authorship?	32
2) Can the authors of these samples be identified?	32
Public documentation of test objects and methods.....	33
Appendix	34
A. Publication date and time of the SQLite-Version 3.13.0	34
B. Configuration of all samples examined in this analysis	37
C. Sample 421and: ControlFlow com.android.services.CallLogs.run()	59
D. Sample adalet: ControlFlow org.customer.fu.e.a.run()	60

Introduction

Per request of the Gesellschaft für Freiheitsrechte¹, the Chaos Computer Club examined 28 different software applications ("samples") for the Android operating system. The reason for the investigation is the suspicion of violation of export control regulations according to § 18 para. 2 no. 1 and para. 5 no. 1 of the Foreign Trade and Payments Act by FinFisher GmbH, FinFisher Labs GmbH and Elaman GmbH ("FinFisher²"). The GFF has filed a criminal complaint in this context.³

Of particular importance in this context is a sample which, according to GFF, was used in Turkey in 2017 against political opponents. The analysis at hand proves that this malware sample was created in 2016 at the earliest. Investigation further could prove that this sample originated from the "FinSpy" malware family.

Object of investigation

In the following, the term "sample" refers to a specific malware package. Each sample can be referenced by a unique and singular SHA256 checksum. This report only considers Android variants of the malware samples. The starting point of the investigation was initially a malware sample with the checksum `c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e`. This malware was offered for download from the `adaleticinyuru.com` domain according to information provided by the GFF. This website has imitated a website used by the Turkish opposition movement for organizational purposes; presumably in order to attract and deceive users. The sample will be referred to as the "adalet" sample in the following.

This study focuses on the similarities and differences between the "adalet" sample and other samples of known, suspected or unknown origin. Samples investigated date from 2012-2019. CCC currently has access to 26 samples of malware from public and non-public sources that show structural similarities to the "adalet" sample and which have been examined in the context of this analysis. An additional sample was used to install another malware, which also resembles the samples already described. In total, 28 samples were analyzed.

¹ The Society for Freedom Rights (GFF) is a non-profit association based in Berlin, Germany which was founded in 2015. It pursues the goal of achieving the preservation and development of basic and human rights through strategic litigation."

https://de.wikipedia.org/wiki/Gesellschaft_für_Freiheitsrechte accessed 19 December 2019

² The FinFisher / FinSpy malware family is - presumably to disguise business practices - marketed under different names by different limited liability companies and international companies, with a high degree of overlap in terms of personnel and location. To simplify matters, the authors use the term "FinFisher Group" in the following to designate this conglomerate.

³ GFF: „Export von Überwachungssoftware“ <https://freiheitsrechte.org/export-von-uberwachungsssoftware/> accessed 19 December 2019

Central questions

The following questions were the basis of the analysis:

A) Production period

- 1) When were the samples produced and used?
- 2) Is this date or period before or after July 18 2015⁴?

B) Origin

- 1) Do the samples derive from different sources, or is there clear indication of co-authorship?
- 2) Is it possible to identify the authors of the samples?

For this purpose, similarities and differences of the individual samples have been analyzed and documented over a period of seven years. Table 1 shows the Android samples included in the study. Samples that can be attributed without any doubt to the FinFisher group⁵ are highlighted in color. The "adalet" sample, which forms the main focus of this study, is marked in green.

⁴ Date of incorporation of so-called intrusion software in the Foreign Trade and Payments Ordinance (Außenwirtschaftsverordnung, AWV), in order to implement the requirements of the EU Dual-Use Regulation in national law as of 1 January 2015. Since this date, intrusion software that is manufactured in the EU and sold outside the EU is subject to approval.

⁵ The samples in question originate from a leak in 2014. See paragraph 4. References to samples of known origin, page 11

Table 1 - Overview of the samples examined in this analysis

SHA256	TargetID
2e96e343ac10f5d9ace680e456c083e4eceb108f7209aa1e849f11a239e7a682	again
0d798ca0b2d0ea9bad251125973d8800ad3043e51d4cc6d0d57b971a97d3af2d	JHANUK
72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537	Andriod
363172a2f2b228c7b00b614178e4ffa00a3a124200ceef4e6d7edb25a4696345	derise
1935f2e52832df910edc1b7ef17b53d8c852fe66ec3afbe490ffc2ef057452b3	AKDemo
045161094b9f6b98c4ef87d2324f4bb8a0d0fbf58e349a37d11622aef2e6b051	ANDDemo
84d39e5c6db75801a85cc5d2557ab536abe40496b23eba6b3e5c1722975d8f32	428
587b110da2ef9c59b18f01e97e9b12628f3e7b2e88611f7cb28a6efccb0aaba2	tmWoot
abcb11c4787c62ab90cecc262f6fc98bd7f73335ac631c2e9aec8734088e91aa	ANDR
2795e777d897857ab6fa19f85687a23a6071ab665299a47b8b952cb4e7056a07	Android
704d599fe51e7a0f982438c13983fb936dd1530f659e8036bee69752221ef7d7	ANDxJoe
26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafba1	421and
1507ee069906d2a42216d77ef51d42a35efcc59b005b55d8ea771749057296db	trekki
1ea335d1d5f99aebela516d6b267ba53c38438648874752eb0438edfffde380d	zefix
60dc08ab28db5ba3a56734097954861a525b4b384e8067e3eaac551c4cb8ece3	testAD
84d231e6ea1e2e3283c3e9cbfcabeded0d7e5723852e378e0caf5bb001501938	defs
46690ef267f21b5840377833e7af51b19fbd343bac97d6eb66b186d58ba3f9b3	flash28
23f154723213452634abe6063fd07bd3a38700a6b0ba4117db3224ae1411dada	flash28
c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e	adalet
77b4d11e369ac5dec4e951e5879248c1c9a84d756c06d89875f113e4c6469464	cleaner
31fa1129d8e682a90913cc28b4e5d6b064131c93a6d86118d94f93918ed6e2f8	whistel
49c12654aaee1b089268931307f36a5d0d020325226328f780dc152b2f04b281	\$container\$
269227c4c4770e109e53c6cf87bd9bde367843c4806f5975c5aa317f318e28a9	PyawApp
241c38fd3cafc37f496fb7e1872924f21bf1263e17a81d03981dd29b531e4623	network
d8f6abc6cb1388da6b2870f06d52036a435407d6bf2c0b43684fd72edc4a9e77	Disk
aa299745edf2e55531c9a8304b57f9bee8f37a4c3f4be56260bad096c7ea1c03	FunVoic
3f8baeae01980e77fa905216e291b6478105295c8372a003d73e9086b0b3e964	Diary
ff8aaf49f4377e6ee162f1f0778f98e33dd2a8df2d96de6ba766851ee436467e	myphone

Method

The following chapter explains the investigation methods used and their different significance or evidentiary value.

A. Determination of the production date

We decompiled⁶ the samples we had available and examined them for indications that would allow us to conclude time or period of compilation. When compiling, the program is completed, bundled and made ready for use. The time of compilation therefore marks the *creation or completion of the sample* and thus its time of production. All programming work must have been completed before that date. Changes to the program code will require the software to be recompiled.

Indications for the manufacturing time or period could be obtained in particular from the following attributes:

1. Versions of software and libraries used during creation

Libraries are additional modules that provide frequently used functionalities. Programs invoke libraries to be able to use the features provided by these libraries in the program without having to reprogram them. Thus, one does not need to "reinvent the wheel".

The version number of a library or a program used to build the library can be an indication for the point in time after which a sample has been produced. Although the use of older software and libraries is possible, the use of software or libraries from the future is not.

2. Timestamps in certificates

All Android applications must be digitally signed by the developer⁷. The signature secures authenticity and integrity (freedom from subsequent changes) of the application through cryptographic operations. Any, even minimal, changes to the application will result in a signature verification failure so that the application is no longer executable. This measure serves to protect against any subsequent manipulation of the application by third parties.

The signature is created using so-called certificates, which consist of a public and a private part⁸. The public part is included in the application and is used to check the signature. The private part of the certificate remains with the manufacturer and is used to create the signature.

The certificates used to sign the application have a validity period whose beginning and duration can be chosen at will. The start of the validity period of the certificate is a comparatively weak indicator for the first time the application is used because the operating system Android can under certain conditions also run software that was signed with certificates that will be valid only in the future.

⁶ Decompiling means the reverse translation of machine or byte code into human-readable program code. Android applications, like these samples, are a compressed archive in a standardized structure which contains the different parts of the application, e.g. program code, images, or data. By "decompiling", this archive will be restored as a folder structure, and a more in-depth analysis of the sample is possible.

⁷ "Android requires that all APKs be digitally signed with a certificate before they are installed on a device or updated." <https://developer.android.com/studio/publish/app-signing>

⁸ For a detailed description of the concept, see https://en.wikipedia.org/wiki/Digital_signature

3. Timestamps in configurations and log files

The compilation timestamp is noted in various files as part of the application. These *timestamps* also have evidence character since they could be faked with a manageable amount of effort, although the reason or motivation for this would be questionable. This is especially true concerning the objective of the investigation, which is to determine the *earliest* time of creation. To set a wrong track for this determination, the application would have to be manipulated "into the future" during its compilation, which in turn would be conspicuous until this day in the future.

4. Public documentation

The earliest publicly documented time of deployment can be researched using publicly available information on virus analysis platforms such as *VirusTotal*⁹ or by the *Wayback Machine*¹⁰ from public sources. If this time is later than a given date, only the later use of the software, but not the time of its manufacturing, is proven beyond doubt.

B. Determination of the origin

Due to economic interests and possible legal consequences it can be assumed that the manufacturer has little motivation to assert or admit his authorship of malware publicly without necessity.

The analysis therefore follows a *bottom-up* approach, systematically examining similarities of different samples: Indications for the origin and authorship of the malware must be derived from similarities and differences to other samples whose origin is known beyond doubt. The alternative explanations of a copy or intentional deception by third parties must always be considered. Only when this is sufficiently implausible can it be assumed to be circumstantial or conclusive.

Thus, if the "adalet" sample in question bears a clear similarity to a sample of clear origin which cannot be explained by chance, that must be regarded as a strong indication that the "adalet" sample is also attributable to the identified author.

In the context of the present analysis, indications of origin and authorship could be obtained in particular through the following attributes:

1. Certificates used

The meaning of code signing certificates is explained in paragraph A-2. Timestamps in Certificates on page 7 explained. These certificates are also of particular importance when determining the origin of a sample.

Due to cryptographic protection, the construction, i.e. forgery or "replication" of a modern certificate would be impossible as of today or only possible with enormous amounts of time, energy and resources. If the private part of the certificate has not been passed on, stolen or published, applications signed with the same certificate come from the same source¹¹.

2. Conformity of proprietary routines

⁹ *VirusTotal* is a free online service operated by Google Inc. to have individual files analyzed by over 70 different anti-virus programs and malware scanners. The service is available at <https://www.virustotal.com/> and is probably one of the largest analysis platforms for malware. For each analyzed file, the date on which it was first analyzed is recorded.

¹⁰ The *Wayback Machine* is a project of the non-profit Internet Archive. With the so-called *Wayback Machine*, archived websites can be viewed that may have been changed or removed in the meantime. The *Wayback Machine* always indicates the time of retrieval.

¹¹ The term "source" can be used in the case of commercial espionage software to designate both the manufacturer and its customers

Proprietary routines – in contrast to open software libraries which are routinely used in many different software projects of various manufacturers¹² – refer to those parts of the program code that were genuinely conceived, designed and programmed by the manufacturer himself and, moreover, were not made available to the public as source code but only as a compiled program.

Since these routines are not available to the public as source code, an application with largely "literally identical" routines as in software written by independent third parties in any case requires explanation but is implausible from a certain level of complexity onwards: In principle it is possible that independent and unknown third parties have carried out an examination of the sample and copied functionalities for their own independent piece of software. However, with increasing complexity and mutual dependencies on third-party components this hypothesis becomes implausible.

The degree of implausibility depends on the following factors:

a) Effort of a re-implementation

Software routines that are not only locally executable but also require other external systems or programs for their function would, in the case of a copy by a third party, require the effort also to implement the unknown technical counterpart.

The effort required to re-implement the unknown component would usually quickly exceed the effort that would have been saved by copying the known component. Proprietary routines that have additional dependencies – especially on an equally proprietary remote peer – are almost certainly an indication of equal authorship.

b) Further development

There are two main motivations for copying proprietary routines by third parties:

- (a) Saving effort and costs and
- (b) Misleading potential analyzes to identify the manufacturer.

Further technical development of the copied routines would be inconsistent with both motivations and is, therefore, proof of fundamental knowledge of the technical relationships without which further development is not possible.

A continuous and consistent further development of used software routines over different chronological samples can consequently be seen as a strong indication of equal authorship.

c) Concealment techniques

Software manufacturers try to defend themselves technically against decompiling their products with different methods and for various reasons. On the one hand, this is done to prevent extraction and copying, but in the case of malicious software it is also done in particular to conceal the malicious functionality of the software and thus make detection (e.g. by malware scanners) more difficult. This process is called obfuscation¹³.

The use of *obfuscation* techniques has a strong influence on the effort required to copy and transfer foreign code routines and hence has corresponding consequences for the a) Effort of a re-implementation or even a b) Further development of the extracted code.

Furthermore, *obfuscation* is a technical "one-way street", which at the time of compilation, creates a more complex compilation than would normally result from the source code.

¹² see 1. Versions of software and libraries used during creation, page 7

¹³ For an explanation of terms, see [https://en.wikipedia.org/wiki/Obfuscation_\(software\)](https://en.wikipedia.org/wiki/Obfuscation_(software))

Using different methods of *obfuscation* results in very different compilations. The used *obfuscation* methods also allow conclusions about the origin: The *obfuscation* method itself is a feature of the software.

Programmers can choose from a variety of available free or proprietary *obfuscation* methods, or individually develop and apply their own methods of *obfuscation*. The use of an independently developed *obfuscation* can be considered as proof for the same origin and authorship since the original *obfuscation* routine is or was not accessible to third parties.

3. Naming of functions and variables

Program code is usually structured in modules and *functions*¹⁴. Functions cover partial functionalities of the program and are usually named according to their functionality to keep the source code readable and understandable for the programmers.

The goal of legibility is also served by the postulation and observance of *coding conventions*¹⁵ within a software project. These form a kind of "writing style" within a team or a company and usually include, among many other aspects of programming, specifications regarding

- Structuring of the code
- Use of patterns
- Naming conventions for functions and variables
- Use of Code Obfuscation
- Modularization

If two samples are highly similar with respect to these attributes, this is a strong indication that the samples are from the same programming team.

4. References to samples of known origin

All of the information so far provided in section B. *Determination* of the origin is only suitable for analyzing different samples for similarities and therefore limited to answering the question of whether the samples come from the same or different sources.

Provided that there is sufficient evidence of identical authorship, the *attribution* to a specific manufacturer can be done using samples of known origin: If the authorship of a sample is properly established, the methods described above can be used to determine the probability that a different sample comes from the same source.

In August 2014, it became known that the company *Gamma International*, which is part of the FinFisher group of companies, had been compromised by a hacker. As a result of this attack, a total of over 40GB of extracted data was published¹⁶. Part of the publication¹⁷ included source code of various malware products from the FinFisher group, as well as advertising material and internal information. The material

¹⁴ For a definition of terms see <https://en.wikipedia.org/wiki/Subroutine>

¹⁵ For a definition of terms, see https://en.wikipedia.org/wiki/Coding_conventions

¹⁶ Netzpolitik.org; Meister, Andre (2014): Gamma FinFisher hacked: ad videos of exploits and source code of FinFly Web published <https://netzpolitik.org/2014/gamma-finfisher-gehackt-werbe-videos-von-exploits-und-quel-text-von-finfly-web-veroeffentlicht/> accessed 19 December 2019.

¹⁷ the complete data set is publicly available as Torrent:
<https://www.dropbox.com/s/n7xch2vqc9p5x3e/finfisher.torrent?dl=1> accessed on 19 December 2019

Magnet-Link:

<magnet:?xt=urn:btih:4e8564f0edcb3875ad2dbb9658ca3d615cc6c152&dn=finfisher&tr=http://bt.careland.com.cn:6969/announce&tr=udp://tracker.coppersurfer.tk:6969/announce&tr=udp://tracker.openbittorrent.com/announce>

find . -name "*.apk" -exec shasum -a 256 {} \;
abcb11c4787c62ab90cecc262f6fc98bd7f73335ac631c2e9aec8734088e91aa ./qateam/ak/ANDR.apk
2795e777d897857ab6fa19f85687a23a6071ab665299a47b8b952cb4e7056a07 ./qateam/ak/demo-de/4.40/Android.apk
1935f2e52832df910edc1b7ef17b53d8c852fe66ec3afbe490ffc2ef057452b3 ./qateam/ak/demo-de/4.51/Android/AKDEMO.apk
045161094b9f6b98c4ef87d2324f4bb8a0d0fbf58e349a37d11622aef2e6b051 ./qateam/ak/demo-de/4.51/Android/ANDDemo.apk
587b110da2ef9c59b18f01e97e9b12628f3e7b2e88611f7cb28a6efccb0aaba2 ./qateam/tm/tmWoot.apk
704d599fe51e7a0f982438c13983fb936dd1530f659e8036bee69752221ef7d7 ./qateam/ta/440/ANDxJoe.apk
1ea335d1d5f99aebel1a516d6b267ba53c38438648874752eb0438edffde380d ./qateam/ta/430/zefix.apk
1507ee069906d2a42216d77ef51d42a35efcc59b005b55d8ea771749057296db ./qateam/ta/438/trekki.apk
60dc08ab28db5ba3a56734097954861a525b4b384e8067e3eaac551c4cb8ece3 ./qateam/ta/428/testAD.apk
84d39e5c6db75801a85cc5d2557ab536abe40496b23eba6b3e5c1722975d8f32 ./qateam/ta/428/428.apk
84d231e6eale2e3283c3e9cbfcabeded0d7e5723852e378e0caf5bb001501938 ./qateam/ta/428/defs.apk
26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafba1 ./qateam/ta/release421/421and.apk

These samples of doubtlessly clarified origin serve as a reference for the analysis of differences, similarities and further developments of the other available samples from an origin that has not been clarified beyond doubt. Differences – unless they can be explained as further developments – are taken as indications against joint authorship. Similarities and further developments, on the other hand, are considered as indications of joint authorship.

Results

A. Production time of the "adalet" sample

Several independent studies have already been published on the production date of the "adalet" sample¹⁹. Beyond the evidence already provided, further indications could be collected. These are summarized below along with the replication of selected previously known findings.

Versions of software and libraries used in the creation

a) *SQLite-Version 3.13.0*

The "adalet" sample has two native code libraries `library.so` and `libsqliteY.so` in the APK resources. The `library.so` library exports the functions shown in Figure 2.












Name	Address	Ordinal
 <code>Java_com_esn_wal_audio_ogg_VorbisFileInputStream_readStreamIdx</code>	000000000006E194	
 <code>Java_com_esn_wal_audio_ogg_VorbisFileInputStream_closeStreamIdx</code>	000000000006E3A8	
 <code>rinit</code>	00000000000192B0	
 <code>Java_com_esn_wal_audio_ogg_VorbisFileInputStream_create</code>	000000000006DED0	
 <code>Java_com_esn_wal_audio_ogg_VorbisFileInputStream_skipStreamIdx</code>	000000000006E2F4	
 <code>Java_org_customer_fu_aud10_o1g1g_Aud10FileOutputStream_closeStreamIdx</code>	000000000006DBA8	
 <code>JNI_OnLoad</code>	000000000001C314	
 <code>Java_org_customer_fu_aud10_o1g1g_Aud10FileOutputStream_SoundInByte</code>	000000000006DE54	
 <code>Java_org_customer_fu_aud10_o1g1g_Aud10FileOutputStream_create</code>	000000000006D360	
 <code>Java_org_customer_fu_aud10_o1g1g_Aud10FileOutputStream_writeStreamIdx</code>	000000000006D780	
 <code>start</code>	0000000000017D90	[main entry]

Figure 2 - Functions exported by the native library `library.so`

This library is loaded by the Java application and can be used via the Java Native Interface (JNI). The JNI allows Java developers to use code and implementations of other programming languages and compilers.

SQLite is an open-source, freely available database software. It is delivered with applications in the form of a software library. On Android systems, such a library has the file extension `.so` for "*shared object*". The SQLite version 3.13.0 in use was released on 18 May 2016 (See Appendix A Release Date of SQLite Version 3.13.0).

This **proves** that the "adalet" sample was produced on **May 18, 2016 at the earliest**.

¹⁹ See in particular:

Cure53; Mario Heiderich (2018): Summary-Report ECCHR Plausibility Check https://cdn.netzpolitik.org/wp-upload/2019/09/2018-05-07_Cure53_ECCHR_Plausibility-check.pdf, accessed 19 December 2019

Access Now; Gustaf Björkstén, Lucie Krahulcova (2018): ALERT: FINFISHER CHANGES TACTICS TO HOOK CRITICS <https://www.accessnow.org/cms/assets/uploads/2018/05/FinFisher-changes-tactics-to-hook-critics-AN.pdf>, accessed 19 December 2019

GFF: Annex 1: Technical Appendix (2019): <https://freiheitsrechte.org/home/wp-content/uploads/2019/09/2019-07-04-Fin-Fisher-Anhang-1-Technischer-Appendix-EN.pdf>, accessed 19 December 2019

Timestamps in Certificates

The certificate used to sign the "adalet" sample can be examined by using OpenSSL²⁰ or keytool²¹. The validity of the **certificate begins on October 10, 2016**, which usually marks the **time of the creation** of the certificate.

The "adalet" sample is signed with a certificate that is **valid from October 10, 2016**. It can be assumed with a high degree of probability that it was produced **after this date**. This finding is already mentioned in the technical appendix to the GFF²² criminal complaint and could be replicated in the context of the present analysis.

```
$ keytool -printcert -file 10_apktool-output/adalet.out/original/META-INF/CERT.RSA
Owner: CN=RMS
Issuer: CN=RMS
Serial number: 36891ece
Valid from: Mon Oct 10 05:17:01 CEST 2016 until: Fri Oct 04 05:17:01 CEST 2041
Certificate fingerprints:
    SHA1: 98:5D:08:CD:5F:1B:B3:30:28:CA:C6:20:AE:D1:93:2D:DD:26:91:E1
    SHA256:
1E:62:1A:88:3B:CD:9D:1B:D6:D5:61:11:C4:88:EE:10:D4:67:1D:2C:A6:64:F7:27:FE:72:59:47
:8A:68:79:67
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 95 E9 6E 35 21 63 62 98   AF A7 24 C6 9B EF 33 EA   ..n5!cb...$.3.
0010: 98 A4 18 89               ....
]
]
```

Listing 1. Start of validity of the certificate used to sign the "adalet" sample

²⁰ <https://www.openssl.org/>, accessed 19 December 2019

²¹ <https://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html>, accessed on 19 December 2019

²² https://netzpolitik.org/2019/wir-stellen-straftanzeige-zollkriminalamt-ermittelt-gegen-finfisher-wegen-illegalem-export-des-staatstrojaners/#2019-07-05_Straftanzeige-FinFisher-Tuerkei_Anhang-Technik_A_Konfigurationsoptionen, accessed 25 September 2019.

Timestamps in configurations and log files

The file `resources/build-data.properties` of the „adalet“ sample contains information about the exact time of compilation of the component `GSMcore`:

```
build.changelist.as.int=134102376
build.depot.path=//depot/branches/gmscore_apks_release_branch/127717789.1/google3
build.client=build-secure-info\:(SrcFS)
build.citc.snapshot=-1
build.verifiable=1
build.time=Fri Sep 23 14\:39\:54 2016 (1474666794)
build.versionmap=map 127717789 default { // } import buildenv/9666;
build.label=gmscore_v6_RC40_sdk_only
build.build_id=3c22240a-40cb-4352-b63a-bf1baaf5201e
build.timestamp=1474666794
build.timestamp.as.int=1474666794
build.target=blaze-out/gcc-4.X.Y-crosstool-v18-hybrid-grtev4-k8-
opt/bin/java/com/google/android/gmscore/integ/client/3p_monolithic_raw_pre_munge_de
ploy.jar
build.changelist=134102376
build.tool=Blaze, release blaze-2016.07.09-3 (mainline @126938038)
build.client_mint_status=1
build.gplatform=gcc-4.X.Y-crosstool-v18-hybrid-grtev4-k8
build.location=social-builder-pool-
gmscore@vnay84\:/google/src/files/134102376/depot/branches/gmscore_apks_release_bra
nch/127717789.1/READONLY
```

Listing 2 Build properties of the „adalet“ sample

The property `build.time` indicates that parts of the "adalet" sample were created **on 23 September 2016**. This finding is already mentioned in the technical appendix to the GFF and could be replicated in the scope of the present analysis.

B. Sample origin

Used Certificates

Some of the samples examined in the present analysis were signed using the same certificate. *Table 2- Overview of the certificates used to sign the different samples* provides an overview and summarizes hashes, target ID, fingerprint of the signature certificate and its date of issue.

The origin of sample *421and* has been clarified beyond doubt. The samples *JHANUK*, *Andriod* and *derise* were signed with the same certificate and can also be unambiguously assigned to the FinFisher group using this method and are highlighted in color in Table 2.

The samples *JHANUK*, *Andriod* and *derise* also originate from the FinFisher group of companies.

The samples *AKDemo*, *ANDDemo*, *428*, *tmWoot*, *ANDR*, *Android*, *ANDxJoe*, *trekki*, *zefix*, *testAD* and *defs* were signed in the same year with a different certificate than the samples mentioned above. They all come from the same source as sample *421and* and can be clearly assigned to the FinFisher group of companies based on the leak.

The samples *AKDemo*, *ANDDemo*, *428*, *tmWoot*, *ANDR*, *Android*, *ANDxJoe*, *421and*, *trekk*, *zefix*, *testAD* and *defs* are from the FinFisher group.

The samples *flash28* `46690ef267f21b5840377833e7af51b19fbd343bac97d6eb66b186d58ba3f9b3` and *adalet* also have a common signature certificate and come from the same source.

The remaining samples were each signed with individual certificates. From the manufacturer's point of view this would be a follow-up reaction to the publication of the leak in 2014.

Table 2- Overview of the certificates used to sign the different samples

hash	target id	cert sha1	cert date
2e96e343ac10f5d9ace680e456c083e4eceb108f7209aa1e849f11a239e7a682	again	BE:57:C4:31:27:F3:44:B4:75:CA:F7:D7:BC:F1:3F:BC:03:CF:A9:F0	Tue Dec 06 11:52:53 CET 2011
0d798ca0b2d0ea9bad251125973d8800ad3043e51d4cc6d0d57b971a97d3af2d	JHANUK	60:6D:58:9D:C6:F4:42:17:0A:75:B7:BC:03:20:59:34:58:C7:C0:F2	Wed Feb 22 09:53:18 CET 2012
72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537	Andriod	60:6D:58:9D:C6:F4:42:17:0A:75:B7:BC:03:20:59:34:58:C7:C0:F2	Wed Feb 22 09:53:18 CET 2012
363172a2f2b228c7b00b614178e4ffa00a3a124200ceef4e6d7edb25a4696345	derise	60:6D:58:9D:C6:F4:42:17:0A:75:B7:BC:03:20:59:34:58:C7:C0:F2	Wed Feb 22 09:53:18 CET 2012
1935f2e52832df910edc1b7ef17b53d8c852fe66ec3afbe490ffc2ef057452b3	AKDemo	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
045161094b9f6b98c4ef87d2324f4bb8a0d0fbf58e349a37d11622aef2e6b051	ANDDemo	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
84d39e5c6db75801a85cc5d2557ab536abe40496b23eba6b3e5c1722975d8f32	428	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
587b110da2ef9c59b18f01e97e9b12628f3e7b2e88611f7cb28a6efccb0aaba2	tmWoot	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
abcb11c4787c62ab90cecc262f6fc98bd7f73335ac631c2e9aec8734088e91aa	ANDR	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
2795e777d897857ab6fa19f85687a23a6071ab665299a47b8b952cb4e7056a07	Android	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
704d599fe51e7a0f982438c13983fb936dd1530f659e8036bee69752221ef7d7	ANDxJoe	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafba1	421and	60:6D:58:9D:C6:F4:42:17:0A:75:B7:BC:03:20:59:34:58:C7:C0:F2	Wed Feb 22 09:53:18 CET 2012
1507ee069906d2a42216d77ef51d42a35efcc59b005b55d8ea771749057296db	trekki	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
1ea335d1d5f99aabe1a516d6b267ba53c38438648874752eb0438edfffd380d	zefix	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
60dc08ab28db5ba3a56734097954861a525b4b384e8067e3eaac551c4cb8ece3	testAD	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
84d231e6eale2e3283c3e9bcfbcabed0d7e5723852e378e0caf5bb001501938	defs	01:AB:EB:87:BA:F5:AC:8C:52:C7:85:F9:F3:83:84:C2:40:25:62:EB	Wed Oct 10 11:42:12 CEST 2012
46690ef267f21b5840377833e7af51b19fbd343bac97d6eb66b186d58ba3f9b3	flash28	98:5D:08:CD:5F:1B:B3:30:28:CA:C6:20:AE:D1:93:2D:DD:26:91:E1	Mon Oct 10 05:17:01 CEST 2016
23f154723213452634abe6063fd07bd3a38700a6b0ba4117db3224ae1411dada	flash28	35:D6:63:83:05:EB:5E:46:FB:FF:BE:17:AA:6A:27:3B:E9:9B:A6:3F	Tue Jul 18 14:01:19 CEST 2017
c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52dlb2ad7212ac29926436e	adalet	98:5D:08:CD:5F:1B:B3:30:28:CA:C6:20:AE:D1:93:2D:DD:26:91:E1	Mon Oct 10 05:17:01 CEST 2016
77b4d11e369ac5dec4e951e5879248c1c9a84d756c06d89875f113e4c6469464	cleaner	F2:9A:B4:44:A3:6C:EB:B6:41:4F:4A:8F:90:AC:5F:EE:A8:99:3A:AD	Fri Dec 21 06:56:17 CET 2018
31fa1129d8e682a90913cc28b4e5d6b064131c93a6d86118d94f93918ed6e2f8	whistel	8B:90:35:F5:15:37:4A:6E:72:67:C0:9C:11:88:F6:EA:AC:BF:30:56	Fri Dec 21 06:50:19 CET 2018
49c12654aaee1b089268931307f36a5d0d020325226328f780dc152b2f04b281	\$container\$	3B:9B:00:BF:E8:97:68:49:B3:6C:C1:61:69:BA:D7:EB:A3:AE:D3:EE	Fri Sep 07 06:43:41 CEST 2018
269227c4c4770e109e53c6cf87bd9bde367843c4806f5975c5aa317f318e28a9	PyawApp	59:11:E3:5C:9F:0F:A3:5B:40:A8:43:50:50:C2:61:BF:ED:3E:03:FB	Wed Jun 20 10:19:44 CEST 2018
241c38fd3cafc37f496fb7e1872924f21bf1263e17a81d03981dd29b531e4623	network	ED:F1:C2:AE:F0:33:DE:43:D3:F1:6E:61:F3:26:7A:2D:42:9D:4A:06	Wed Dec 05 08:54:34 CET 2018
d8f6abc6cb1388da6b2870f06d52036a435407d6bf2c0b43684fd72edc4a9e77	Disk	E7:FF:50:BE:EC:65:DD:82:23:67:4F:C3:B8:F9:0C:57:04:73:9C:1A	Wed Dec 05 08:45:37 CET 2018
aa299745edf2e55531c9a8304b57f9bee8f37a4c3f4be56260bad096c7ealc03	FunVoic	FF:9C:FD:9F:FD:87:C5:66:54:47:81:22:60:C8:22:83:E0:BE:DB:52	Fri Dec 21 06:53:39 CET 2018
3f8baeae01980e77fa905216e291b6478105295c8372a003d73e9086b0b3e964	Diary	AD:71:4F:FA:27:E7:33:A1:96:B1:AC:F0:7C:5B:E5:51:8F:6B:D3:32	Fri Dec 14 07:09:54 CET 2018
ff8aaf49f4377e6ee162f1f0778f98e33dd2a8df2d96de6ba766851ee436467e	myphone	15:43:02:7A:5D:53:9C:67:5E:9F:F3:80:64:FF:2D:AC:DB:86:26:A2	Fri Dec 21 07:00:23 CET 2018

Matching proprietary routines

a) Provisioning

Assumption: All relevant samples use the **same proprietary mechanism** for the provisioning of malware by the end customer: The case specific configuration is hidden in the Android APK via *covert channel*²³. All extractable configurations are available in a binary format whose structures have the same patterns among each other.

Definition of terms: The process of provisioning refers to setting up user rights and the corresponding allocation of connections, services, applications and storage space. For example, in the case of malware of the "Trojan horse" type, provisioning refers to the customer-specific settings of the *Command-and-Control* infrastructure which represents the return channel for extracted information. In the hypothetical case of malware being used by government agencies in Egypt²⁴ and Germany for example, it would be common practice to maintain differently configured samples and different infrastructures for the two customers of the FinFisher group: The samples of the two customers would be subject to *different commissions*.

Since the provisioning conceals sensitive information about the client, it is understandable that measures are taken to conceal this information so that the origin of the samples cannot be easily determined in the event of discovery. The more specific and complex this concealment takes place, the less likely it is that samples from different manufacturers will be randomly similar.

At the same time, manufacturers of commercial malware are forced to use individual obfuscation mechanisms to avoid simple detection or automated detection. In the case of these samples, a so-called *Covert Channel* is used.

Proof: The present samples from the years 2012 to 2019 all bear a striking similarity to a demo Trojan²⁵ published in 2012, apparently attributable to the company Gamma International (see Figure 3), and another sample²⁶ from the same year, which was published via a Vietnamese IP address. This sample has been analyzed by numerous organizations, including the Trustwave Spiderlabs²⁷. The same mechanism is also used in Sample 421 and which can undoubtedly be attributed to the same manufacturer, since it originates from the data leak from the FinFisher group described above.

To be able to configure a Trojan with the appropriate parameters, the examined samples relied on metadata in file attributes in the Android APK files. The approach is similar to that of steganography²⁸.

From a technical point of view, an APK file is a ZIP archive in which several files are bundled together. The specification provides 6 characters for each file in an APK in which file attributes can be encoded. The files themselves can also be empty.

²³ A *covert channel* is a parasitic communication channel that uses bandwidth (excess information capacity) of a legitimate communication channel to transmit information. See https://en.wikipedia.org/wiki/Covert_channel, accessed 01 January 2020

²⁴ For other hypothetical examples see Bil Marczak, John Scott-Railton, Adam Senft, Irene Poetranto, and Sarah McKune (2015): Pay No Attention to the Server Behind the Proxy Mapping FinFisher's Continuing Proliferation <https://citizenlab.ca/2015/10/mapping-finfishers-continuing-proliferation/>, accessed 19 December 2019

²⁵ 72a522d0d3dcd0dc026b02ab9535e87a9f55664bc5587fd33bb4a48094bce0537

²⁶ 363172a2f2b228c7b00b614178e4ffa00a3a124200ceef4e6d7edb25a4696345

²⁷ Grunzweig, Josh (2012): FinSpy Mobile - Configuration and Insight <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/finspy-mobile-configuration-and-insight/>, accessed 19 December 2019

²⁸ „Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video.“, see <https://en.wikipedia.org/wiki/Steganography>, accessed 19 January 2020

39

/ 60

Community Score

39 engines detected this file

72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537

08CFFA8F55BE4BBED2704395876B618F.apk

android

apk

139.47 KB

Size

2019-09-04 18:44:49 UTC

19 days ago

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 1

AegisLab

① FinFisher_1

AhnLab-V3

① Android-Trojan/FinSpy.5919b

Alibaba

① Monitor.Android/FinSpy.dafbe56e

Arcabit

① Android.Monitor.FinSpy.B

Avast

① Android.CardServ-AU [Trj]

Avast-Mobile

① Android.FinFisher-A [Trj]

AVG

① Android.CardServ-AU [Trj]

Avira (no cloud)

① ANDROID/Agent.AFV.Gen

BitDefender

① Android.Monitor.FinSpy.B

CAT-QuickHeal

① Android.FinSpy.A (PUP)

ClamAV

① Andr.Malware.Agent-1687991

Comodo

① Malware@#1jxdb4dfwxuvx

Cyren

① AndroidOS/FinSpy.A

DrWeb

① Android.Finspy.origin

Emsisoft

① Android.Monitor.FinSpy.B (B)

ESET-NOD32

① Android/Belesak.A

F-Prot

① AndroidOS/FinSpy.A

FireEye

① Android.Monitor.FinSpy.B

Fortinet

① Android/FinSpy.Altr

GData

① Android.Trojan.FinSpy.A

Ikarus

① Trojan.AndroidOS.Belesak

K7AntiVirus

① Trojan (0001140e1)

K7GW

① Trojan (0001140e1)

Kaspersky

① Not-a-virus.HEUR.Monitor.AndroidOS.FI...

MAX

① Malware (ai Score=100)

McAfee

① Artemis!08CFFA8F55BE

McAfee-GW-Edition

① Artemis!PUP

Microsoft

① PUA:Win32/Bitrepeyp.B

NANO-Antivirus

① Trojan.Android.Finspy.bdoxek

Qihoo-360

① Trojan.Android.Gen

Sophos AV

① Andr/FinSpy-A

Symantec

① Trojan.Gen.MBT

Symantec Mobile Insight

① Backdoor.Finfish

Tencent

① Privacy.Android.Finspy.a

Figure 4 - Classification of the sample „Andriod“ as FinFisher/FinSpy by established anti -virus detectors³¹

³¹ Analysis of the Sample 72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537
<https://www.virustotal.com/gui/file/72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537/detect/on>, accessed 19 December 2019

The *Android-Demo-Sample* from 2012 reveals the following configuration after combing the meta data of the empty files, which is already visible in the output of the tool *hexdump*.

```
$ hexdump -C Andriod.conf
00000000 21 02 00 00 90 5b fe 00 19 02 00 00 a0 33 84 00 |!....[.....3...|
00000010 0c 00 00 00 50 13 fe 00 00 00 00 00 10 00 00 00 |....P.....|
00000020 60 57 fe 00 00 00 00 00 00 00 00 00 0c 00 00 00 |`W.....|
00000030 40 15 fe 00 00 00 00 00 0f 00 00 00 70 58 fe 00 |@.....pX...|
00000040 41 6e 64 72 69 6f 64 0c 00 00 00 40 61 84 00 3c |Andriod....@a.<|
00000050 00 00 00 0d 00 00 00 90 64 84 00 82 87 86 81 83 |.....d.....|
00000060 26 00 00 00 70 37 80 00 64 65 6d 6f 2d 30 31 2e |&...p7..demo-01.|
00000070 67 61 6d 6d 61 2d 69 6e 74 65 72 6e 61 74 69 6f |gamma-internatio|
00000080 6e 61 6c 2e 64 65 0c 00 00 00 40 38 80 00 57 04 |nal.de....@8..W.|
00000090 00 00 0c 00 00 00 40 38 80 00 58 04 00 00 0c 00 |.....@8..X.....|
000000a0 00 00 40 38 80 00 59 04 00 00 15 00 00 00 70 63 |..@8..Y.....pc|
000000b0 84 00 2b 34 39 31 37 32 36 36 36 32 33 36 34 16 |..+491726662364.|
000000c0 00 00 00 70 6a 84 00 2b 34 39 38 39 35 34 39 39 |...pj...+49895499|
000000d0 38 39 38 39 30 13 00 00 00 70 6a 84 00 2b 36 35 |89890....pj...+65|
000000e0 39 37 32 39 34 37 30 34 0f 00 00 00 70 66 84 00 |97294704....pf...|
000000f0 41 6e 64 72 69 6f 64 0c 00 00 00 40 65 84 00 81 |Andriod....@e...|
00000100 74 63 0f 0c 00 00 00 40 21 fe 00 f3 03 00 00 0c |tc.....@!.....|
00000110 00 00 00 40 0d 80 00 0a 00 00 00 0c 00 00 00 40 |...@.....@|
00000120 68 84 00 00 00 00 00 0c 00 00 00 40 3b 80 00 a8 |h.....@;...|
00000130 00 00 00 0a 00 00 00 90 60 84 00 fd 10 0a 00 00 |.....|
00000140 00 90 62 84 00 c0 00 09 00 00 00 b0 67 84 00 00 |..b.....g...|
00000150 08 00 00 00 90 c6 71 00 8c 00 00 00 90 79 84 00 |.....q.....y...|
00000160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
000001a0 01 01 01 01 00 01 01 00 00 00 00 00 00 00 00 00 |.....|
000001b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
000001e0 00 00 00 00 3d 00 00 00 90 34 45 00 35 00 00 00 |....=....4E.5...|
000001f0 a0 33 45 00 0c 00 00 00 40 41 45 00 e8 03 00 00 |.3E.....@AE.....|
00000200 0c 00 00 00 40 40 45 00 58 02 00 00 09 00 00 00 |....@@E.X.....|
00000210 30 42 45 00 00 0c 00 00 00 90 64 84 00 87 86 85 |0BE.....d.....|
00000220 81 0a |..|
```

The *FinSpy* variants for Android which appeared at the earliest in 2016, at the latest in January 2017 (*flash28*³²) and July 2017 (*adalet*³³) and beyond that in March and May 2019, use the same procedure to store the case-based configuration in the Android APK during provisioning by the end customer.

The individual configuration data from 2012 to 2019 show a similarity in their nature which can by no means be accidental. All variants hide the configurations in the *Central Directory Structure* (CDS) blocks of the APK PK-Zip file³⁴ as *Internal* and *External File Attributes*, base64 encoded.

Thus, 6 bytes of base64-encoded data can be hidden in each CDS section. The malware extracts all of these hidden blocks from the APK, concatenates the characters and decodes the result using the usual "base64" algorithm.

All configurations are in this proprietary binary format. From the simple hex dumps of this binary data it can be seen that all examples have a basic similarity in their binary structure.

To extract the configuration data, we used the malware analysis tool³⁵ `extractConfig.rb` which was published in 2012 on the open source code platform *Github*.

³² 46690ef267f21b5840377833e7af51b19fbd343bac97d6eb66b186d58ba3f9b3

³³ c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e

³⁴ <http://www.fileformat.info/format/zip/corion.htm>

³⁵ <https://github.com/SpiderLabs/malware-analysis/blob/master/Ruby/FinSpy/extractConfig.rb>

We modified, extended, reprogrammed and published it as well³⁶ to extract configurations also from newer samples. With the help of the tools³⁷ we also published on *Github*, the configurations of the samples *Andriod*, *derise* and *421and* can be read and displayed in a human readable way:

The "Andriod" sample contains a configuration with German mobile and Munich landline numbers, as well as a domain that can be assigned to the FinFisher group. Several parts of the FinFisher group are located in Munich. Furthermore it was publicly documented in 2013 that the FinFisher group maintains or has maintained "development offices in Obersendling in Munich"³⁸. These indications are in line with the evidence already presented in section 1.

Section 1. Certificates used on page 8 documented proof that the *Andriod* sample can be assigned to the FinFisher group.

```
TlvTypeMobileEncryption =
  b'\x19\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig =
  b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID = "Andriod" (15)
TlvTypeMobileTargetHeartbeatInterval = 60 (12)
TlvTypeMobileTargetPositioning = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy = "demo-01.gamma-international.de" (38)
TlvTypeConfigTargetPort = 1111 (12)
TlvTypeConfigTargetPort = 1112 (12)
TlvTypeConfigTargetPort = 1113 (12)
TlvTypeConfigSMSPhoneNumber = "+491726662364" (21)
TlvTypeConfigCallPhoneNumber = "+4989549989890" (22)
TlvTypeConfigCallPhoneNumber = "+6597294704" (19)
TlvTypeMobileTrojanID = "Andriod" (15)
TlvTypeMobileTrojanUID = b'\x81tc\x0f' (12)
TlvTypeUserID = 1011 (12)
TlvTypeTrojanMaxInfections = 10 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 4349 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules = Logging: Off | Spy Call: On | Call
  Interception: On | SMS: On | Address Book: On | Tracking: On | Phone Logs:
  On | (140)
TlvTypeMobileTrackingConfigRaw =
  b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00
  @@E\x00X' (61)
TlvTypeMobileTrackingConfig =
  b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00X' (53)
TlvTypeMobileTrackingDistance = 1000 (12)
```

³⁶ <https://github.com/devio/FinSpy-Tools>

³⁷ <https://github.com/devio/FinSpy-Tools>

³⁸ Sueddeutsche Zeitung; Bastian Brinkmann, Jasmin Klofta und Frederik Obermaier (2013): FinFisher-Entwickler Gamma: Spam vom Staat. <https://www.sueddeutsche.de/digital/finfisher-entwickler-gamma-spam-vom-staat-1.1595253-0>, accessed 19 December 2019.

The *derise* sample from 2012 has a configuration with an IP address which is assigned to Vietnam³⁹, and a telephone number with the country code of Vietnam:

```
TlvTypeMobileEncryption =
  b'\xf9\x01\x00\x00\xa03\x84\x00\x0c\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig =
  b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID = "derise" (14)
TlvTypeMobileTargetHeartbeatInterval = 60 (12)
TlvTypeMobileTargetPositioning = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy = "183.91.2.199" (20)
TlvTypeConfigTargetPort = 9111 (12)
TlvTypeConfigTargetPort = 9112 (12)
TlvTypeConfigTargetPort = 9113 (12)
TlvTypeConfigSMSPhoneNumber = "+841257725403" (21)
TlvTypeConfigCallPhoneNumber = "08888" (13)
TlvTypeConfigCallPhoneNumber = "+8408888" (16)
TlvTypeMobileTrojanID = "derise" (14)
TlvTypeMobileTrojanUID = b'\x820,\x00' (12)
TlvTypeUserID = 1000 (12)
TlvTypeTrojanMaxInfections = 3 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 189 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules = Logging: Off | Spy Call: Off | Call
  Interception: Off | SMS: On | Address Book: Off | Tracking: On | Phone
  Logs: On | (140)
TlvTypeMobileTrackingConfigRaw =
  b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\x88\x13\x00\x00\x0c\x00\x00\x00
  @@E\x00X' (61)
TlvTypeMobileTrackingConfig =
  b'\x0c\x00\x00\x00@AE\x00\x88\x13\x00\x00\x0c\x00\x00\x00@@E\x00X' (53)
TlvTypeMobileTrackingDistance = 5000 (12)
```

The *421and* sample from 2014 also has a configuration with a German mobile and Munich landline number, as well as a domain that can be assigned to the FinFisher group again:

```
[...]
TlvTypeMobileTargetID = "421and" (14)
TlvTypeMobileTargetHeartbeatInterval = 120 (12)
TlvTypeMobileTargetPositioning = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy = "qa01.gamma-international.de" (35)
TlvTypeConfigTargetPort = 1111 (12)
TlvTypeConfigTargetPort = 1112 (12)
TlvTypeConfigTargetPort = 1113 (12)
TlvTypeConfigTargetPort = 80 (12)
TlvTypeConfigSMSPhoneNumber = "+491726652007" (21)
TlvTypeConfigCallPhoneNumber = "+4989549989909" (22)
TlvTypeMobileTrojanID = "421and" (14)
TlvTypeMobileTrojanUID = b'J\x99\x8f\x00' (12)
TlvTypeUserID = 1003 (12)
TlvTypeTrojanMaxInfections = 999 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 4269 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules = Logging: Off | Spy Call: On | Call
  Interception: On | SMS: On | Address Book: On | Tracking: On | Phone Logs:
  On | (140)
[...]
```

³⁹ <https://ipinfo.io/183.91.2.199>, accessed 19 December 2019, locates this IP address in Vietnam at CMC Telecom Infrastructure Company (cmctelecom.vn)

The "adalet" sample (after 2016) has a configuration with an IP address⁴⁰ assigned to the Federal Republic of Germany and a telephone number with the country code⁴¹ of Israel.

```
[...]
TlvTypeMobileTargetID = "adalet" (14)
TlvTypeMobileTargetHeartbeatInterval = 86400 (12)
TlvTypeMobileTargetPositioning = b'\x86\x82\x87\x81\x83' (13)
TlvTypeConfigTargetProxy = "94.23.165.112" (21)
TlvTypeConfigTargetPort = 443 (12)
TlvTypeConfigTargetPort = 80 (12)
TlvTypeConfigTargetPort = 53 (12)
TlvTypeConfigTargetPort = 8080 (12)
TlvTypeConfigTargetPort = 9001 (12)
TlvTypeConfigTargetPort = 9050 (12)
TlvTypeConfigTargetPort = 9040 (12)
TlvTypeConfigSMSPhoneNumber = "+97260260260" (20)
TlvTypeConfigCallPhoneNumber = "+97918918918" (20)
TlvTypeMobileTrojanID = "adalet" (14)
[...]
TlvTypeInstalledModules = Logging: Off | Spy Call: Off |
Call Interception: Off | SMS: Off | Address Book: Off | Tracking: Off |
Phone Logs: Off | (140)
```

The *flash28* sample (after 2016) has a specifically unsuspicious domain (marketconsulting.ddns.net), as well as telephone numbers with the country code of Israel.

```
[...]
TlvTypeMobileTargetID = "flash28" (15)
TlvTypeMobileTargetHeartbeatInterval = 43200 (12)
TlvTypeMobileTargetPositioning = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy = "103.208.86.204" (22)
TlvTypeConfigTargetProxy = "marketconsulting.ddns.net" (33)
TlvTypeConfigTargetPort = 80 (12)
TlvTypeConfigTargetPort = 8080 (12)
TlvTypeConfigTargetPort = 443 (12)
TlvTypeConfigSMSPhoneNumber = "+97260260260" (20)
TlvTypeConfigCallPhoneNumber = "+97918918918" (20)
TlvTypeMobileTrojanID = "flash28" (15)
TlvTypeMobileTrojanUID = " r" (12)
TlvTypeUserID = 1015 (12)
TlvTypeTrojanMaxInfections = 10 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 173 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
[...]
TlvTypeInstalledModules = Logging: Off | Spy Call: Off | Call
Interception: Off | SMS: On | Address Book: On | Tracking: Off | Phone
Logs: On | (140)
[...]
```

In this section, only the samples relevant for the demonstration of evidence are discussed. A complete documentation of the configuration of all examined samples can be found in Appendix B - Configuration of all samples examined in this analysis.

⁴⁰ <https://ipinfo.io/94.23.165.112>, accessed 19 December 2019, locates this IP address at OVH GmbH in Saarbrücken, Saarland, Germany

⁴¹ https://en.wikipedia.org/wiki/List_of_country_calling_codes

The complete configuration of the samples released from spring 2019 onwards, such as

- *cleaner*
- *PyawApp*
- *FunVoic*
- *Diary*
- etc.

cannot yet be *entirely* extracted with the existing tools. The configuration data can be extracted from the APK file as described, but interpreting the configuration parameters requires further analysis of the respective samples.

It can be assumed that new configuration parameters have been introduced which are not yet taken into account by the tools available so far. We intend to adapt our published analysis software⁴² as soon as the corresponding parameters are known.

Nevertheless, when looking at the binary data, the similarity to the older samples is obvious: it is a further development of the method used over a period of seven years. Such a further development is obvious and plausible. It can be considered a strong indication that samples using this method were developed by the same team.

Evaluation:

The analysis of the correspondence of proprietary routines in the context of provisioning has two primary findings as a result:

1. All variants use the same method for hiding configurations in the Central Directory Structure (CDS) blocks of the APK PK-Zip file⁴³ as *Internal* and *External File Attributes*, base64-encoded
2. All data extracted in this way has the same proprietary binary format.

The findings for (1.) correct reading of the configuration data and (2.) correct decoding of the proprietary binary protocol were obtained by reverse engineering⁴⁴ of the provided samples.

This is a clear indication that the samples from 2012 to 2019 are from the same vendor.

Assuming the alternative hypothesis that one or more of the samples are from a different vendor, it would have to be explained how the same methods are used for the covert channel and the encoding, even though the tools required for this would not be available to a second vendor.

Although *reverse engineering* would make it possible to implement the tools used to create this provisioning data, it would be less effort for a third, uninvolved vendor to implement alternative methods of their own.

The sample *421and* comes directly from the FinFisher leak of 2014 and can be clearly assigned to the FinFisher group.

This is a clear indication that the samples - including the "adalet" sample - are from the FinFisher group.

b) Use of the same functions

In this analysis step, exemplary structural similarities in code structure and functions are analyzed to show parallels and differences. The results are presented as examples to illustrate conspicuous and coincidental

⁴² <https://github.com/devio/FinSpy-Tools>, see also section *Public documentation of test objects and methods*, page 33

⁴³ The ZIP Archive File Format, Original Documentation: <http://www.fileformat.info/format/zip/corion.htm>, accessed 19 December 2019

⁴⁴ For a definition of terms see https://de.wikipedia.org/wiki/Reverse_Engineering

matches. Due to the research question of the GFF, the focus here is on the "adalet" sample and its similarities to samples that undoubtedly originate from the FinFisher group of companies.

Using the output of the Java decompiler, a strong structural and contextual similarity between the versions of 2014 and 2016 can be illustrated.

The function `run()` in class `SmsInbox` (`Smsinbox.java`, `com\android\services\sms`) of sample 421 and in line 36, a function called `run()` is implemented. This function and its structure is shown in Figure 5 as an example.

In the class `e` (`e.java`, `org\customer\fu\slms`) of the sample *adalet*, a function called `run()` is implemented in line 66, which is shown as an example in Figure 6.

Sample 421 and originates from the FinFisher group as described in chapter References to samples of known origin on page 28.

A comparison of the code structure suggests that a *refactoring*⁴⁵ has taken place in the current version after 2014 to hide similarities in the program syntax. This would be a plausible and obvious reaction to the publication of the malware samples from the FinFisher group in 2014: *refactoring* can reduce the probability of automated detection of the malware.

Apparently, classes, variables and functions were renamed and, in addition, so-called obfuscators were used. Obfuscators specifically reduce the readability and comparability of the program code, but have no influence on the actual functionality.

In the newer variants, a so-called "leet-speak" variant of individual words is used extensively.

Here, certain letters are replaced by similar-looking numbers, for example, the letter "i" is replaced by "1", the letter "e" by "3", the letter "t" by the number "7", etc.

There are similarities with respect to the naming of code structures. Furthermore, the names themselves indicate that the author of the source code is German-speaking:

1. The class structure `com/android/services/sms` was renamed to `org/customer/fu/slms`. The character string "s1ms" is a codified version of the word "sims", which in turn is derived from "simsen"⁴⁶ or "SMS", respectively—"simsen" is a verb used exclusively in German-speaking areas⁴⁷.
2. The use of the word "fu", which in German is pronounced like the English "foo", in place of exactly that word also indicates that the naming was done by a German-speaking developer.

In addition to the similar syntax and context of both versions of the function, significant similarity can be seen in the control flow graphs of individual functions, as shown in Figure 7 and Figure 8.

Here you can compare the corresponding functions `CallLogs.run()` from 2014 and `a.run()` from 2016. In both versions, individually selected strings, such as `tmp420` in the above examples, are chosen. These are chosen consciously in context and the probability of a coincidental similarity can almost be excluded: Both versions are essentially based on an identical source code base.

⁴⁵ Refactoring a structural change of source code while maintaining the observable program behavior. For an explanation of terms see https://en.wikipedia.org/wiki/Code_refactoring

⁴⁶ compare German term "simsen": <https://www.duden.de/rechtschreibung/simsen>

⁴⁷ In the English-speaking world, the term "text" is commonly used to designate both a short message and the writing of such a message.

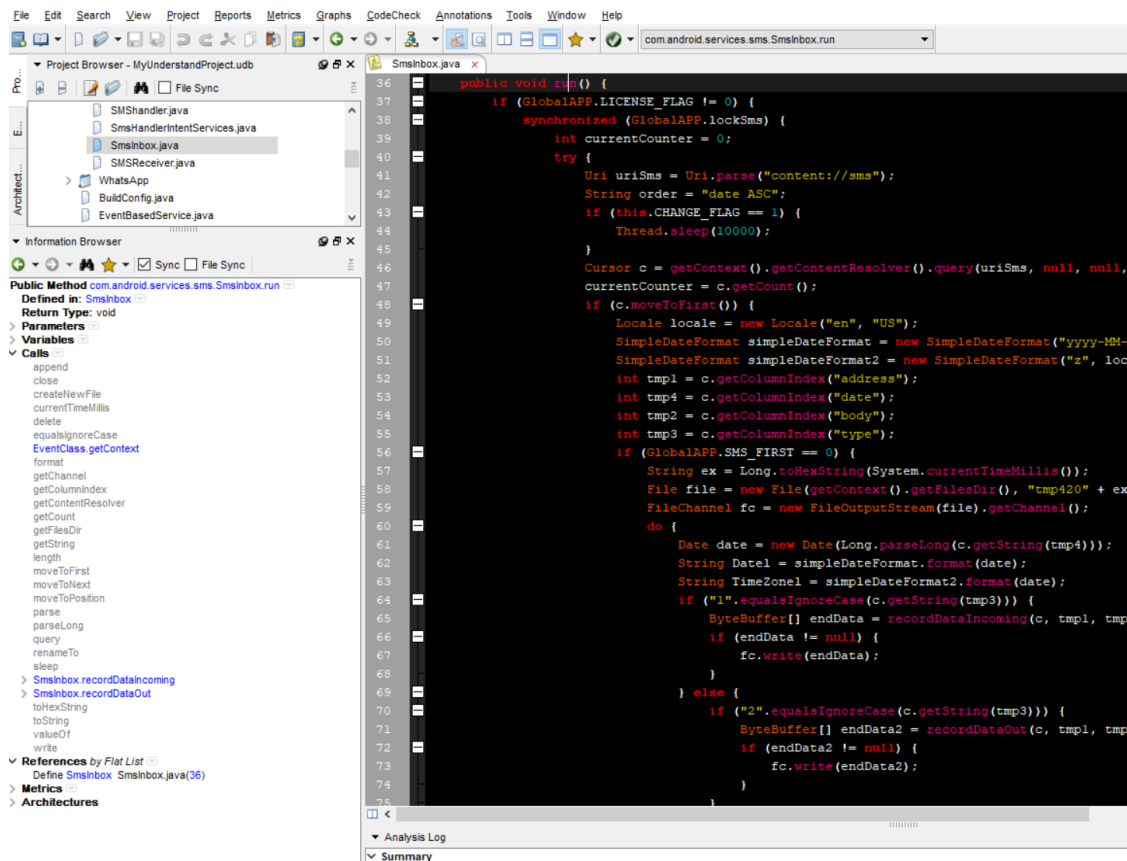


Figure 5 - Sample 421and: com\android\services\sms

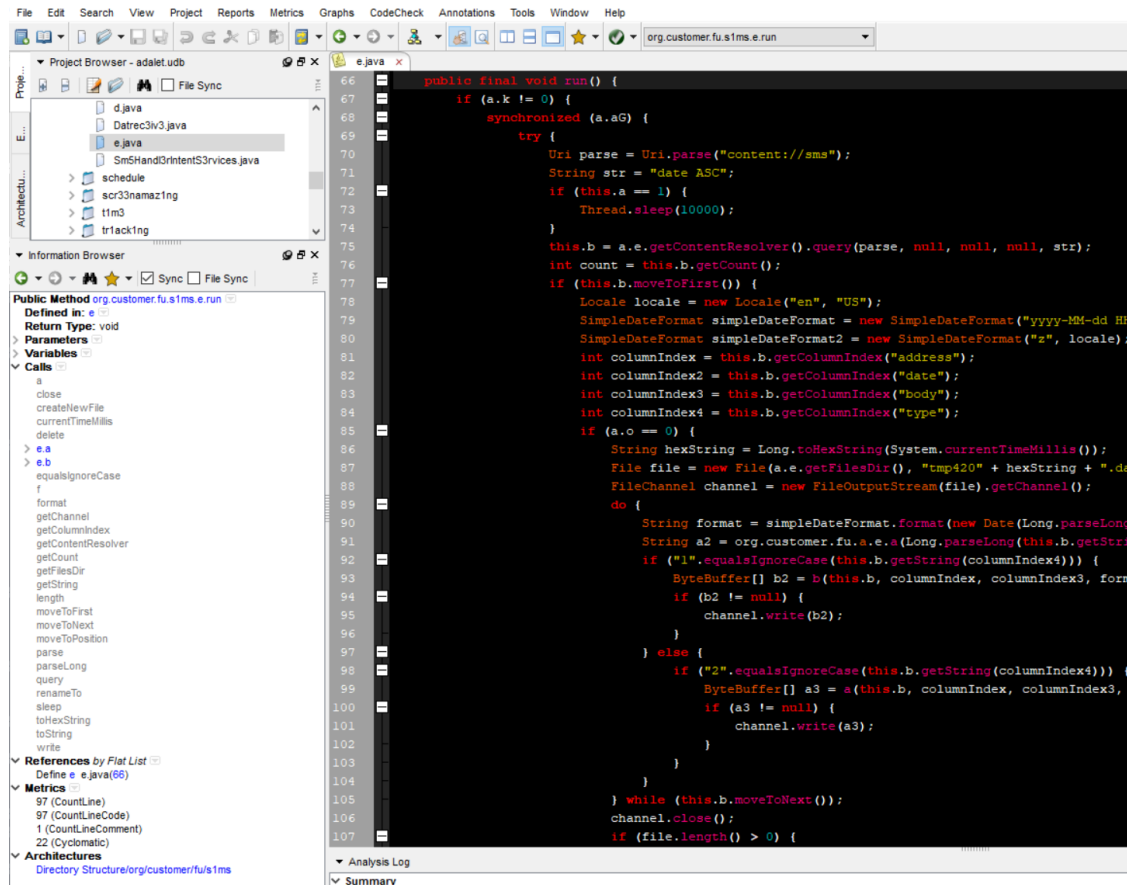


Figure 6 - Sample adalet: org\customer\fu\s1ms

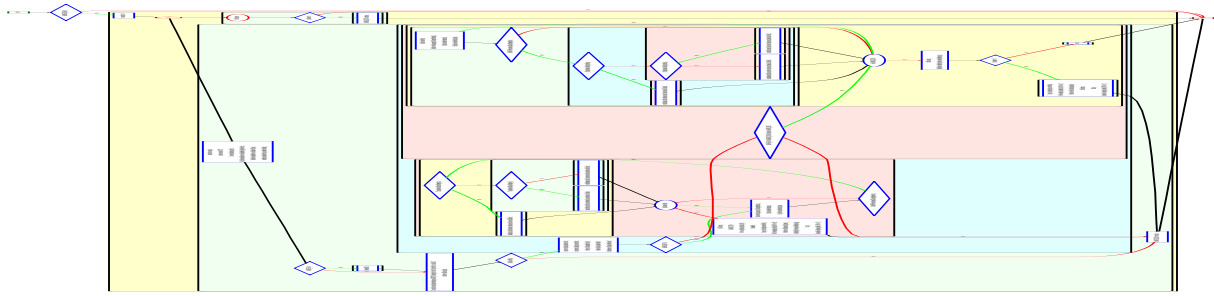


Figure 7 - Sample 421 and: ControlFlow *com.android.services.CallLogs.run()*

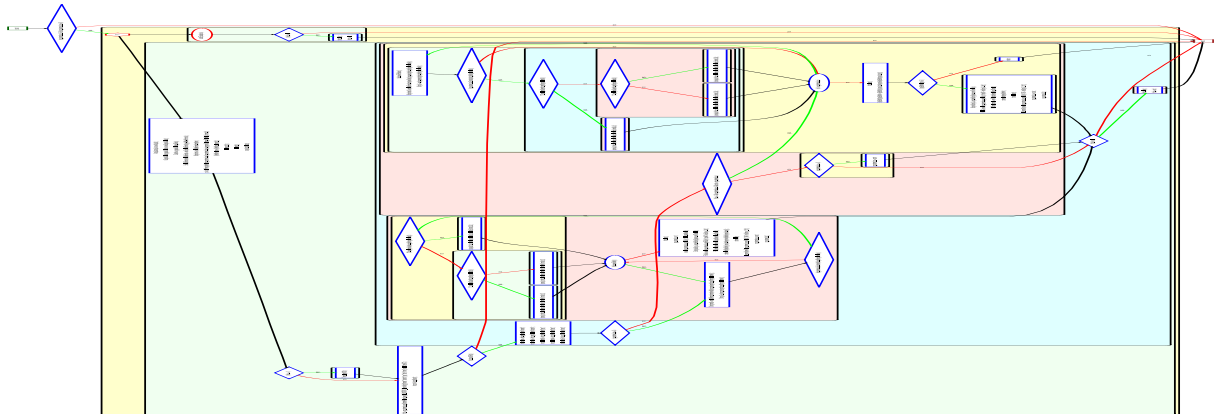


Figure 8 - Sample *adalet*: ControlFlow *org.customer.fu.e.a.run()*

Figure 7 and Figure 8 are presented in Appendix C & D in an enlarged form, and as individual files in PDF format in the software repository for this report (see section Public documentation of test objects and methods, page 33).

References to samples of known origin

b) Similarity

Section a) SQLite-Version 3.13.0 on page 13 already described the SQLite software libraries used in the *adalet* sample. The Sample *flash28* also uses the exact same libraries, as a comparison of the SHA-256 checksums shows.

adalet SHA-256 checksums:

25c7ab9603506adb1e5ec475734763a519f9be19db94e2eeddf25604471541f21	lib/arm64-v8a/library.so
18689d9d4b76a011e410802204445980ff187065b120a1c7876a04e4633dbb89	lib/arm64-v8a/libsqliteY.so
2cc662cd13e5bd6720ff1217d77baf507558fea2b7469e9926f805f5a25f5d13	lib/armeabi/library.so
e2340cfb97e7bcd2b938759291a2872d3dbe0b80b316922b7392a8b68f08d9e6	lib/armeabi/libsqliteY.so

flash28 SHA-256 checksums:

25c7ab9603506adb1e5ec475734763a519f9be19db94e2eeddf25604471541f21	lib/arm64-v8a/library.so
18689d9d4b76a011e410802204445980ff187065b120a1c7876a04e4633dbb89	lib/arm64-v8a/libsqliteY.so
2cc662cd13e5bd6720ff1217d77baf507558fea2b7469e9926f805f5a25f5d13	lib/armeabi/library.so
e2340cfb97e7bcd2b938759291a2872d3dbe0b80b316922b7392a8b68f08d9e6	lib/armeabi/libsqliteY.so

The newer samples, released in 2019, also use the Java Native Interface and deliver various libraries in the APK. In these samples there is usually only one file called `libhelper.so` which seems to be similar to the `library.so` and `libsqliteY.so` files of the older versions. If a similarity plays a role in the evaluation of significant evidence, the authors recommend that the similarities be verified in detail by examining and comparing the content of the various `.so` files.

Based on the similarities of the native libraries in the APKs of the samples from 2016 onwards it can be concluded that they use a common code base attributed to the FinFisher group.

Significant deviation

The sample with the SHA256 sum

49c12654aaee1b089268931307f36a5d0d020325226328f780dc152b2f04b281 is an exception: It has no direct similarities to the other samples. This sample contains numerous programs and a shell script in the `assets/` directory:

```
dirtycow:      ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically
linked, interpreter /system/, BuildID[sha1]=5ffa7ee5cda06134d4dfb4fc9cf838edf02e6cb1, stripped
dirtycow64:    ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically
linked, interpreter /system/, BuildID[sha1]=d9b36c62746751b05d053a8d5e92472753e6507f, stripped
holycow:       ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically
linked, interpreter /system/, BuildID[sha1]=cf3abc89fb02d3f69c4619284bce2003cbcdeea7, stripped
holysht:       a /system/bin/sh script, ASCII text executable
inst:          ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically
linked, interpreter /system/, BuildID[sha1]=cc698d6e410f74741b6306d029f90195b2f96008, stripped
myrun_as:      ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically
linked, interpreter /system/, BuildID[sha1]=1e8fd9368a845d6af8ea8dd4b367afb763005ca7, stripped
raw:           ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically
linked, interpreter \010, corrupted section header size
sepolicy-inject: ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically
linked, interpreter /system/, stripped
supersu.zip:    Java archive data (JAR)
sy.apk:         Zip archive data, at least v2.0 to extract
unzip:          ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), statically linked,
for GNU/Linux 2.6.16, stripped
```

This sample appears to be a container containing the local kernel exploit known as *dirtycow*⁴⁸. This is used for *privilege escalation*⁴⁹: By exploiting a vulnerability, an unprivileged user or process can use *dirtycow* to gain administrative privileges and thus complete control over the device. This is a prerequisite for making permanent changes to the system, for example to install malware.

This sample also contains a tool suite to "root" an Android phone⁵⁰. This tool suite calls itself *SuperSU*⁵¹ and is contained in the ZIP file of the same name. The sample has all the prerequisites to install (additional) malware on a target device.

Through a superficial analysis we were able to determine that another sample is located in this APK. This sample has the file name `sy.apk` and is identical to the sample *PyawApp*

269227c4c4770e109e53c6cf87bd9bde367843c4806f5975c5aa317f318e28a9.

This sample has already been examined by experts from the malware specialist *Kaspersky* and has been assigned to the FinSpy software family⁵². This finding is consistent with our observations. The name *PyawApp* suggests that this sample was used against Burmese citizens, as *Pyaw* is the name of a popular Burmese social network⁵³.

⁴⁸ See <https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails>, accessed 19 December 2019

⁴⁹ For an explanation of terms see https://en.wikipedia.org/wiki/Privilege_escalaion, accessed 19 December 2019

⁵⁰ For an explanation of terms see <https://www.heise.de/select/ct/2018/16/1533001012731723>, accessed 19 December 2019

⁵¹ See <https://supersu.org/>, accessed on 19 December 2019

⁵² *Kaspersky*; GREAT, AMR (2019): New FinSpy iOS and Android implants revealed ITW <https://securelist.com/new-finspy-ios-and-android-implants-revealed-itw/91685/>, accessed on 19. December 2019

⁵³ See https://play.google.com/store/apps/details?id=mm.com.pyaw&hl=en_US and <https://www.pyaw.com.mm/>, both accessed 19 December 2019

The authors presume that sample

49c12654aaee1b089268931307f36a5d0d020325226328f780dc152b2f04b281 serves the purpose to install *FinSpy* on the target device. A detailed analysis of sample

49c12654aaee1b089268931307f36a5d0d020325226328f780dc152b2f04b281 has not been conducted by the Chaos Computer Clubs, due to time constraints and because it would have been outside the scope of the central questions.

Table 3 - Overview of analysed samples

[illegible]

Conclusion

A. Determination of the date of creation

1) When was the "adalet" sample produced and used?

The "adalet" sample could not have been created before 18 May 2016. Parts of the "adalet" samples were apparently created on 23 September 2016. In addition, it seems likely that the "adalet" sample was only used after 10 October 2016.

2) Is the date or period [of creation] before or after 18 July 2015?

It has been **proven** that the sample submitted for investigation by the Gesellschaft für Freiheitsrechte was produced on **18 May 2016 at the earliest** (see SQLite-Version 3.13.0, page13). It has thus been **proven** that the sample was created **after 18 July 2015**.

The presumed earliest use could be limited to a period **after 10 October 2016** (see 2. Timestamps in certificates, page 7).

B. Determination of origin

1) Do the samples derive from different sources, or is there clear indication of co-authorship?

All samples analyzed in this study share clear indications of common authorship.

2) Can the authors of these samples be identified?

Table 3 - Overview of analysed samples shows the development of the examined samples over time. Samples that are known to originate from the FinFisher group are highlighted in red. Samples that can be assigned to the FinFisher group as a result of the analysis are highlighted in yellow. The "adalet" sample and its similarities to other samples are highlighted in orange.

The iterative developments over the years show that the sample

c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e

("adalet") submitted for examination represents a further development of the samples

abcb11c4787c62ab90cecc262f6fc98bd7f73335ac631c2e9aec8734088e91aa
2795e777d897857ab6fa19f85687a23a6071ab665299a47b8b952cb4e7056a07,
1935f2e52832df910edc1b7ef17b53d8c852fe66ec3afbe490ffc2ef057452b3,
045161094b9f6b98c4ef87d2324f4bb8a0d0fbf58e349a37d11622aef2e6b051,
587b110da2ef9c59b18f01e97e9b12628f3e7b2e88611f7cb28a6efccb0aaba2,
704d599fe51e7a0f982438c13983fb936dd1530f659e8036bee69752221ef7d7,
1ea335d1d5f99aebela516d6b267ba53c38438648874752eb0438edfffd380d,
1507ee069906d2a42216d77ef51d42a35efcc59b005b55d8ea771749057296db,
60dc08ab28db5ba3a56734097954861a525b4b384e8067e3eaac551c4cb8ece3,
84d39e5c6db75801a85cc5d2557ab536abe40496b23eba6b3e5c1722975d8f32,
84d231e6ea1e2e3283c3e9cbfcabeded0d7e5723852e378e0caf5bb001501938 and
26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafba1

from the years 2012-2014. These samples can be clearly assigned to the FinFisher group.

The authors consider it proven that the sample **derives from this source**.

Public documentation of test objects and methods

The Chaos Computer Club e. V. (CCC) is the largest European hacker association and has been mediating between technical and social developments for over thirty years. The present analysis was carried out voluntarily by Thorsten Schröder and Linus Neumann to the best of their knowledge and belief.

One of the principles of hacker ethics⁵⁴ is freedom of information. In order to meet the scientific claim of this report, the authors have published all samples analyzed in the context of this evaluation, their extracts, as well as tools developed and used by themselves.

The present analysis can be fully comprehended by professionals and those who wish to become professionals with the help of the tools and documentation we publish. We invite the German and international research community to critically review, supplement and –if necessary– correct our results.

In particular, we ask German investigating authorities, who are also customers of the FinFisher group of companies, to trace our analysis steps on the samples available to them.

All the research objects and methods required to carry out our analysis are available in the following repositories:

- **FinSpy-Tools:** Tools for the analysis of the samples mentioned here and other Android-based FinSpy samples.
<https://github.com/devio/FinSpy-Tools>
- **FinSpy documentation:** Documentation of the analyzes of individual components of the FinSpy malware, extracts, samples and helper scripts.
<https://github.com/linuzifer/FinSpy-Dokumentation>

⁵⁴ <https://www.ccc.de/de/hackerethik>

Appendix

A. Publication date and time of the SQLite-Version 3.13.0

As illustrated in Figure 9, a checksum is also published in the notification of publication, in addition to the time stamp of the publication. This combination is referred to here as `SQLITE_SOURCE_ID`.



SQLite Release 3.13.0 On 2016-05-18

1. Postpone I/O associated with TEMP files for as long as possible, with the hope that the I/O can ultimately be avoided completely.
2. Merged the [session](#) extension into trunk.
3. Added the ".auth ON|OFF" command to the [command-line shell](#).
4. Added the "--indent" option to the ".schema" and ".fullschema" commands of the [command-line shell](#), to turn on pretty-printing.
5. Added the ".eqp full" option to the [command-line shell](#), that does both [EXPLAIN](#) and [EXPLAIN QUERY PLAN](#) on each statement that is evaluated.
6. Improved unicode filename handling in the [command-line shell](#) on Windows.
7. Improved resistance against goofy query planner decisions caused by incomplete or incorrect modifications to the [sqlite_stat1](#) table by the application.
8. Added the [sqlite3_db_config\(db,SQLITE_DBCONFIG_ENABLE_LOAD_EXTENSION\)](#) interface which allows the [sqlite3_load_extension\(\)](#) C-API to be enabled while keeping the [load_extension\(\)](#) SQL function disabled for security.
9. Change the [temporary directory search algorithm](#) on Unix to allow directories with write and execute permission, but without read permission, to serve as temporary directories. Apply this same standard to the "." fallback directory.

Bug Fixes:

10. Fix a problem with the multi-row one-pass DELETE optimization that was causing it to compute incorrect answers with a self-referential subquery in the WHERE clause. Fix for ticket [dc6ebda9396087](#)
11. Fix a possible segfault with DELETE when table is a [rowid table](#) with an [INTEGER PRIMARY KEY](#) and the WHERE clause contains a OR and the table has one or more indexes that are able to trigger the OR optimization, but none of the indexes reference any table columns other than the INTEGER PRIMARY KEY. Ticket [16c9801ceba49](#).
12. When checking for the WHERE-clause push-down optimization, verify that all terms of the compound inner SELECT are non-aggregate, not just the last term. Fix for ticket [f7f8c97e97597](#).
13. Fix a locking race condition in Windows that can occur when two or more processes attempt to recover the same [hot journal](#) at the same time.

Hashes:

14. `SQLITE_SOURCE_ID: "2016-05-18 10:57:30 fc49f556e48970561d7ab6a2f24fdd7d9eb81ff2"`
15. SHA1 for `sqlite3.c`: `9b9171b1e6ce7a980e6b714e9c0d9112657ad552`

Bug fixes backported into patch release 3.8.3 (2014-02-03):

16. Added support for [common table expressions](#) and the [WITH clause](#).
17. Added the [printf\(\)](#) SQL function.
18. Added [SQLITE_DETERMINISTIC](#) as an optional bit in the 4th argument to the [sqlite3_create_function\(\)](#) and related interfaces, providing applications with the ability to create new functions that can be factored out of inner loops when they have constant arguments.

Figure 9 - https://www.sqlite.org/releaselog/3_13_0.html

The same `SQLITE_SOURCE_ID` with the value 2016-05-18 10:57:30

`fc49f556e48970561d7ab6a2f24fdd7d9eb81ff2` is also located in the file `arm64-v8a\libssqliteY.so` contained in the "adalet" sample (see Figure 10 - Disassembly of the file `libssqliteY.so` in IDA Pro 7.3)

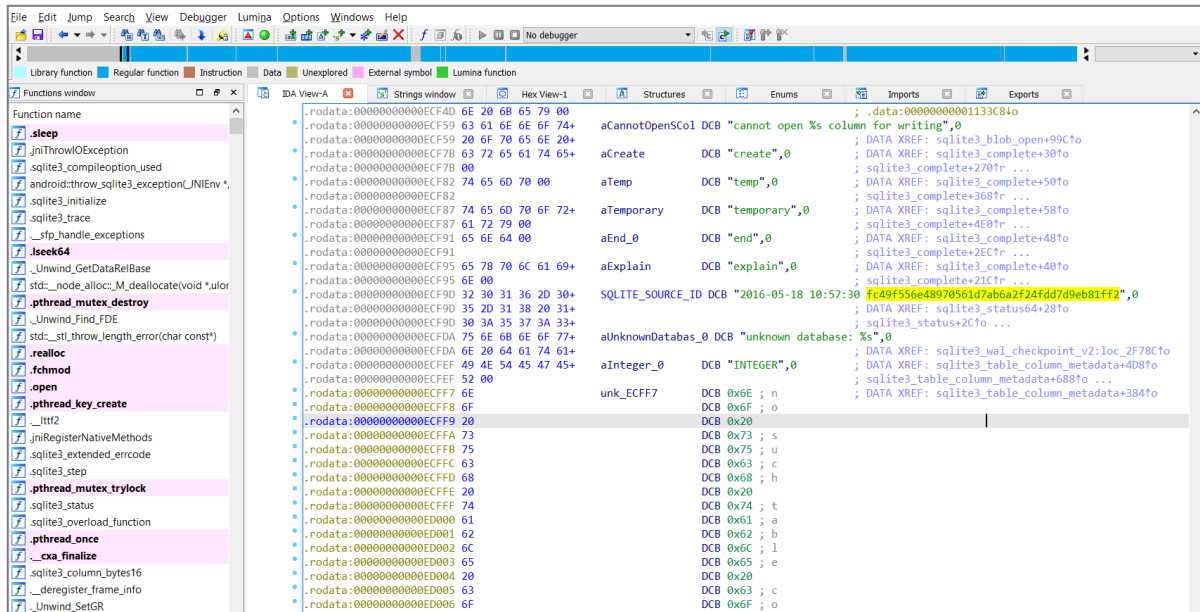


Figure 10 - Disassembly of the file libsqliteY.so in IDA Pro 7.3

The binary files of the library were provided for 32-bit and 64-bit ARM processors in the "adalet" sample. An analysis of the used compiler versions of the individual libraries was done with the tool `objdump`.

```
$ objdump -s --section .comment libsqliteY.so
```

```
libsqliteY.so:      file format elf32-little
```

```
Contents of section .comment:
```

```
0000 00474343 3a202847 4e552920 342e392e .GCC: (GNU) 4.9.
0010 78203230 31353031 32332028 70726572 x 20150123 (prer
0020 656c6561 73652900 4f626675 73636174 elease).Obfuscate
0030 6f722d20 636c616e 67207665 7273696f or- clang versio
0040 6e20332e 352e3020 28746167 732f5245 n 3.5.0 (tags/RE
0050 4c454153 455f3335 302f6669 6e616c29 LEASE_350/final)
0060 20286261 73656420 6f6e204c 4c564d20 (based on LLVM
0070 332e352e 3073766e 2900416e 64726f69 3.5.0svn).Android
0080 6420636c 616e6720 76657273 696f6e20 d clang version
0090 332e382e 32353632 32392020 28626173 3.8.256229 (bas
00a0 6564206f 6e204c4c 564d2033 2e382e32 ed on LLVM 3.8.2
00b0 35363232 392900 56229) .
```

The result for the 64bit version contains the same reference to the version of the used compiler.

```
$ objdump -s --section .comment libsqliteY.so
```

```
libsqliteY.so:      file format elf64-little
```

```
Contents of section .comment:
```

```
0000 4743433a 2028474e 55292034 2e392e78 GCC: (GNU) 4.9.x
0010 20323031 35303132 33202870 72657265 20150123 (prere
0020 6c656173 6529004f 62667573 6361746f lease).Obfuscato
0030 722d2063 6c616e67 20766572 73696f6e r- clang version
0040 20332e35 2e302028 74616773 2f52454c 3.5.0 (tags/REL
0050 45415345 5f333530 2f66696e 616c2920 EASE_350/final)
0060 28626173 6564206f 6e204c4c 564d2033 (based on LLVM 3
0070 2e352e30 73766e29 00416e64 726f6964 .5.0svn).Android
0080 20636c61 6e672076 65727369 6f6e2033 clang version 3
0090 2e382e32 35363232 39202028 62617365 .8.256229 (base
00a0 64206f6e 204c4c56 4d20332e 382e3235 d on LLVM 3.8.25
00b0 36323239 2900 6229) .
```

It can be seen that Android clang version 3.8.256229 has been used in the development.

Android clang version 3.8.256229 is based on LLVM 3.8.256229.

LLVM version 3.8.0 was released in March 2016, so version 3.8.256229 cannot have been deployed in 2015 (see Figure 11 - LLVM v3.8 release a LLVM v3.8 release announcement)

[llvm-announce] LLVM 3.8 Release

Hans Wennborg via **llvm-announce** [llvm-announce at lists.llvm.org](mailto:llvm-announce@lists.llvm.org)
Tue Mar 8 10:37:38 PST 2016

• **Messages sorted by:** [\[date\]](#) [\[thread\]](#) [\[subject\]](#) [\[author\]](#)

It is my pleasure to announce that LLVM 3.8.0 is now available!

Get it here: <http://www.llvm.org/releases/download.html#3.8.0>

This release contains the work of the LLVM community over the past six months: deprecated autoconf build, shrink-wrapping on by default, overhauled MSVC-compatible exception handling, updated Kaleidoscope tutorial, emutls, OpenMP supported by default, as well as improved optimizations, many bug fixes, and more.

Release notes for more details:
<http://llvm.org/releases/3.8.0/docs/ReleaseNotes.html>
<http://llvm.org/releases/3.8.0/tools/clang/docs/ReleaseNotes.html>

Huge thanks to everyone who helped with testing, bug fixing, packaging, and getting the release into a good state!

Special thanks to the volunteer release builders and testers, without whom there would be no releases: Dimitry Andric, Brian Cain, Ismail Donmez, Renato Golin, Sylvestre Ledru, Elias Pipping, Ben Pope, Daniel Sanders, and Nikola Smiljanic!

If you have any questions or comments about the release, please contact the community on the mailing lists. Onward to 3.9!

- Hans

(LLVM 3.7.1 Release Announcement:
<http://lists.llvm.org/pipermail/llvm-announce/2016-January/000066.html>)

• **Messages sorted by:** [\[date\]](#) [\[thread\]](#) [\[subject\]](#) [\[author\]](#)

[More information about the llvm-announce mailing list](#)

Figure 11 - LLVM v3.8 release a LLVM v3.8 release announcement

B. Configuration of all samples examined in this analysis

For complete documentation, the configurations of all samples examined in the context of this analysis are listed below. To improve readability, the display is in landscape format. The samples are sorted by the date of their first public discovery.

The telephone numbers and IP addresses contained in the configurations may, under certain circumstances, allow conclusions to be drawn about the manufacturer or customer, provided that reliable information about the subscribers can be obtained. However, a simple conclusion based on a country code or the geo-location of an IP address is not allowed, as both the registration of IP addresses and the switching of telephone connections can be achieved worldwide with little effort using a variety of concealment options.

Configuration of the sample 2e96e343ac10f5d9ace680e456c083e4eceb108f7209aa1e849f11a239e7a682

TlvTypeMobileEncryption	= b'\xff\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "again" (13)
TlvTypeMobileTargetHeartbeatInterval	= 86400 (12)
TlvTypeMobileTargetPositioning	= b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy	= "80.95.253.44" (20)
TlvTypeConfigTargetPort	= 80 (12)
TlvTypeConfigTargetPort	= 443 (12)
TlvTypeConfigTargetPort	= 4111 (12)
TlvTypeConfigTargetPort	= 22 (12)
TlvTypeConfigTargetPort	= 53 (12)
TlvTypeConfigSMSPhoneNumber	= "+420725988592" (21)
TlvTypeConfigCallPhoneNumber	= "+420725988592" (21)
TlvTypeMobileTrojanID	= "again" (13)
TlvTypeMobileTrojanUID	= b'\x9d*\x0f' (12)
TlvTypeUserID	= 1010 (12)
TlvTypeTrojanMaxInfections	= 3 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Feb 2 01:00:00 2012 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 48 (12)
TlvTypeMobileTargetHeartbeatEvents	= 191 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeInstalledModules	= Logging: Off Spy Call: On Call Interception: On SMS: On Address Book: Off
Tracking: On Phone Logs: On (140)	
TlvTypeMobileTrackingConfigRaw (61)	= b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,'
TlvTypeMobileTrackingConfig	= b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,' (53)
TlvTypeMobileTrackingDistance	= 1000 (12)

Configuration of the sample 0d798ca0b2d0ea9bad251125973d8800ad3043e51d4cc6d0d57b971a97d3af2d

TlvTypeMobileEncryption	= b'\xfe\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "JHANUK" (14)
TlvTypeMobileTargetHeartbeatInterval	= 600 (12)
TlvTypeMobileTargetPositioning	= b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy	= "212.56.102.38" (21)
TlvTypeConfigTargetPort	= 22 (12)
TlvTypeConfigTargetPort	= 53 (12)
TlvTypeConfigTargetPort	= 443 (12)
TlvTypeConfigTargetPort	= 8080 (12)
TlvTypeConfigSMSPhoneNumber	= "+447902513419" (21)
TlvTypeConfigCallPhoneNumber	= "+447747441129" (21)
TlvTypeMobileTrojanID	= "JHANUK" (14)
TlvTypeMobileTrojanUID	= b'\x05\xaa\x0f\x00' (12)
TlvTypeUserID	= 1000 (12)
TlvTypeTrojanMaxInfections	= 25 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 168 (12)
TlvTypeMobileTargetHeartbeatEvents	= 33021 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeMobileTargetLocationChangedRange	= 5 (9)
TlvTypeInstalledModules	= Logging: Off Spy Call: On Call Interception: Off SMS: On Address Book: On
Tracking: On Phone Logs: On (140)	
TlvTypeMobileTrackingConfigRaw	= b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,' (61)
TlvTypeMobileTrackingConfig	= b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,' (53)
TlvTypeMobileTrackingDistance	= 1000 (12)

Configuration of the sample 72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537

TlvTypeMobileEncryption	= b'\x19\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "Andriod" (15)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)
TlvTypeMobileTargetPositioning	= b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy	= "demo-01.gamma-international.de" (38)
TlvTypeConfigTargetPort	= 1111 (12)
TlvTypeConfigTargetPort	= 1112 (12)
TlvTypeConfigTargetPort	= 1113 (12)
TlvTypeConfigSMSPhoneNumber	= "+491726662364" (21)
TlvTypeConfigCallPhoneNumber	= "+4989549989890" (22)
TlvTypeConfigCallPhoneNumber	= "+6597294704" (19)
TlvTypeMobileTrojanID	= "Andriod" (15)
TlvTypeMobileTrojanUID	= b'\x81tc\x0f' (12)
TlvTypeUserID	= 1011 (12)
TlvTypeTrojanMaxInfections	= 10 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 168 (12)
TlvTypeMobileTargetHeartbeatEvents	= 4349 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeInstalledModules	= Logging: Off Spy Call: On Call Interception: On SMS: On Address Book: On
Tracking: On Phone Logs: On	(140)
TlvTypeMobileTrackingConfigRaw	= b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00X'
(61)	
TlvTypeMobileTrackingConfig	= b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00X' (53)
TlvTypeMobileTrackingDistance	= 1000 (12)

Configuration of the sample 363172a2f2b228c7b00b614178e4ffa00a3a124200ceef4e6d7edb25a4696345

TlvTypeMobileEncryption	= b'\xf9\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "derise" (14)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)
TlvTypeMobileTargetPositioning	= b'\x82\x87\x81\x86\x83' (13)
TlvTypeConfigTargetProxy	= "183.91.2.199" (20)
TlvTypeConfigTargetPort	= 9111 (12)
TlvTypeConfigTargetPort	= 9112 (12)
TlvTypeConfigTargetPort	= 9113 (12)
TlvTypeConfigSMSPhoneNumber	= "+841257725403" (21)
TlvTypeConfigCallPhoneNumber	= "08888" (13)
TlvTypeConfigCallPhoneNumber	= "+8408888" (16)
TlvTypeMobileTrojanID	= "derise" (14)
TlvTypeMobileTrojanUID	= b'\x820,\x00' (12)
TlvTypeUserID	= 1000 (12)
TlvTypeTrojanMaxInfections	= 3 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 168 (12)
TlvTypeMobileTargetHeartbeatEvents	= 189 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeInstalledModules	= Logging: Off Spy Call: Off Call Interception: Off SMS: On Address Book: Off
Tracking: On Phone Logs: On	(140)
TlvTypeMobileTrackingConfigRaw	= b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\x88\x13\x00\x00\x0c\x00\x00\x00@@E\x00X' (61)
TlvTypeMobileTrackingConfig	= b'\x0c\x00\x00\x00@AE\x00\x88\x13\x00\x00\x0c\x00\x00\x00@@E\x00X' (53)
TlvTypeMobileTrackingDistance	= 5000 (12)

Configuration of the sample 1935f2e52832df910edc1b7ef17b53d8c852fe66ec3afbe490ffc2ef057452b3

```

TlvTypeMobileEncryption                = b'\x7f\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig       = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID                  = "AKDEMO" (14)
TlvTypeMobileTargetHeartbeatInterval   = 60 (12)
TlvTypeMobileTargetPositioning         = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy                = "50.116.43.43" (20)
TlvTypeConfigTargetPort                = 80 (12)
TlvTypeConfigTargetPort                = 8080 (12)
TlvTypeConfigTargetPort                = 4343 (12)
TlvTypeConfigSMSPhoneNumber             = "+972312460121" (21)
TlvTypeConfigCallPhoneNumber            = "+974762113957" (21)
TlvTypeMobileTrojanID                  = "AKDEMO" (14)
TlvTypeMobileTrojanUID                 = b'\xf4\x8d\x91\x03' (12)
TlvTypeUserID                          = 1043 (12)
TlvTypeTrojanMaxInfections              = 999 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy      = 168 (12)
TlvTypeMobileTargetHeartbeatEvents     = 189 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules                = Logging: Off | Spy Call: On | Call Interception: Off | SMS: On | Address Book: On |
  Tracking: On | Phone Logs: On | (140)
TlvTypeMobileTrackingConfigRaw         =
  b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x
  00\x00\x00\x90d\x84\x00\x87\x86\x85\x81' (61)
TlvTypeMobileTrackingConfig           =
  b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x00\x00\x00\x90d\x84\x0
  0\x87\x86\x85\x81' (53)
TlvTypeMobileTrackingDistance          = 1000 (12)
TlvTypeMobileTrackingTimeInterval     = 300 (12)
TlvTypeMobileTargetPositioning         = b'\x87' (12)

```

Configuration of the sample 045161094b9f6b98c4ef87d2324f4bb8a0d0fbf58e349a37d11622aef2e6b051

```
TlvTypeMobileEncryption = b'\x81\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID = "ANDDemo" (15)
TlvTypeMobileTargetHeartbeatInterval = 300 (12)
TlvTypeMobileTargetPositioning = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy = "50.116.43.43" (20)
TlvTypeConfigTargetPort = 80 (12)
TlvTypeConfigTargetPort = 8080 (12)
TlvTypeConfigTargetPort = 4343 (12)
TlvTypeConfigSMSPhoneNumber = "+972312460121" (21)
TlvTypeConfigCallPhoneNumber = "+974762113957" (21)
TlvTypeMobileTrojanID = "ANDDemo" (15)
TlvTypeMobileTrojanUID = b'\x87\xa3B\x03' (12)
TlvTypeUserID = 1064 (12)
TlvTypeTrojanMaxInfections = 321 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 189 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules = Logging: Off | Spy Call: On | Call Interception: Off | SMS: On | Address Book: On |
Tracking: On | Phone Logs: On | (140)
TlvTypeMobileTrackingConfigRaw =
b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x
00\x00\x00\x90d\x84\x00\x87\x86\x85\x81' (61)
TlvTypeMobileTrackingConfig =
b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x00\x00\x00\x90d\x84\x0
0\x87\x86\x85\x81' (53)
TlvTypeMobileTrackingDistance = 1000 (12)
TlvTypeMobileTrackingTimeInterval = 300 (12)
TlvTypeMobileTargetPositioning = b'\x87' (12)
```

Configuration of the sample 84d39e5c6db75801a85cc5d2557ab536abe40496b23eba6b3e5c1722975d8f32

```
TlvTypeMobileEncryption          = b'\xf1\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID            = "428" (11)
TlvTypeMobileTargetHeartbeatInterval = 120 (12)
TlvTypeMobileTargetPositioning    = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy          = "blog.is-found.org" (25)
TlvTypeConfigTargetPort          = 1111 (12)
TlvTypeConfigTargetPort          = 1112 (12)
TlvTypeConfigTargetPort          = 1113 (12)
TlvTypeConfigSMSPhoneNumber       = "+491726652007" (21)
TlvTypeConfigCallPhoneNumber      = "+4989549989909" (22)
TlvTypeMobileTrojanID            = "428" (11)
TlvTypeMobileTrojanUID           = b'\n\xf2\x08\x01' (12)
TlvTypeUserID                    = 1003 (12)
TlvTypeTrojanMaxInfections        = 666 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 173 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules          = Logging: Off | Spy Call: On | Call Interception: Off | SMS: On | Address Book: On |
  Tracking: On | Phone Logs: On | (140)
TlvTypeMobileTrackingConfigRaw    =
  b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01' (61)
TlvTypeMobileTrackingConfig      = b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01' (53)
TlvTypeMobileTrackingDistance     = 1000 (12)
TlvTypeMobileTrackingTimeInterval = 44 (12)
```

Configuration of the sample 587b110da2ef9c59b18f01e97e9b12628f3e7b2e88611f7cb28a6efccb0aaba2

```

TlvTypeMobileEncryption                = b'\xa1\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig       = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID                  = "tmWoot" (14)
TlvTypeMobileTargetHeartbeatInterval   = 60 (12)
TlvTypeMobileTargetPositioning          = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy                = "69.164.211.41" (21)
TlvTypeConfigTargetPort                = 80 (12)
TlvTypeConfigTargetPort                = 443 (12)
TlvTypeConfigTargetPort                = 993 (12)
TlvTypeConfigTargetPort                = 995 (12)
TlvTypeConfigSMSPhoneNumber             = "+972368815537" (21)
TlvTypeConfigCallPhoneNumber            = "+972368881403" (21)
TlvTypeConfigCallPhoneNumber            = "+972366383884" (21)
TlvTypeMobileTrojanID                  = "tmWoot" (14)
TlvTypeMobileTrojanUID                  = b'y\xe13\x03' (12)
TlvTypeUserID                           = 1003 (12)
TlvTypeTrojanMaxInfections              = 99 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy       = 168 (12)
TlvTypeMobileTargetHeartbeatEvents      = 173 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules                 = Logging: Off | Spy Call: On | Call Interception: Off | SMS: On | Address Book: On |
    Tracking: On | Phone Logs: On | (140)
TlvTypeMobileTrackingConfigRaw          =
    b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x00\x00\x00\x90d\x84\x00\x87\x86\x85\x81' (61)
TlvTypeMobileTrackingConfig             =
    b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x00\x00\x00\x90d\x84\x00\x87\x86\x85\x81' (53)
TlvTypeMobileTrackingDistance            = 1000 (12)
TlvTypeMobileTrackingTimeInterval       = 300 (12)
TlvTypeMobileTargetPositioning           = b'\x87' (12)

```

Configuration of the sample abcb11c4787c62ab90cecc262f6fc98bd7f73335ac631c2e9aec8734088e91aa

```

TlvTypeMobileEncryption                = b'}\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig       = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID                  = "ANDR" (12)
TlvTypeMobileTargetHeartbeatInterval   = 60 (12)
TlvTypeMobileTargetPositioning          = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy                = "192.168.222.90" (22)
TlvTypeConfigTargetPort                 = 80 (12)
TlvTypeConfigTargetPort                 = 8080 (12)
TlvTypeConfigTargetPort                 = 4343 (12)
TlvTypeConfigSMSPhoneNumber             = "+972312460121" (21)
TlvTypeConfigCallPhoneNumber            = "+974762113957" (21)
TlvTypeMobileTrojanID                   = "ANDR" (12)
TlvTypeMobileTrojanUID                  = "]pB" (12)
TlvTypeUserID                           = 1043 (12)
TlvTypeTrojanMaxInfections              = 999 (12)
TlvTypeConfigMobileAutoRemovalDateTime  = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy       = 168 (12)
TlvTypeMobileTargetHeartbeatEvents      = 189 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules                 = Logging: Off | Spy Call: On | Call Interception: Off | SMS: On | Address Book: On |
  Tracking: On | Phone Logs: On | (140)
TlvTypeMobileTrackingConfigRaw          =
  b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x
  00\x00\x00\x90d\x84\x00\x87\x86\x85\x81' (61)
TlvTypeMobileTrackingConfig             =
  b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x00\x00\x00\x90d\x84\x0
  0\x87\x86\x85\x81' (53)
TlvTypeMobileTrackingDistance           = 1000 (12)
TlvTypeMobileTrackingTimeInterval       = 300 (12)
TlvTypeMobileTargetPositioning          = b'\x87' (12)

```

Configuration of the sample 2795e777d897857ab6fa19f85687a23a6071ab665299a47b8b952cb4e7056a07

```

TlvTypeMobileEncryption                = b'f\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig       = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID                  = "Android" (15)
TlvTypeMobileTargetHeartbeatInterval   = 60 (12)
TlvTypeMobileTargetPositioning          = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy                = "50.116.43.43" (20)
TlvTypeConfigTargetPort                 = 80 (12)
TlvTypeConfigTargetPort                 = 8080 (12)
TlvTypeConfigTargetPort                 = 4343 (12)
TlvTypeConfigSMSPhoneNumber             = "+972368810455" (21)
TlvTypeConfigCallPhoneNumber            = "+9747197747754" (22)
TlvTypeMobileTrojanID                   = "Android" (15)
TlvTypeMobileTrojanUID                  = b'\xadz\x03\x03' (12)
TlvTypeUserID                           = 1043 (12)
TlvTypeTrojanMaxInfections              = 334 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy       = 168 (12)
TlvTypeMobileTargetHeartbeatEvents      = 189 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules                 = Logging: Off | Spy Call: On | Call Interception: Off | SMS: On | Address Book: On |
  Tracking: On | Phone Logs: On | (140)
TlvTypeMobileTrackingConfigRaw          =
  b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x
  00\x00\x00\x90d\x84\x00\x87\x86\x85\x81' (61)
TlvTypeMobileTrackingConfig             =
  b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x00\x00\x00\x90d\x84\x0
  0\x87\x86\x85\x81' (53)
TlvTypeMobileTrackingDistance           = 1000 (12)
TlvTypeMobileTrackingTimeInterval       = 300 (12)
TlvTypeMobileTargetPositioning          = b'\x87' (12)

```

Configuration of the sample 704d599fe51e7a0f982438c13983fb936dd1530f659e8036bee69752221ef7d7

TlvTypeMobileEncryption	= b'\xee\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "ANDxJoe" (15)
TlvTypeMobileTargetHeartbeatInterval	= 120 (12)
TlvTypeMobileTargetPositioning	= b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy	= "blog.podzone.net" (24)
TlvTypeConfigTargetProxy	= "50.116.43.43" (20)
TlvTypeConfigTargetPort	= 80 (12)
TlvTypeConfigTargetPort	= 8080 (12)
TlvTypeConfigTargetPort	= 4343 (12)
TlvTypeConfigSMSPhoneNumber	= "+972368810455" (21)
TlvTypeConfigCallPhoneNumber	= "+9747197747754" (22)
TlvTypeMobileTrojanID	= "ANDxJoe" (15)
TlvTypeMobileTrojanUID	= "uH" (12)
TlvTypeUserID	= 1089 (12)
TlvTypeTrojanMaxInfections	= 66 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 168 (12)
TlvTypeMobileTargetHeartbeatEvents	= 24765 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeInstalledModules	= Logging: Off Spy Call: Off Call Interception: Off SMS: Off Address Book: Off
Tracking: Off Phone Logs: Off	(140)

Configuration of the sample 26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafba1

TlvTypeMobileEncryption	= b'\r\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "421and" (14)
TlvTypeMobileTargetHeartbeatInterval	= 120 (12)
TlvTypeMobileTargetPositioning	= b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy	= "qa01.gamma-international.de" (35)
TlvTypeConfigTargetPort	= 1111 (12)
TlvTypeConfigTargetPort	= 1112 (12)
TlvTypeConfigTargetPort	= 1113 (12)
TlvTypeConfigTargetPort	= 80 (12)
TlvTypeConfigSMSPhoneNumber	= "+491726652007" (21)
TlvTypeConfigCallPhoneNumber	= "+4989549989909" (22)
TlvTypeMobileTrojanID	= "421and" (14)
TlvTypeMobileTrojanUID	= b'J\x99\x8f\x00' (12)
TlvTypeUserID	= 1003 (12)
TlvTypeTrojanMaxInfections	= 999 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 168 (12)
TlvTypeMobileTargetHeartbeatEvents	= 4269 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeInstalledModules	= Logging: Off Spy Call: On Call Interception: On SMS: On Address Book: On
Tracking: On Phone Logs: On (140)	
TlvTypeMobileTrackingConfigRaw (61)	= b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,'
TlvTypeMobileTrackingConfig	= b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,' (53)
TlvTypeMobileTrackingDistance	= 1000 (12)

Configuration of the sample 1507ee069906d2a42216d77ef51d42a35efcc59b005b55d8ea771749057296db

TlvTypeMobileEncryption	= b'\xf4\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "trekki" (14)
TlvTypeMobileTargetHeartbeatInterval	= 120 (12)
TlvTypeMobileTargetPositioning	= b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy	= "10.0.0.153" (18)
TlvTypeConfigTargetPort	= 1337 (12)
TlvTypeConfigSMSPhoneNumber	= "+97260260260" (20)
TlvTypeConfigCallPhoneNumber	= "+97918918918" (20)
TlvTypeMobileTrojanID	= "trekki" (14)
TlvTypeMobileTrojanUID	= b'\x1f\xel\x15\x02' (12)
TlvTypeUserID	= 1002 (12)
TlvTypeTrojanMaxInfections	= 55 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 168 (12)
TlvTypeMobileTargetHeartbeatEvents	= 173 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeInstalledModules	= Logging: Off Spy Call: Off Call Interception: Off SMS: Off Address Book: Off Tracking: On Phone Logs: Off (140)
TlvTypeMobileTrackingConfigRaw	= b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x00\x00\x00\x90d\x84\x00\x87\x86\x85\x81' (61)
TlvTypeMobileTrackingConfig	= b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01\x00\x00\t\x00\x00\x000BE\x00\x00\x0c\x00\x00\x00\x90d\x84\x00\x87\x86\x85\x81' (53)
TlvTypeMobileTrackingDistance	= 1000 (12)
TlvTypeMobileTrackingTimeInterval	= 300 (12)
TlvTypeMobileTargetPositioning	= b'\x87' (12)
TlvTypeEncryption	= "5" (31)

Configuration of the sample 1ea335d1d5f99aeb1a516d6b267ba53c38438648874752eb0438edffffde380d

```

TlvTypeMobileEncryption          = b'\xf5\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID            = "zefix" (13)
TlvTypeMobileTargetHeartbeatInterval = 120 (12)
TlvTypeMobileTargetPositioning    = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy         = "blog.is-found.org" (25)
TlvTypeConfigTargetPort          = 1114 (12)
TlvTypeConfigTargetPort          = 1115 (12)
TlvTypeConfigTargetPort          = 1116 (12)
TlvTypeConfigSMSPhoneNumber      = "+491726650079" (21)
TlvTypeConfigCallPhoneNumber     = "+4989549989907" (22)
TlvTypeMobileTrojanID           = "zefix" (13)
TlvTypeMobileTrojanUID          = b'g\xcb-\x01' (12)
TlvTypeUserID                   = 1003 (12)
TlvTypeTrojanMaxInfections       = 555 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 173 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules         = Logging: Off | Spy Call: On | Call Interception: Off | SMS: On | Address Book: On |
  Tracking: On | Phone Logs: On | (140)
TlvTypeMobileTrackingConfigRaw   =
  b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01' (61)
TlvTypeMobileTrackingConfig      = b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,\x01' (53)
TlvTypeMobileTrackingDistance    = 1000 (12)
TlvTypeMobileTrackingTimeInterval = 44 (12)

```

Configuration of the sample 60dc08ab28db5ba3a56734097954861a525b4b384e8067e3eaac551c4cb8ece3

TlvTypeMobileEncryption	= b'\x03\x02\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "testAD" (14)
TlvTypeMobileTargetHeartbeatInterval	= 120 (12)
TlvTypeMobileTargetPositioning	= b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy	= "blog.is-found.org" (25)
TlvTypeConfigTargetPort	= 1114 (12)
TlvTypeConfigTargetPort	= 1115 (12)
TlvTypeConfigTargetPort	= 1116 (12)
TlvTypeConfigTargetPort	= 80 (12)
TlvTypeConfigSMSPhoneNumber	= "+491726650079" (21)
TlvTypeConfigCallPhoneNumber	= "+4989549989907" (22)
TlvTypeMobileTrojanID	= "testAD" (14)
TlvTypeMobileTrojanUID	= b'l\xel\x08\x01' (12)
TlvTypeUserID	= 1003 (12)
TlvTypeTrojanMaxInfections	= 666 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 168 (12)
TlvTypeMobileTargetHeartbeatEvents	= 4269 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeInstalledModules	= Logging: Off Spy Call: On Call Interception: On SMS: On Address Book: On
Tracking: On Phone Logs: On	(140)
TlvTypeMobileTrackingConfigRaw	= b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,' (61)
TlvTypeMobileTrackingConfig	= b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@@E\x00,' (53)
TlvTypeMobileTrackingDistance	= 1000 (12)

Configuration of the sample 84d231e6ea1e2e3283c3e9cbfcabeded0d7e5723852e378e0caf5bb001501938

```

TlvTypeMobileEncryption          = b'\xf3\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID            = "defs" (12)
TlvTypeMobileTargetHeartbeatInterval = 120 (12)
TlvTypeMobileTargetPositioning    = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy         = "blog.is-found.org" (25)
TlvTypeConfigTargetPort          = 1114 (12)
TlvTypeConfigTargetPort          = 1115 (12)
TlvTypeConfigTargetPort          = 1116 (12)
TlvTypeConfigSMSPhoneNumber      = "+491726650079" (21)
TlvTypeConfigCallPhoneNumber     = "+4989549989907" (22)
TlvTypeMobileTrojanID           = "defs" (12)
TlvTypeMobileTrojanUID           = b'\xdf\xee\x08\x01' (12)
TlvTypeUserID                    = 1003 (12)
TlvTypeTrojanMaxInfections       = 666 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 4269 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeInstalledModules          = Logging: Off | Spy Call: On | Call Interception: On | SMS: On | Address Book: On |
    Tracking: On | Phone Logs: On | (140)
TlvTypeMobileTrackingConfigRaw   =
    b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@E\x00,\x01' (61)
TlvTypeMobileTrackingConfig      = b'\x0c\x00\x00\x00@AE\x00\xe8\x03\x00\x00\x0c\x00\x00\x00@E\x00,\x01' (53)
TlvTypeMobileTrackingDistance    = 1000 (12)
TlvTypeMobileTrackingTimeInterval = 44 (12)

```

Configuration of the sample 46690ef267f21b5840377833e7af51b19fbd343bac97d6eb66b186d58ba3f9b3

```
TlvTypeMobileEncryption = b'\x16\x03\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig = b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID = "flash28" (15)
TlvTypeMobileTargetHeartbeatInterval = 43200 (12)
TlvTypeMobileTargetPositioning = b'\x82\x87\x86\x81\x83' (13)
TlvTypeConfigTargetProxy = "103.208.86.204" (22)
TlvTypeConfigTargetProxy = "marketconsulting.ddns.net" (33)
TlvTypeConfigTargetPort = 80 (12)
TlvTypeConfigTargetPort = 8080 (12)
TlvTypeConfigTargetPort = 443 (12)
TlvTypeConfigSMSPhoneNumber = "+97260260260" (20)
TlvTypeConfigCallPhoneNumber = "+97918918918" (20)
TlvTypeMobileTrojanID = "flash28" (15)
TlvTypeMobileTrojanUID = " r" (12)
TlvTypeUserID = 1015 (12)
TlvTypeTrojanMaxInfections = 10 (12)
TlvTypeConfigMobileAutoRemovalDateTime = Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy = 168 (12)
TlvTypeMobileTargetHeartbeatEvents = 173 (10)
TlvTypeMobileTargetHeartbeatRestrictions = b'\xc0\x00' (10)
TlvTypeMasterAgentUserPermission = b' \x00\x00\x00\xa0\xc8q\x00\x0c\x00\x00\x00@\xcaq\x00' [...]
TlvTypeMasterAgentUserPermissionValuePacket =
  b'\x0c\x00\x00\x00@\xcaq\x00\xeb\x03\x00\x00\x0c\x00\x00\x00@\xccq\x000\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xeb\x03\x00\x00' (12)
TlvTypeMasterAgentUserPermissionValueData= "O" (12)
TlvTypeMasterAgentUserPermissionValuePacket =
  b'\x0c\x00\x00\x00@\xcaq\x00\xec\x03\x00\x00\x0c\x00\x00\x00@\xccq\x000\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xec\x03\x00\x00' (12)
TlvTypeMasterAgentUserPermissionValueData= "O" (12)
TlvTypeMasterAgentUserPermissionValuePacket =
  b'\x0c\x00\x00\x00@\xcaq\x00\xed\x03\x00\x00\x0c\x00\x00\x00@\xccq\x000\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xed\x03\x00\x00' (12)
TlvTypeMasterAgentUserPermissionValueData= "O" (12)
TlvTypeMasterAgentUserPermissionValuePacket =
  b'\x0c\x00\x00\x00@\xcaq\x00\xee\x03\x00\x00\x0c\x00\x00\x00@\xccq\x000\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xee\x03\x00\x00' (12)
TlvTypeMasterAgentUserPermissionValueData= "O" (12)
TlvTypeMasterAgentUserPermissionValuePacket =
  b'\x0c\x00\x00\x00@\xcaq\x00\xef\x03\x00\x00\x0c\x00\x00\x00@\xccq\x000\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xef\x03\x00\x00' (12)
TlvTypeMasterAgentUserPermissionValueData= "O" (12)
```

```

TlvTypeMasterAgentUserPermissionValuePacket      =
  b'\x0c\x00\x00\x00@\xcaq\x00\xf1\x03\x00\x0c\x00\x00\x00@\xccq\x000\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xf1\x03\x00\x00' (12)
TlvTypeMasterAgentUserPermissionValueData= "0" (12)
TlvTypeMasterAgentUserPermissionValuePacket      =
  b'\x0c\x00\x00\x00@\xcaq\x00\xf2\x03\x00\x0c\x00\x00\x00@\xccq\x000\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xf2\x03\x00\x00' (12)
TlvTypeMasterAgentUserPermissionValueData= "0" (12)
TlvTypeMasterAgentUserPermissionValuePacket      =
  b'\x0c\x00\x00\x00@\xcaq\x00\xf4\x03\x00\x0c\x00\x00\x00@\xccq\x000\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xf4\x03\x00\x00' (12)
TlvTypeMasterAgentUserPermissionValueData= "0" (12)
TlvTypeMasterAgentUserPermissionValuePacket      = b'\x0c\x00\x00\x00@\xcaq\x00\xf6\x03\x00\x0c\x00\x00\x00' (32)
TlvTypeMasterAgentUserPermissionValueName= b'\xf6\x03\x00\x00' (12)
TlvTypeInstalledModules                         = Logging: Off | Spy Call: Off | Call Interception: Off | SMS: On | Address Book: On |
  Tracking: Off | Phone Logs: On | (140)

```

Configuration of the sample c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e

TlvTypeMobileEncryption	= b'\xfa\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "adalet" (14)
TlvTypeMobileTargetHeartbeatInterval	= 86400 (12)
TlvTypeMobileTargetPositioning	= b'\x86\x82\x87\x81\x83' (13)
TlvTypeConfigTargetProxy	= "94.23.165.112" (21)
TlvTypeConfigTargetPort	= 443 (12)
TlvTypeConfigTargetPort	= 80 (12)
TlvTypeConfigTargetPort	= 53 (12)
TlvTypeConfigTargetPort	= 8080 (12)
TlvTypeConfigTargetPort	= 9001 (12)
TlvTypeConfigTargetPort	= 9050 (12)
TlvTypeConfigTargetPort	= 9040 (12)
TlvTypeConfigSMSPhoneNumber	= "+97260260260" (20)
TlvTypeConfigCallPhoneNumber	= "+97918918918" (20)
TlvTypeMobileTrojanID	= "adalet" (14)
TlvTypeMobileTrojanUID	= "<V" (12)
TlvTypeUserID	= 1002 (12)
TlvTypeTrojanMaxInfections	= 999 (12)
TlvTypeConfigMobileAutoRemovalDateTime	= Thu Jan 1 01:00:00 1970 (12)
TlvTypeConfigAutoRemovalIfNoProxy	= 168 (12)
TlvTypeMobileTargetHeartbeatEvents	= 173 (10)
TlvTypeMobileTargetHeartbeatRestrictions	= b'\xc0\x00' (10)
TlvTypeInstalledModules	= Logging: Off Spy Call: Off Call Interception: Off SMS: Off Address Book: Off Tracking: Off Phone Logs: Off (140)

Configuration of the sample 77b4d11e369ac5dec4e951e5879248c1c9a84d756c06d89875f113e4c6469464

TlvTypeMobileEncryption	= b'\xd8\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "cleaner" (15)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)

Configuration of the sample 31fa1129d8e682a90913cc28b4e5d6b064131c93a6d86118d94f93918ed6e2f8

TlvTypeMobileEncryption	= b'\xd8\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "whistel" (15)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)

Configuration of the sample 241c38fd3cafc37f496fb7e1872924f21bf1263e17a81d03981dd29b531e4623

TlvTypeMobileEncryption	= b'\x88\x03\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "network" (15)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)

Configuration of the sample d8f6abc6cb1388da6b2870f06d52036a435407d6bf2c0b43684fd72edc4a9e77

TlvTypeMobileEncryption	= b'\x82\x03\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "Disk" (12)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)

Configuration of the sample aa299745edf2e55531c9a8304b57f9bee8f37a4c3f4be56260bad096c7ea1c03

TlvTypeMobileEncryption	= b'\xd8\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "FunVoic" (15)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)

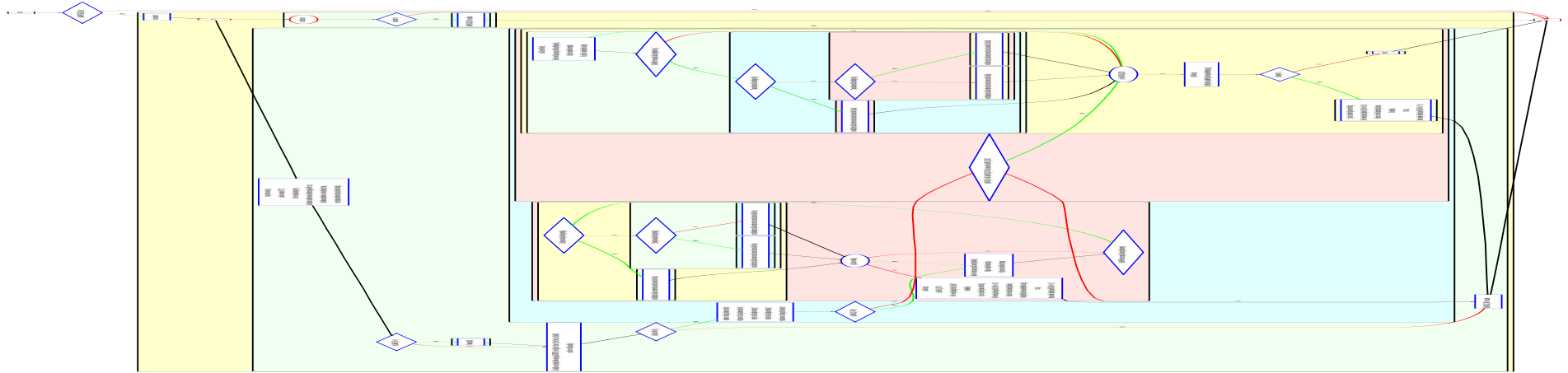
Configuration of the sample 3f8baeae01980e77fa905216e291b6478105295c8372a003d73e9086b0b3e964

TlvTypeMobileEncryption	= b'\xfc\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "Diary" (13)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)

Configuration of the sample ff8aaf49f4377e6ee162f1f0778f98e33dd2a8df2d96de6ba766851ee436467e

TlvTypeMobileEncryption	= b'\xd8\x01\x00\x00\xa03\x84\x00\x0c\x00\x00\x00P\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig	= b'\x0c\x00\x00\x00P\x13\xfe\x00\x00\x00\x00\x00\x10\x00\x00\x00' [...]
TlvTypeMobileTargetID	= "myphone" (15)
TlvTypeMobileTargetHeartbeatInterval	= 60 (12)

C. Sample 421and: ControlFlow com.android.services.CallLogs.run()



D. Sample adalet: ControlFlow org.customer.fu.e.a.run()

