

Evolution einer privatwirtschaftlichen Schadsoftware für staatliche Akteure

FinFisher FinSpy für Android 2012-2019

Thorsten Schröder, Linus Neumann

Chaos Computer Club

Version 1.0, 28. Dezember 2019

Abstract

Der Chaos Computer Club hat mehrere Schadsoftware-Samples hinsichtlich a) Herkunft und b) Erstellungsdatum analysiert.

Im Rahmen des vorliegenden Berichts wurden insgesamt 28 Exemplare („Samples“) von Spionagesoftware aus den Jahren 2012 bis 2019 untersucht. Ergebnis der Analyse ist, dass diese Samples demselben Hersteller zugeordnet werden können:

- A) Alle Samples nutzen denselben proprietären Mechanismus zum Provisionieren der Schadsoftware durch den Endkunden: Die fallspezifische Konfiguration wird über den identischen *Covert Channel* im Android APK versteckt.
- B) Alle aus A) extrahierbaren Konfigurationen liegen in einem Binärformat vor, dessen Strukturen untereinander gleiche Muster aufweisen.
- C) Neben dem identischen Provisionierungsmechanismus der Samples über alle Generationen hinweg finden sich große Ähnlichkeiten des Java-Programmcodes zwischen den Samples aus den Jahren 2014 und 2016.
- D) Ein Sample aus dem Jahr 2012 und mehrere Samples aus dem Jahr 2014 sind eindeutig der Firma Gamma International Deutschland bzw. FinFisher zuzuordnen.
- E) Aufgrund der Syntax und Wahl der Variablen- und Funktionsnamen im Java-Programmcodes kann davon ausgegangen werden, dass die Software von deutschsprachigen Entwicklern hergestellt wurde.
- F) Aufgrund von Metadaten in den Shared-Object-Dateien eines von der *Gesellschaft für Freiheitsrechte* eingereichten Samples kann zweifelsfrei belegt werden, dass dieses Sample frühestens im Jahr 2016 hergestellt wurde.

Inhalt

| | |
|---|-----------|
| Abstract | 2 |
| Inhalt | 3 |
| Einleitung | 4 |
| Untersuchungsgegenstand | 4 |
| Fragestellung | 5 |
| Methodik | 7 |
| A. Feststellung des Herstellungszeitpunktes | 7 |
| 1. Beim Erstellen verwendete Versionen von Software und Bibliotheken | 7 |
| 2. Timestamps in Zertifikaten | 7 |
| 3. Timestamps in Konfigurationen und Logfiles | 8 |
| 4. Öffentliche Dokumentation | 8 |
| B. Feststellung der Herkunft | 8 |
| 1. Verwendete Zertifikate | 8 |
| 2. Übereinstimmung proprietärer Routinen | 9 |
| 3. Benennung von Funktionen und Variablen | 10 |
| 4. Bezüge zu Samples bekannter Herkunft | 10 |
| Ergebnisse | 13 |
| A. Herstellungszeitpunkt des „adalet“-Samples | 13 |
| Beim Erstellen verwendete Versionen von Software und Bibliotheken | 13 |
| Timestamps in Zertifikaten | 14 |
| Timestamps in Konfigurationen und Logfiles | 15 |
| B. Herkunft der Samples | 16 |
| Verwendete Zertifikate | 16 |
| Übereinstimmung proprietärer Routinen | 18 |
| Bezüge zu Samples bekannter Herkunft | 29 |
| Fazit | 32 |
| A. Feststellung des Herstellungszeitpunktes | 32 |
| 1) Wann wurde das „adalet“-Sample produziert und eingesetzt? | 32 |
| 2) Liegt der Zeitpunkt bzw. Zeitraum vor oder nach dem 18. Juli 2015? | 32 |
| B. Feststellung der Herkunft | 32 |
| 1) Stammen die Samples aus unterschiedlichen Quellen, oder gibt es eindeutige Hinweise auf eine gemeinsame Urheberschaft? | 32 |
| 2) Können die Urheber der Samples identifiziert werden? | 32 |
| Öffentliche Dokumentation von Untersuchungsgegenständen und -methoden | 33 |
| Appendix | 34 |
| A. Veröffentlichungszeitpunkt der SQLite-Version 3.13.0 | 34 |
| B. Konfiguration sämtlicher im Rahmen dieser Analyse untersuchten Samples | 37 |
| C. Sample 421and: ControlFlow com.android.services.CallLogs.run() | 59 |
| D. Sample adalet: ControlFlow org.customer.fu.e.a.run() | 60 |

Einleitung

Auf Bitte der Gesellschaft für Freiheitsrechte¹ hat der Chaos Computer Club 28 verschiedene Software-Applikationen („Samples“) für das Betriebssystem Android untersucht. Anlass für die Untersuchung ist der Verdacht des Verstoßes gegen Exportkontrollvorschriften gemäß § 18 Abs. 2 Nr. 1 und Abs. 5 Nr. 1 des Außenwirtschaftsgesetzes durch die FinFisher GmbH, FinFisher Labs GmbH und Elaman GmbH („FinFisher“²). Die GFF hat in diesem Zusammenhang Strafanzeige erstattet.³

Besondere Bedeutung kommt in diesem Zusammenhang einem Sample zu, welches nach Angaben der GFF im Jahr 2017 in der Türkei gegen politische Oppositionelle eingesetzt wurde. Im Rahmen der vorliegenden Analyse konnte bewiesen werden, dass dieses Schadsoftware-Sample frühestens im Jahr 2016 erstellt wurde. Es konnte weiterhin belegt werden, dass dieses Sample aus der Schadsoftware-Familie „FinSpy“ stammt.

Untersuchungsgegenstand

Der Begriff „Sample“ bezeichnet im Folgenden ein konkretes Schadsoftware-Paket. Jedes Sample ist durch eine eindeutige und einmalige SHA256-Prüfsumme referenzierbar. Dieser Bericht berücksichtigt ausschließlich Android-Varianten der Schadsoftware-Samples.

Ausgangspunkt der Untersuchung war zunächst ein Schadsoftware-Sample mit der Prüfsumme `c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e`. Diese Schadsoftware wurde nach Angaben der GFF über die Domain `adaleticinuru.com` zum Download angeboten. Diese Website hat eine von der türkischen Oppositionsbewegung zu Organisationszwecken genutzte Website nachgeahmt; mutmaßlich, um entsprechende Nutzerinnen anzulocken und zu täuschen. Das Sample wird im Folgenden als das „adalet“-Sample bezeichnet.

Schwerpunkt der vorliegenden Untersuchung sind Gemeinsamkeiten und Unterschiede des „adalet“-Samples zu anderen Samples bekannter, vermuteter oder unbekannter Herkunft. Die hierzu herangezogenen Samples stammen aus dem Zeitraum 2012–2019. Dem CCC liegen aus öffentlichen und nicht-öffentlichen Quellen gegenwärtig 26 Schadsoftware-Samples vor, die strukturelle Ähnlichkeiten zum „adalet“-Sample aufweisen und im Rahmen der vorliegenden Analyse untersucht wurden. Ein weiteres Sample dient der Installation einer Schadsoftware, die wiederum den anderen Samples entspricht. Insgesamt wurden somit 28 Samples analysiert.

¹ „Die Gesellschaft für Freiheitsrechte e. V. (GFF) ist ein gemeinnütziger Verein mit Sitz in Berlin, der 2015 gegründet wurde. Sie verfolgt das Ziel, mit strategischer Klageführung den Erhalt und den Ausbau der Grund- und Menschenrechte zu erreichen.“

https://de.wikipedia.org/wiki/Gesellschaft_für_Freiheitsrechte zuletzt abgerufen am 19. Dezember 2019

² Die Schadsoftware-Familie FinFisher / FinSpy wird – mutmaßlich zur Verschleierung der Geschäftspraktiken – unter verschiedenen Namen von verschiedenen GmbHs und internationalen Unternehmen vertrieben, die hohe personelle und örtliche Überlappungen aufweisen. Zur Vereinfachung verwenden die Autoren im Folgenden den Begriff „Firmengruppe FinFisher“ zur Bezeichnung dieses Konglomerats.

³ GFF: „Export von Überwachungssoftware“ <https://freiheitsrechte.org/export-von-uberwachungssoftware/> zuletzt abgerufen am 19. Dezember 2019

Fragestellung

Folgende Fragestellungen waren die Grundlage der Analyse:

A. Herstellungszeitraum

- 1) Wann sind die Samples produziert und eingesetzt worden?
- 2) Liegt der Zeitpunkt bzw. Zeitraum vor oder nach dem 18. Juli 2015⁴?

B. Herkunft

- 1) Stammen die Samples aus unterschiedlichen Quellen oder gibt es eindeutige Hinweise auf eine gemeinsame Urheberschaft?
- 2) Ist es möglich, die Urheber der Samples zu identifizieren?

Hierfür sind Gemeinsamkeiten und Unterschiede der einzelnen Samples über einen Zeitraum von sieben Jahren analysiert und dokumentiert worden. Tabelle 1 zeigt die in der Untersuchung berücksichtigten Android-Samples. Samples, die sich zweifelsfrei der Firmengruppe FinFisher zuordnen lassen⁵, sind farblich hervorgehoben. Das "adalet"-Sample, welches den Untersuchungsschwerpunkt bildet, ist grün markiert.

⁴ Zeitpunkt der Aufnahme sogenannter *Intrusion-Software*, in die Außenwirtschaftsverordnung (AWV), um damit die Vorgaben der EU-Dual-Use-Verordnung seit dem 1. Januar 2015 in nationales Recht umzusetzen. Seit diesem Tag ist *Intrusion Software*, die in der EU hergestellt und außerhalb der EU verkauft werden soll, genehmigungspflichtig.

⁵ Die betreffenden Samples stammen aus einem Leak aus dem Jahr 2014. Siehe hierzu Absatz 4. *Bezüge zu Samples bekannter Herkunft*, Seite 10

Tabelle 1. Übersicht der im Rahmen dieser Analyse untersuchten Samples

| 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 |
| 1ea335d1d5f99aebela516d6b267ba53c38438648874752eb0438edffde380d | 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 |

Methodik

Das folgende Kapitel erläutert die angewendeten Untersuchungsmethoden und ihre unterschiedliche Bedeutung bzw. Beweiskraft.

A. Feststellung des Herstellungszeitpunktes

Wir haben die uns vorliegenden Samples dekompiert⁶ und auf Hinweise untersucht, die Rückschlüsse auf Zeitpunkt oder Zeitraum des Kompilierens zulassen. Beim Kompilieren wird das Programm fertig gestellt, gebündelt und einsatzbereit gemacht. Der Zeitpunkt des Kompilierens markiert daher die Erstellung bzw. Fertigstellung des Samples und damit seinen *Herstellungszeitpunkt*. Sämtliche Programmierarbeiten müssen vor diesem Zeitpunkt abgeschlossen worden sein. Änderungen am Programmcode führen dazu, dass die Software neu kompiliert werden muss.

Hinweise auf den Herstellungszeitpunkt und -zeitraum konnten dabei insbesondere durch folgende Attribute erlangt werden:

1. Beim Erstellen verwendete Versionen von Software und Bibliotheken

Bibliotheken sind Hilfsmodule, die häufig genutzte Funktionalitäten bereitstellen. Programme binden Bibliotheken ein, um die von diesen bereitgestellten Funktionalitäten im Programm nutzen zu können, ohne sie erneut programmieren zu müssen. So muss „das Rad nicht neu erfunden“ werden.

Die Versionsnummer einer Bibliothek oder eines zur Herstellung verwendeten Programmes kann ein Indiz für den Zeitpunkt sein, *nach* dem die Herstellung eines Samples erfolgt ist, weil zwar das Verwenden veralteter Software und Bibliotheken, nicht jedoch das Verwenden von Software oder Bibliotheken aus der Zukunft möglich ist.

2. Timestamps in Zertifikaten

Alle Android-Applikationen müssen vom Hersteller digital signiert werden⁷. Die Signatur sichert Authentizität und Integrität (Freiheit von nachträglichen Veränderungen) der Applikation durch kryptographische Operationen. Jegliche, auch minimale, Änderungen der Applikation führen dazu, dass die Prüfung der Signatur fehlschlägt, sodass die Applikation nicht mehr ausführbar ist. Diese Maßnahme dient dem Schutz vor jeglicher nachträglicher Manipulation der Applikation durch Dritte.

Die Signatur wird mithilfe sogenannter Zertifikate erstellt, welche aus einem öffentlichen und einem privaten Teil bestehen⁸. Der öffentliche Teil liegt der Applikation bei und dient der Überprüfung der Signatur. Der private Teil des Zertifikates verbleibt beim Hersteller und dient dem Erstellen der Signatur.

Die für das Signieren der Applikation verwendeten Zertifikate haben einen Gültigkeitszeitraum, dessen Beginn und Dauer prinzipiell beliebig gewählt werden können. Der Beginn des Gültigkeitszeitraums des Zertifikates ist ein vergleichsweise schwaches Indiz für den ersten Einsatzzeitpunkt der Applikation, da das Betriebssystem Android unter bestimmten Bedingungen auch Software ausführen kann, die mit erst in der Zukunft gültigen Zertifikaten signiert ist.

⁶ Unter Dekompilieren versteht das Rückübersetzen von Maschinen- oder Bytecode in menschenlesbaren Programmcode. Android-Applikationen wie die vorliegenden Samples sind ein komprimiertes Archiv standardisierter Struktur, in dem sich die unterschiedlichen Teile der Applikation – zum Beispiel Programmcode, Bilder, oder Daten – befinden. Durch das „Dekompilieren“ wird dieses Archiv wieder als Ordnerstruktur zugänglich und eine tiefergehende Analyse des Samples möglich.

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 Für eine ausführliche Beschreibung des Konzeptes siehe https://de.wikipedia.org/wiki/Digitale_Signatur

3. Timestamps in Konfigurationen und Logfiles

Der Zeitpunkt des Kompilierens wird in verschiedenen Dateien vermerkt, die Teil der Applikation sind. Diese *Timestamps* haben ebenfalls den Charakter von Indizien, da sie mit überschaubarem Aufwand gefälscht werden könnten, wenngleich Anlass oder Motivation dazu fraglich wären. Dies gilt insbesondere im Hinblick auf das Ziel der Untersuchung, den *frühesten* Zeitpunkt der Erstellung festzustellen. Um für diese Feststellung eine falsche Fährte zu legen, müsste die Applikation bei ihrer Kompilierung gezielt „in die Zukunft“ manipuliert werden, was wiederum bis zu diesem in der Zukunft liegenden Tag auffällig wäre.

4. Öffentliche Dokumentation

Der früheste öffentlich dokumentierte Zeitpunkt des Einsatzes kann anhand öffentlich verfügbarer Informationen auf Viren-Analyse-Plattformen wie *VirusTotal*⁹ oder durch die *Wayback Machine*¹⁰ aus öffentlichen Quellen recherchiert werden. Liegt dieser Zeitpunkt nach einem gegebenen Datum, so wird dadurch nur der spätere Einsatz der Software, nicht jedoch der Zeitpunkt ihrer Herstellung zweifelsfrei belegt.

B. Feststellung der Herkunft

Aufgrund wirtschaftlicher Interessen und möglicher juristischer Konsequenzen ist von einer geringen Motivation des Herstellers auszugehen, seine Urheberschaft an einer Schadsoftware öffentlich geltend zu machen oder öffentlich ohne Not einzugestehen.

Die Analyse folgt daher einem *bottom-up*-Ansatz, bei dem Gemeinsamkeiten unterschiedlicher Samples systematisch untersucht werden: Hinweise für die Herkunft und Urheberschaft der Schadsoftware müssen aus Ähnlichkeiten und Unterschieden zu anderen Samples abgeleitet werden, deren Herkunft zweifelsfrei bekannt ist. Die Alternativ-Erklärung einer Kopie oder absichtlichen Täuschung durch Dritte muss dabei grundsätzlich immer in Betracht gezogen werden. Erst wenn diese hinreichend unplausibel ist, kann von Indiz- oder Beweiskraft ausgegangen werden.

Weist also das fragliche „*adalet*“-Sample zu einem Sample einwandfrei geklärt Herkunft eine eindeutige, nicht durch Zufall erklärbare Ähnlichkeit auf, ist dies als starkes Indiz dafür zu werten, dass auch das „*adalet*“-Sample dem identifizierten Urheber zuzuordnen ist.

Im Rahmen der vorliegenden Analyse konnten Hinweise auf Herkunft und Urheberschaft insbesondere durch folgende Attribute erlangt werden:

1. Verwendete Zertifikate

Die Bedeutung von *Code-Signing*-Zertifikaten wird in Absatz A-2. *Timestamps in Zertifikaten* auf Seite 7 erläutert. Auch bei der Feststellung der Herkunft eines Samples kommt diesen Zertifikaten eine besondere Bedeutung zu.

Das Konstruieren, also Fälschen oder „Nachbauen“ eines modernen Zertifikates wäre aufgrund kryptographischer Absicherung zum heutigen Zeitpunkt unmöglich oder nur unter enorm hohem Zeit-, Energie- und Ressourcenaufwand möglich. Sofern der private Teil des Zertifikates nicht weitergegeben,

⁹ *VirusTotal* ist vom Unternehmen *Google Inc.* betriebener kostenloser Online-Dienst, um einzelne Dateien durch über 70 verschiedene Antivirenprogramme und Schadsoftware-Scanner analysieren zu lassen. Der Dienst ist unter <https://www.virustotal.com/> erreichbar und eine der wohl größten Analyseplattformen für Schadsoftware. Zu jeder analysierten Datei wird dort festgehalten, zu welchem Datum sie erstmalig analysiert wurde.

¹⁰ Die *Wayback Machine* ist Projekt des gemeinnützigen *Internet Archive*. Mit der sogenannten *Wayback Machine* können archivierte Websites betrachtet werden, die in der Zwischenzeit ggf. verändert oder entfernt wurden. Die *Wayback Machine* gibt jeweils den Zeitpunkt des Abrufs mit an.

entwendet oder veröffentlicht wurde, stammen Applikationen, die mit dem gleichen Zertifikat signiert sind, aus der gleichen Quelle¹¹.

2. Übereinstimmung proprietärer Routinen

Proprietäre Routinen bezeichnen – im Gegensatz zu quelloffenen Software-Bibliotheken, die bei vielen unterschiedlichen Software-Projekten diverser Hersteller routinemäßig zum Einsatz kommen¹² – jene Teile des Programmcodes, die vom Hersteller genuin selbst erdacht, konzipiert und programmiert wurden, und darüber hinaus der Öffentlichkeit nicht als Quellcode, sondern nur als kompiliertes Programm zur Verfügung gestellt wurden.

Da diese Routinen nicht als Quelltext der Öffentlichkeit vorliegen, ist eine Anwendung in weiten Teilen „wortgleicher“ Routinen in von unabhängigen Dritten geschriebener Software in jedem Fall erklärungsbedürftig, ab einem gewissen Ausmaß der Komplexität jedoch unplausibel: Grundsätzlich ist es zwar möglich, dass unabhängige und unbekannte Dritte eine Untersuchung des Samples durchgeführt und dabei Funktionalitäten für eine eigenen, unabhängigen Software-Stamm kopiert haben. Mit zunehmender Komplexität und gegenseitigen Abhängigkeiten zu Dritt-Komponenten wird diese Hypothese jedoch unplausibel.

Der Grad der Unplausibilität ist dabei abhängig von folgenden Faktoren:

a) Aufwand einer Re-Implementierung

Software-Routinen, die nicht nur lokal ausführbar sind, sondern für Ihre Funktion auch noch weitere externe Systeme oder Programme voraussetzen, würden bei einer Kopie durch Dritte den Aufwand verlangen, auch die unbekannte technische Gegenseite zu implementieren.

Der Aufwand einer Re-Implementierung der unbekanntenen Komponente würde in der Regel schnell den Aufwand übersteigen, der beim Kopieren der bekannten Komponente gespart worden wäre. Proprietäre Routinen, die weitere Abhängigkeiten – insbesondere von einer ebenfalls proprietären Gegenstelle – haben, deuten daher mit an Sicherheit grenzender Wahrscheinlichkeit auf gleiche Urheberschaft.

b) Weiterentwicklung

Für das Kopieren proprietärer Routinen durch Dritte kommen primär zwei Motivationen in Frage:

(a) Aufwand- und Kostenersparnis und

(b) Irreführung potenzieller Analysen zur Feststellung des Herstellers.

Eine technische Weiterentwicklung der kopierten Routinen würde beiden Motivationen entgegenlaufen und ist darüber ein Beweis für fundamentale Kenntnis der technischen Zusammenhänge, ohne die eine Weiterentwicklung nicht möglich ist.

Eine sukzessive und konsistente Weiterentwicklung verwendeter Software-Routinen über verschiedene chronologische Samples lässt sich daher als starker Hinweis auf gleiche Urheberschaft werten.

c) Verschleierungstechniken

Software-Hersteller versuchen sich mit verschiedenen Methoden und aus verschiedenen Gründen gegen das Dekompilieren Ihrer Produkte technisch zur Wehr zu setzen. Dies geschieht einerseits, um Extraktion und Kopien zu verhindern, im Fall von Schadsoftware jedoch insbesondere auch, um die schädlichen Funktionalitäten der Software zu verschleiern um somit ein Erkennen (bspw. durch Schadsoftware-Scanner) zu erschweren. Der Prozess der Verschleierung wird als *Obfuscation*¹³ bezeichnet.

¹¹ Der Begriff „Quelle“ kann im Falle kommerzieller Spionage-Software sowohl den Hersteller als auch dessen Kunden bezeichnen.

¹² siehe hierzu A-1. Beim Erstellen verwendete Versionen von Software und Bibliotheken, Seite 4.

¹³ Für eine Begriffsklärung siehe [https://de.wikipedia.org/wiki/Obfuscation_\(Software\)](https://de.wikipedia.org/wiki/Obfuscation_(Software))

Die Anwendung von Verschleierungstechniken hat einen starken Einfluss auf den Aufwand, der zum Kopieren und Transferieren von fremden Code-Routinen notwendig ist und somit entsprechende Konsequenzen für den a) Aufwand einer Re-Implementierung oder gar einer b) Weiterentwicklung des extrahierten Codes.

Darüber hinaus handelt es sich bei *Obfuscation* um technische „Einbahnstraßen“, die zum Zeitpunkt der Kompilation ein komplexeres Kompilat erzeugen, als normalerweise aus dem Quelltext resultieren würde. Die Verwendung unterschiedlicher Methoden der *Obfuscation* führt zu sehr unterschiedlichem Kompilaten. Verwendete Methoden der *Obfuscation* erlauben daher ebenfalls Rückschlüsse auf die Herkunft: Die *Obfuscation*-Methode ist ihrerseits ein Merkmal der Software.

Programmierer können aus einer Vielzahl verfügbarer freier oder proprietärer *Obfuscation*-Methoden wählen, oder individuelle eigene Methoden der *Obfuscation* entwickeln und anwenden. Die Verwendung einer individuell entwickelten *Obfuscation* kann als Beweis für gleiche Herkunft und Urheberschaft gelten, da die ursprüngliche *Obfuscation*-Routine Dritten nicht zugänglich ist oder war.

3. Benennung von Funktionen und Variablen

Programmcode wird üblicherweise in Modulen und *Funktionen*¹⁴ strukturiert. Funktionen decken Teilfunktionalitäten des Programmes ab und werden üblicherweise gemäß ihrer Funktionalität benannt, um den Quellcode für die Programmierenden leserlich und verständlich zu halten.

Dem Ziel der Leserlichkeit dient auch das Postulieren und Einhalten von *Coding Conventions*¹⁵ innerhalb eines Software-Projektes. Diese bilden eine Art „Schreibstil“ innerhalb eines Teams oder eines Unternehmens und umfassen neben vielen anderen Aspekten der Programmierung in der Regel auch Vorgaben hinsichtlich

- Strukturierung des Codes
- Verwendung von Pattern
- Namenskonventionen für Funktionen und Variablen
- Verwendung von Code Obfuscation
- Modularisierung

Gleichen sich zwei Samples hinsichtlich dieser Attribute in hohem Ausmaß, so ist dies ein starkes Indiz dafür, dass die Samples vom gleichen Programmier-Team stammen.

4. Bezüge zu Samples bekannter Herkunft

Sämtliche bisher in Abschnitt B. *Feststellung der Herkunft* beschriebenen Methoden sind nur geeignet zur Analyse verschiedener Samples auf Ähnlichkeiten und daher limitiert auf die Beantwortung der Untersuchungsfrage, ob die Samples aus gleicher oder unterschiedlichen Quellen stammen .

Hinreichende Belege für eine gleiche Urheberschaft vorausgesetzt, kann die *Attribution* auf einen spezifischen Hersteller anhand von Samples bekannter Herkunft erfolgen: Wenn die Urheberschaft eines Samples einwandfrei geklärt ist, kann über die oben beschriebenen Methoden die Wahrscheinlichkeit festgestellt werden, mit der ein davon unterschiedliches Sample aus der gleichen Quelle stammt.

Im August 2014 wurde bekannt, dass das Unternehmen *Gamma International*, welches Teil der Firmengruppe FinFisher ist, von einem Hacker kompromittiert worden war. In der Folge dieses Angriffs

¹⁴ Für eine Begriffsklärung siehe [https://de.wikipedia.org/wiki/Funktion_\(Programmierung\)](https://de.wikipedia.org/wiki/Funktion_(Programmierung))

¹⁵ Für eine Begriffsklärung siehe <https://de.wikipedia.org/wiki/Programmierstil>

wurden insgesamt über 40GB an extrahierten Daten veröffentlicht.¹⁶ Teil der Veröffentlichung¹⁷ waren Quelltexte von unterschiedlichen Schadsoftware-Produkten aus der Firmengruppe FinFisher, sowie Werbematerialien und interne Informationen. Unter dem Material fand sich auch eine Aufstellung von Computern, die per Trojaner von Bahrain aus überwacht wurden, sowie Support-Anfragen von Kunden des Schadsoftware-Herstellers¹⁸.

Unter den veröffentlichten Daten befinden sich mehrere Samples, welche somit zweifelsfrei der Firmengruppe FinFisher zugeordnet werden können. Abbildung 1 zeigt beispielhaft einen Screenshot der Datei im *Torrent Inspector* der Software *Transmission*, welche zum Herunterladen sogenannter *torrents* verwendet wird. Die SHA256-Prüfsumme dieses zweifelsfrei zugeordneten Samples ist 26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafba1.

Analog haben wir die SHA256-Prüfsummen sämtlicher in diesem Leak veröffentlichten APK-Samples bestimmt. Für diese Samples gilt: Sie stammen zweifelsfrei aus der Firmengruppe FinFisher und sind in Tabelle 1 entsprechend hervorgehoben.

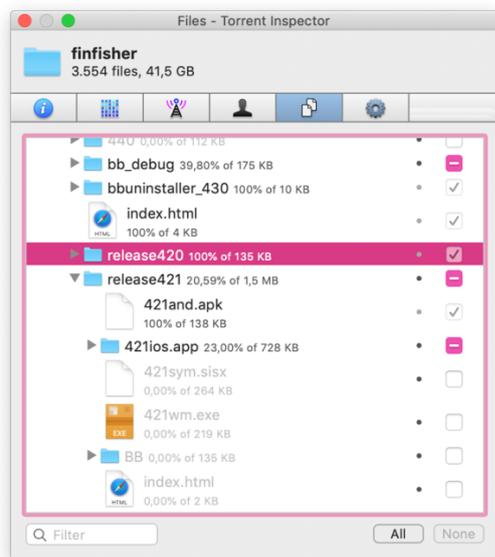


Abbildung 1. Ursprung des FinSpy 421 Samples: Leak von 2014

¹⁶ Netzpolitik.org; Meister, Andre (2014): *Gamma FinFisher gehackt: Werbe-Videos von Exploits und Quelltext von FinFly Web veröffentlicht* <https://netzpolitik.org/2014/gamma-finfisher-gehackt-werbe-videos-von-exploits-und-quelltext-von-finfly-web-veroeffentlicht/> zuletzt aufgerufen am 19. Dezember 2019.

¹⁷ Der gesamte Datensatz ist als Torrent öffentlich verfügbar: <https://www.dropbox.com/s/n7xch2vqc9p5x3e/finfisher.torrent?dl=1> zuletzt abgerufen am 19. Dezember 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>

¹⁸ Heise online; Detlef Borchers (2014): *FinFisher-Hack zeigt Überwachung von Oppositionellen* <https://www.heise.de/newsticker/meldung/FinFisher-Hack-zeigt-Ueberwachung-von-Oppositionellen-2289532.html>, zuletzt abgerufen am 19. Dezember 2019.

```
# 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  
1ea335d1d5f99aebela516d6b267ba53c38438648874752eb0438edffde380d  
  ./qateam/ta/430/zefix.apk  
1507ee069906d2a42216d77ef51d42a35efcc59b005b55d8ea771749057296db  
  ./qateam/ta/438/trekki.apk  
60dc08ab28db5ba3a56734097954861a525b4b384e8067e3eaac551c4cb8ece3  
  ./qateam/ta/428/testAD.apk  
84d39e5c6db75801a85cc5d2557ab536abe40496b23eba6b3e5c1722975d8f32  
  ./qateam/ta/428/428.apk  
84d231e6ea1e2e3283c3e9cbfcabeded0d7e5723852e378e0caf5bb001501938  
  ./qateam/ta/428/defs.apk  
26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafba1  
  ./qateam/ta/release421/421and.apk
```

Diese Samples zweifelsfrei geklärter Herkunft dienen als Referenz für die Analyse von Unterschieden, Ähnlichkeiten und Weiterentwicklungen der anderen vorliegenden Samples von nicht zweifelsfrei geklärter Herkunft. Unterschiede – sofern nicht als Weiterentwicklungen erklärbar – werden als Indizien gegen eine gemeinsame Urheberschaft gewertet. Ähnlichkeiten und Weiterentwicklungen werden hingegen als Indizien für eine gemeinsame Urheberschaft gewertet.

Ergebnisse

A. Herstellungszeitpunkt des „adalet“-Samples

Zum Herstellungsdatum des „adalet“-Samples wurden bereits mehrere unabhängige Gutachten veröffentlicht¹⁹. Über die schon erbrachten Belege hinaus konnten noch weitere Hinweise gesammelt werden. Diese werden im Folgenden zusammen mit der Replikation ausgewählter bereits bekannter Befunde zusammengefasst.

Beim Erstellen verwendete Versionen von Software und Bibliotheken

a) *SQLite-Version 3.13.0*

Das „adalet“-Sample verfügt in den Resources des APK über zwei native Code-Bibliotheken `library.so` und `libsqliteY.so`. Die Bibliothek `library.so` exportiert die in Abbildung 2 dargestellten Funktionen.

| 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> | 000000000006DEDC | |
|  <code>Java_com_esn_wal_audio_ogg_VorbisFileInputStream_skipStreamIdx</code> | 000000000006E2F4 | |
|  <code>Java_org_customer_fu_aud10_o1g1g_Aud10FileOutputStream_closeStreamIdx</code> | 000000000006DBA8 | |
|  JNI_OnLoad | 00000000001C314 | |
|  <code>Java_org_customer_fu_aud10_o1g1g_Aud10FileOutputStream_SoundlnByte</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] |

Abbildung 2. Von der nativen Bibliothek `library.so` exportierte Routinen

Diese Bibliothek wird von der Java-Anwendung geladen und kann über das *Java Native Interface (JNI)* verwendet werden. Das JNI erlaubt es Java-Entwicklern, Code und Implementierungen anderer Programmiersprachen und Compiler zu nutzen.

SQLite ist eine quelloffene, frei verfügbare Datenbank-Software. Sie wird in Form einer Software-Bibliothek mit Anwendungen ausgeliefert. Auf Android-Systemen hat eine solche Bibliothek die Dateiendung `.so` für „*Shared Object*“. Die verwendete SQLite-Version 3.13.0 wurde am 18. Mai 2016 veröffentlicht (Siehe Appendix A Veröffentlichungszeitpunkt der SQLite-Version 3.13.0).

Es ist somit **bewiesen**, dass das „adalet“-Sample **frühestens am 18. Mai 2016** hergestellt wurde.

¹⁹ Siehe insbesondere:

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, zuletzt abgerufen am 19. Dezember 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>, zuletzt abgerufen am 19. Dezember 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>, zuletzt abgerufen am 19. Dezember 2019

Timestamps in Zertifikaten

Das zur Signatur des „adalet“-Samples verwendete Zertifikat kann beispielsweise mit Hilfe von `openssl`²⁰ oder `keytool`²¹ untersucht werden. Die Gültigkeit des Zertifikates beginnt am 10. Oktober 2016. Dieser Zeitpunkt kennzeichnet üblicherweise den Zeitpunkt der Erstellung des Zertifikates.

Das „adalet“-Sample ist mit einem **erst ab dem 10. Oktober 2016** gültigen Zertifikat signiert. Mit hoher Wahrscheinlichkeit ist davon auszugehen, dass es **nach diesem Zeitpunkt hergestellt** wurde. Dieser Befund findet bereits im technischen Appendix zur Strafanzeige der GFF²² Erwähnung und konnte im Rahmen der vorliegenden Analyse repliziert werden.

```
$ 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               ....
]
]
```

Auszug 1. Beginn der Gültigkeit des zur Signatur des „adalet“-Samples verwendeten Zertifikats

²⁰ <https://www.openssl.org/>, zuletzt abgerufen am 19. Dezember 2019

²¹ <https://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html>, zuletzt abgerufen am 19. Dezember 2019

²² https://netzpolitik.org/2019/wir-stellen-strafanzeige-zollkriminalamt-ermittelt-gegen-finisher-wegen-illegalem-export-des-staatstrojaners/#2019-07-05_Strafanzeige-FinFisher-Tuerkei_Anhang-Technik_A_Konfigurationsoptionen abgerufen am 25. September 2019.

Timestamps in Konfigurationen und Logfiles

Die Datei `resources/build-data.properties` des „adalet“-Samples beinhaltet Hinweise auf den genauen Zeitpunkt der Kompilierung der Komponente `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
```

Auszug 2. Build Properties des „adalet“-Samples

Die Property `build.time` deutet darauf hin, dass Teile des „adalet“-Samples **am 23. September 2016** erstellt wurden. Dieser Befund findet bereits im technischen Appendix zur Strafanzeige der GFF Erwähnung und konnte im Rahmen der vorliegenden Analyse repliziert werden.

B. Herkunft der Samples

Verwendete Zertifikate

Einige der im Rahmen der vorliegenden Analyse untersuchten Samples wurden unter Nutzung desselben Zertifikats signiert. Tabelle 2 gibt einen Überblick und fasst Hashes, target ID, Fingerabdruck des Signaturzertifikats und dessen Ausstellungsdatum zusammen.

Die Herkunft des Samples *421and* ist zweifelsfrei geklärt. Die Samples *JHANUK*, *Andriod* und *derise* wurden mit demselben Zertifikat signiert und können mittels dieser Methode ebenfalls eindeutig der Firmengruppe FinFisher zugeordnet werden und sind entsprechend in Tabelle 2 farblich hervorgehoben.

Die Samples *JHANUK*, *Andriod* und *derise* stammen auch von der Firmengruppe FinFisher.

Die Samples *AKDemo*, *ANDDemo*, *428*, *tmWoot*, *ANDR*, *Android*, *ANDxJoe*, *trekki*, *zefix*, *testAD* und *defs* wurden im gleichen Jahr mit einem anderen Zertifikat als die vorgenannten Samples signiert. Sie stammen alle aus derselben Quelle wie das Sample *421and* und können anhand des Leaks eindeutig der Firmengruppe FinFisher zugeordnet werden.

Die Samples *AKDemo*, *ANDDemo*, *428*, *tmWoot*, *ANDR*, *Android*, *ANDxJoe*, *421and*, *trekk*, *zefix*, *testAD* und *defs* stammen von der Firmengruppe FinFisher.

Die Samples *flash28* `46690ef267f21b5840377833e7af51b19fbd343bac97d6eb66b186d58ba3f9b3` und *adalet* weisen ebenfalls ein gemeinsames Signaturzertifikat auf uns stammen aus derselben Quelle.

Die verbleibenden Samples wurden jeweils mit individuellen Zertifikaten signiert. Aus Sicht des Herstellers wäre dies eine nachvollziehbare Reaktion auf die Veröffentlichung des Leaks 2014.

Tabelle 2. Übersicht der zur Signatur der unterschiedlichen Samples verwendeten Zertifikate

| 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 |
| 72a522d0d3dcd0c026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537 | 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 | 42land | 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 |
| 1ea335d1d5f99aebela516d6b267ba53c38438648874752eb0438edffde380d | 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 |
| 84d231e6eale2e3283c3e9cbfcabed0d7e5723852e378e0caf5bb001501938 | 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 |
| c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e | 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 |

Übereinstimmung proprietärer Routinen

a) *Provisionierung*

These: Alle relevanten Samples nutzen **denselben proprietären Mechanismus** zur Provisionierung der Schadsoftware durch den Endkunden: Die Fall-spezifische Konfiguration wird über *Covert Channel*²³ im Android APK versteckt. Alle extrahierbaren Konfigurationen liegen in einem Binärformat vor, dessen Strukturen untereinander gleiche Muster aufweisen.

Begriffsklärung: Der Vorgang der Provisionierung bezeichnet die Einrichtung von Nutzerrechten und die entsprechende Bereitstellung von Verbindungen, Diensten, Anwendungen und Speicherplatz. Im Falle einer Schadsoftware vom Typ „Trojanisches Pferd“ betrifft die Provisionierung beispielsweise die kundenspezifischen Einstellungen der *Command-and-Control*-Infrastruktur, welche den Rückkanal für extrahierte Informationen darstellt. Für den hypothetischen Fall eines Einsatzes der Schadsoftware durch staatliche Stellen beispielsweise in Ägypten²⁴ und Deutschland wäre es zum Beispiel üblich, für beide Kunden der Firmengruppe unterschiedlich konfigurierte Samples und unterschiedliche Infrastrukturen unterhalten: Die Samples der beiden Kunden wären unterschiedlich *provisioniert*.

Da sich in der Provisionierung sensible Informationen über den Auftraggeber verbergen, ist es nachvollziehbar, dass Maßnahmen zur Verschleierung ergriffen werden, damit im Falle einer Entdeckung nicht ohne weiteres auf die Herkunft geschlossen werden kann. Je spezifischer und komplexer diese Verschleierung stattfindet, desto unwahrscheinlicher werden zufällige Ähnlichkeiten zwischen Samples unterschiedlicher Hersteller.

Gleichzeitig sind Hersteller von kommerzieller Schadsoftware gezwungen, individuelle Mechanismen der Verschleierung zu verwenden, um einer einfachen Entdeckung oder automatisierten Detektion zu entgehen. Im Fall der vorliegenden Samples wird ein sogenannter *Covert Channel* verwendet.

Beweis: Die vorliegenden Samples aus den Jahren 2012 bis 2019 besitzen alle eine markante Ähnlichkeit zu einem 2012 veröffentlichten, augenscheinlich der Firma Gamma International zuzuschreibenden Demo-Trojaner²⁵ (siehe Abbildung 3) und einem weiteren Sample²⁶ aus dem gleichen Jahr, welches über eine vietnamesische IP-Adresse veröffentlicht wurde. Dieses Samples wurden von zahlreichen Stellen analysiert, beispielsweise von den *Trustwave Spiderlabs*²⁷. Derselbe Mechanismus wird darüber hinaus im Sample *421and* verwendet, das zweifelsfrei dem gleichen Hersteller zugeordnet werden kann, da es aus dem bereits beschriebenen Daten-Leck der Firmengruppe FinFisher stammt.

²³ Ein *covert channel* bezeichnet einen parasitären Kommunikationskanal, welcher Bandbreite (überschüssige Informationskapazität) eines legitimierten Kommunikationskanals benutzt, um Informationen zu übermitteln. Siehe https://de.wikipedia.org/wiki/Verdeckter_Kanal, zuletzt abgerufen am 19. Dezember 2019

²⁴ Für andere hypothetische Beispiele siehe Bill 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/>, zuletzt abgerufen am 19. Dezember 2019

²⁵ 72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537

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/> zuletzt abgerufen am 19. Dezember 2019

Um einen Trojaner mit den passenden Parametern konfigurieren zu können, setzen die untersuchten Samples auf Metadaten in Datei-Attributen in den Android-APK-Dateien. Der Ansatz ähnelt dem einer Steganografie²⁸.

Ein APK-File ist aus technischer Sicht ein ZIP-Archiv, in dem mehrere Dateien gebündelt sind. Die Spezifikation sieht für jede Datei in einem APK jeweils 6 Zeichen vor, in denen Dateiattribute kodiert werden können. Die Dateien selbst können dabei auch leer sein.

Zur Realisierung des *Covert Channels* werden genau diese Freifelder für Dateiattribute genutzt. Zum Auslesen der Konfiguration werden die Dateiattribute bestimmter Dateien innerhalb des ZIP-Archivs ausgelesen, aneinandergehängt und schließlich als Konfiguration interpretiert.

Das Einrichten des Covert Channels geschieht durch ein gezieltes Setzen der fraglichen Attribute beim Provisionieren des APK-Paketes. Der Hersteller nutzt hierfür vermutlich ein eigenes Tool, welches nicht öffentlich verfügbar ist.

Im Falle eines Samples aus 2014²⁹, welches zweifelsfrei der Firmengruppe FinFisher zugeordnet werden kann (siehe 4. *Bezüge zu Samples bekannter Herkunft*, Seite 10), wird die insgesamt 534 Zeichen lange Konfiguration in den Meta-Daten von mindestens 89 leeren Dateien versteckt. Im Falle der Samples von 2016 (z.B. *adalet*³⁰) sind es 86 Dateien für eine 515 Zeichen lange Konfiguration.

Der Mechanismus zum Verstecken der Daten ist charakteristisch für Schadsoftware der Firmengruppe FinFisher. Der gleiche Versteckmechanismus wird in den vorliegenden Samples von 2012 bis 2019 verwendet. Der CCC analysierte neun weitere Samples, die erstmals im Frühjahr 2019 öffentlich gemacht worden sind. Dass über einen Zeitraum von mindestens sieben Jahren der gleiche Mechanismus zum Verstecken bzw. Extrahieren der Konfigurationsdaten verwendet wird, ist ein starkes Indiz dafür, dass alle analysierten Samples vom gleichen Hersteller stammen. Um diesen Verdacht zu erhärten, führten die Autoren des vorliegenden Berichts darüber hinaus eine Analyse der Nutzdaten durch, die in den verschiedenen Samples auf diese Weise versteckt wurden.

Weitergehende Analyse der Provisionierungen

Die Nutzdaten eines Demo-Samples der Firmengruppe FinFisher sind in Abbildung 3 illustriert. Teil der Konfiguration sind eine deutsche Mobilfunk- und Festnetznummer, sowie eine Telefonnummer mit der Landesvorwahl von Singapur. Die in der Konfiguration verwendeten Hostnamen (`demo-01.gamma-international.de`) sowie die Münchener Festnetznummer erhärten den Verdacht, dass es sich um eine Konfiguration aus der Firmengruppe FinFisher handelt.

Weitere Samples aus dem Jahr 2012 liegen vor. Das dargestellte Demo-Sample, welches im Folgenden gemäß seiner Target ID als *Andriod* [sic!] bezeichnet wird, wurde 2012 an *VirusTotal* zur Analyse übermittelt – dort klassifiziert ein großer Teil der Antiviren-Firmen dieses Sample als FinFisher- bzw. FinSpy-Trojaner (siehe Abbildung 4).

²⁸ „Steganografie ist die Kunst oder Wissenschaft der verborgenen Speicherung oder Übermittlung von Informationen in einem Trägermedium“ siehe <https://de.wikipedia.org/wiki/Steganographie>, zuletzt abgerufen am 19. Dezember 2019

²⁹ 26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafb1

³⁰ c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e

```

TlvTypeMobileEncryption = b'\x19\x02\x00\x00\xa3\x84\x00\x0c\x00\x00\x13\xfe\x00' [...]
TlvTypeMobileTargetOfflineConfig = b'\x0c\x00\x00\x00P\x13\xfe\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 |
TlvTypeMobileTrackingConfigRaw = b'5\x00\x00\x00\xa03E\x00\x0c\x00\x00\x00@AE\x00\xe8\x03\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'
TlvTypeMobileTrackingDistance = 1000 (12)

```

Abbildung 3. Konfiguration eines Demo-Samples

| DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY |
|-------------------------|---------|---------------------------------|------------------|--|
| AegisLab | | FinFisher_1 | AhnLab-V3 | Android-Trojan/FinSpy.5919b |
| Alibaba | | Monitor.Android/FinSpy.dafbe56e | Arcabit | Android.Monitor.FinSpy.B |
| Avast | | Android.CardServ-AU [Trj] | Avast-Mobile | Android.FinFisher-A [Trj] |
| AVG | | Android.CardServ-AU [Trj] | Avira (no cloud) | ANDROID/Agent.AFV.Gen |
| BitDefender | | Android.Monitor.FinSpy.B | CAT-QuickHeal | Android.FinSpy.A (PUP) |
| ClamAV | | Andr.Malware.Agent-1687991 | Comodo | Malware@#1jxdb4dfvxvux |
| Cyren | | AndroidOS/FinSpy.A | DrWeb | Android.Finspy.origin |
| Emsisoft | | Android.Monitor.FinSpy.B (B) | ESET-NOD32 | Android/Belesak.A |
| F-Prot | | AndroidOS/FinSpy.A | FireEye | Android.Monitor.FinSpy.B |
| Fortinet | | Android/FinSpy.Altr | GData | Android.Trojan.FinSpy.A |
| Ikarus | | Trojan.AndroidOS.Belesak | K7AntiVirus | Trojan (0001140e1) |
| K7GW | | Trojan (0001140e1) | Kaspersky | Not-a-virus.HEUR.Monitor.AndroidOS.Fi... |
| MAX | | Malware (ai Score=100) | McAfee | ArtemisI08CFFA8F55BE |
| McAfee-GW-Edition | | ArtemisPUP | Microsoft | PUA.Win32/Bitrepeyp.B |
| NANO-Antivirus | | Trojan.Android.Finspy.bdoxek | Qihoo-360 | Trojan.Android.Gen |
| Sophos AV | | Andri/FinSpy-A | Symantec | Trojan.Gen.MBT |
| Symantec Mobile Insight | | Backdoor.Finfish | Tencent | Privacy.Android.Finspy.a |

Abbildung 4. Klassifikation des Samples „Andriod“ als FinFisher/FinSpy durch gängige Antiviren-Detektoren³¹

³¹ Analyse des Samples 72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537 <https://www.virustotal.com/gui/file/72a522d0d3dcd0dc026b02ab9535e87a9f5664bc5587fd33bb4a48094bce0537/detection>, zuletzt abgerufen am 19. Dezember 2019

Im *Andriod*-Demo-Sample von 2012 verbirgt sich nach Zusammenfügen der Metadaten der leeren Dateien folgende Konfiguration, die sich mittels Ausgabe Tool `hexdump` schon erkennen lässt.

```

$ 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 |..|

```

Die frühestens 2016, spätestens jedoch im Januar 2017 (*flash28*³²) und Juli 2017 (*adalet*³³) und darüber hinaus im März und Mai 2019 aufgetretenen FinSpy-Varianten für Android verwenden das gleiche Verfahren, um die fallbasierte Konfiguration während einer Provisionierung durch den Endkunden im Android-APK zu speichern.

Die einzelnen Konfigurationsdaten von 2012 bis 2019 weisen in ihrer Beschaffenheit eine Ähnlichkeit auf, die keineswegs zufälliger Natur sein kann. Alle Varianten verstecken Die Konfigurationen in den Central Directory Structure (CDS) Blöcken der APK PK-Zip Datei³⁴ als *Internal* und *External File Attribute*, base64-kodiert.

In jeder CDS-Sektion können somit 6 Byte an base64-kodierten Daten versteckt werden. Die Schadsoftware extrahiert sämtliche dieser versteckten Blöcke aus dem APK, hängt die Zeichen aneinander, und dekodiert das Ergebnis mit dem üblichen Algorithmus „base64“.

Sämtliche Konfigurationen liegen in diesem proprietären Binärformat vor. Aus den einfachen Hex-Dumps dieser Binärdaten wird ersichtlich, dass alle Beispiele in ihrer Binärstruktur eine grundlegende Ähnlichkeit aufweisen. Zum Extrahieren der Konfigurations-Daten haben wir das 2012 auf der Open Source Code

³² 46690ef267f21b5840377833e7af51b19fbd343bac97d6eb66b186d58ba3f9b3

³³ c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e

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

Plattform *Github* veröffentlichte Schadsoftwareanalyse-Werkzeug³⁵ `extractConfig.rb` modifiziert, erweitert, neu programmiert und ebenfalls veröffentlicht³⁶, um auch aus neueren Samples Konfigurationen extrahieren zu können. Mit Hilfe der von uns ebenfalls auf *Github* veröffentlichten Werkzeuge³⁷ lassen sich u. A. die Konfigurationen der Samples *Andriod*, *derise* sowie *421and* auslesen und menschenlesbar darstellen:

Das „Andriod“-Sample enthält eine Konfiguration mit deutscher Mobilfunk- und Münchner Festnetz-Nummer, sowie einer Domain, die der Firmengruppe *FinFisher* zugeordnet werden kann. Am Standort München haben mehrere Teile der Firmengruppe *FinFisher* Ihren Sitz. Weiterhin wurde im Jahr 2013 öffentlich dokumentiert, dass die Firmengruppe *FinFisher* „Entwicklerbüros in Obersendling in München“ unterhält oder unterhalten hat³⁸. Diese Indizien stehen in Übereinstimmung mit der schon in Abschnitt 1. *Verwendete Zertifikate* auf Seite 8 dokumentierten Beweis, dass das Sample *Andriod* der Firmengruppe *FinFisher* zuzuordnen ist.

```
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\x0c\x00\x00\x00@E\x00X' (53)
TlvTypeMobileTrackingDistance = 1000 (12)
```

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

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>, zuletzt abgerufen am 19. Dezember 2019.

Das *derise*-Sample von 2012 verfügt über eine Konfiguration mit einer IP-Adresse, die Vietnam³⁹ zugeordnet wird und eine Telefonnummer mit vietnamesischer Landesvorwahl:

```
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\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)
```

Das Sample *421and* von 2014 verfügt ebenfalls über eine Konfiguration mit deutscher Mobilfunk- und Münchner Festnetz-Nummer, sowie einer Domain, die erneut der Firmengruppe FinFisher zugeordnet werden kann:

```
[...]
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> zuletzt abgerufen am 19. Dezember 2019 verortet diese IP-Adresse in Vietnam bei der CMC Telecom Infrastructure Company (cmctelecom.vn)

Das „adalet“-Sample (nach 2016) verfügt über eine Konfiguration mit einer der Bundesrepublik Deutschland zugeordneten IP-Adresse⁴⁰, sowie eine Telefonnummer mit der Landesvorwahl⁴¹ Israels.

```
[...]
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)
```

Das *flash28*-Sample (nach 2016) verfügt über mit einer gezielt unverdächtigen Domain (marketconsulting.ddns.net), sowie Telefonnummern mit der Landesvorwahl Israels.

```
[...]
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 diesem Abschnitt werden nur die für die Beweisführung relevanten Samples behandelt. Eine vollständige Dokumentation der Konfiguration aller untersuchten Samples findet sich in *Appendix B Konfiguration sämtlicher im Rahmen dieser Analyse untersuchten Samples*.

⁴⁰ <https://ipinfo.io/94.23.165.112> zuletzt abgerufen am 19. Dezember 2019, verortet diese IP-Adresse bei der OVH GmbH in Saarbrücken, Saarland, Germany

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

Die vollständige Konfiguration der ab dem Frühjahr 2019 veröffentlichten Samples wie beispielsweise

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

kann mit den bestehenden Werkzeugen noch nicht vollständig extrahiert werden. Die Konfigurationsdaten lassen sich zwar wie beschrieben aus der APK-Datei extrahieren, ein Interpretieren der Konfigurationsparameter erfordert jedoch eine weitergehende Analyse der jeweiligen Samples.

Es ist davon auszugehen, dass im Rahmen der Weiterentwicklung der Schadsoftware neue Konfigurationsparameter eingeführt wurden, die von den bisher bereitstehenden Werkzeugen noch nicht berücksichtigt werden. Wir beabsichtigen, die von uns veröffentlichte Analyse-Software⁴² anzupassen, sobald die entsprechenden Parameter bekannt sind.

Ungeachtet dessen ist jedoch bei Betrachtung der Binärdaten die Ähnlichkeit zu den älteren Samples offensichtlich: Es handelt sich um eine Weiterentwicklung der über einen Zeitraum von sieben Jahren verwendeten Methode. Eine solche Weiterentwicklung ist naheliegend und plausibel. Sie kann als starkes Indiz gewertet werden, dass Samples, die dieses Verfahren verwenden, von dem gleichen Team entwickelt wurden.

Bewertung:

Die Analyse der Übereinstimmung proprietärer Routinen im Rahmen der Provisionierung hat zwei Primärbefunde als Ergebnis:

1. Sämtliche Varianten verwenden dieselbe Methode zum Verstecken der Konfigurationen in den Central Directory Structure (CDS) Blöcken der APK PK-Zip Datei⁴³ als *Internal* und *External File Attribute*, base64-kodiert.
2. Sämtliche auf diese Weise extrahierten Daten weisen dasselbe proprietäre Binärformat auf.

Die Erkenntnisse zum (1.) korrekten Auslesen der Konfigurationsdaten und (2.) der korrekten Dekodierung des proprietären Binärprotokolls konnten mittels *Reverse Engineering*⁴⁴ der vorliegenden Samples gewonnen werden.

Dies ist ein deutlicher Hinweis darauf, dass die Samples von 2012 bis 2019 vom gleichen Hersteller stammen.

Unter Annahme der Alternativhypothese, dass eines oder mehrere der Samples von einem anderen Herstellern stammen, müsste erklärt werden, wie die gleichen Methoden für *Covert Channel* und Kodierung zum Einsatz kommen, wenngleich die dafür benötigten Tools einem zweiten Hersteller nicht zur Verfügung stünden.

Zwar wäre es mittels *Reverse Engineering* möglich, die zum Erstellen dieser Provisionierungsdaten verwendeten Tools nachzuimplementieren, allerdings wäre es für einen dritten, unbeteiligten Hersteller mit weniger Aufwand verbunden, alternative eigene Methoden zu implementieren.

Das Sample *421and* stammt direkt aus dem FinFisher-Leak von 2014 kann eindeutig der Firmengruppe FinFisher zugeordnet werden.

⁴² <https://github.com/devio/FinSpy-Tools>, siehe auch Abschnitt *Öffentliche Dokumentation von Untersuchungsgegenständen und -methoden*, Seite 31

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

⁴⁴ Für eine Begriffsklärung siehe https://de.wikipedia.org/wiki/Reverse_Engineering

Dies ist ein deutlicher Hinweis darauf, dass die Samples – inklusive dem „adalet“-Sample von der Firmengruppe FinFisher stammen.

b) Verwendung gleicher Funktionen

Bei diesem Analyseschritt werden beispielhaft strukturelle Ähnlichkeiten in Code-Struktur und Funktionen analysiert, um Parallelen und Unterschiede aufzuzeigen. Die Darstellung der Ergebnisse erfolgt beispielhaft zur Illustration auffälliger und überzufälliger Übereinstimmungen. Aufgrund der Untersuchungsfrage der GFF wird hierbei der Schwerpunkt auf das „adalet“-Sample und dessen Ähnlichkeiten zu Samples gelegt, die zweifelsfrei von der Firmengruppe FinFisher stammen.

Anhand der Ausgaben des Java-Decompilers kann eine starke strukturelle und kontextuelle Ähnlichkeit zwischen den Versionen von 2014 und 2016 illustriert werden.

Die Funktion `run()`

In der Klasse `SmsInbox` (`Smsinbox.java`, `com\android\services\sms`) des Samples *421and* ist in Zeile 36 eine Funktion namens `run()` implementiert. Diese Funktion und deren Struktur ist beispielhaft in Abbildung 5 dargestellt.

In der Klasse `e` (`e.java`, `org\customer\fu\slms`) des Samples *adalet* ist in Zeile 66 eine Funktion namens `run()` implementiert, welche beispielhaft in Abbildung 6 abgebildet ist.

Das Sample *421and* stammt von der Firmengruppe FinFisher, wie im Kapitel 4. *Bezüge zu Samples bekannter Herkunft* auf Seite 10 dargelegt.

Ein Vergleich der Code-Struktur lässt vermuten, dass in der aktuelleren Version nach 2014 ein *Refactoring*⁴⁵ stattgefunden hat, um Ähnlichkeiten in der Programm-Syntax zu verschleiern. Dies wäre eine plausible und naheliegende Reaktion auf die 2014 erfolgte Veröffentlichung der Schadsoftware-Samples der Firmengruppe FinFisher: Durch ein Refactoring kann die Wahrscheinlichkeit einer automatisierten Detektion der Schadsoftware verringert werden.

Offenbar wurden Klassen, Variablen und Funktionen umbenannt und darüber hinaus sogenannte Obfuskatoren verwendet. Obfuskatoren verringern gezielt die Leserlichkeit und Vergleichbarkeit des Programmcodes, haben dabei jedoch keinen Einfluss auf die tatsächliche Funktionalität.

In den neueren Varianten wird ausgiebig eine sog. „leet-speak“ Variante einzelner Wörter verwendet. Hierbei werden bestimmte Buchstaben durch ähnlich aussehende Zahlen ersetzt, beispielsweise der Buchstabe „i“ durch „1“, der Buchstabe „e“ durch „3“, der Buchstabe „t“ durch die Zahl „7“ usw.

Die neue Namensgebung einzelner Code-Strukturen ähneln einander und beinhalten Indizien, dass der Code von einem deutschsprachigen Entwickler geschrieben wurde:

1. Die Struktur der Klasse wurde von `com/android/services/sms` nach `org/customer/fu/slms` umbenannt. Betrachten wir das Verb „s1ms“ ergibt sich daraus das Wort „sims“, abgeleitet von „simsen“⁴⁶ bzw. „SMS“ – einer ausschließlich in Deutschland gebräuchlichen⁴⁷ Umgangssprache.
2. Für deutschsprachige Entwickler spricht auch die Verwendung des Namens „fu“ anstelle der verbreiteten englischen Schreibweise „foo“ (Ausgesprochen: „fu“).

Neben der ähnlichen Syntax und des Kontextes beider Versionen der Funktion, sind in den Kontrollfluss-Graphen einzelner Funktionen signifikante Ähnlichkeit festzustellen, wie Abbildung 7 und Abbildung 8 zeigen; hier können die korrespondierenden Funktionen `CallLogs.run()` von 2014 und `a.run()` von 2016 miteinander verglichen werden. In beiden Versionen werden individuell gewählte Zeichenketten,

⁴⁵ Refactoring eine Strukturveränderung von Quelltexten unter Beibehaltung des beobachtbaren Programmverhaltens. Für eine Begriffsklärung siehe <https://de.wikipedia.org/wiki/Refactoring>

⁴⁶ vgl. Duden „simsen“: <https://www.duden.de/rechtschreibung/simsen>

⁴⁷ Im englischen Sprachraum ist der Begriff „text“ sowohl zur Bezeichnung einer Kurznachricht als auch des Verfassens einer solchen gebräuchlich.

wie beispielsweise tmp420 in obigen Beispielen, gewählt. Diese sind im Kontext betrachtet bewusst gewählt und die Wahrscheinlichkeit einer zufälligen Ähnlichkeit kann nahezu ausgeschlossen werden: Beide Versionen basieren im Kern auf einer identischen Source-Code-Basis.

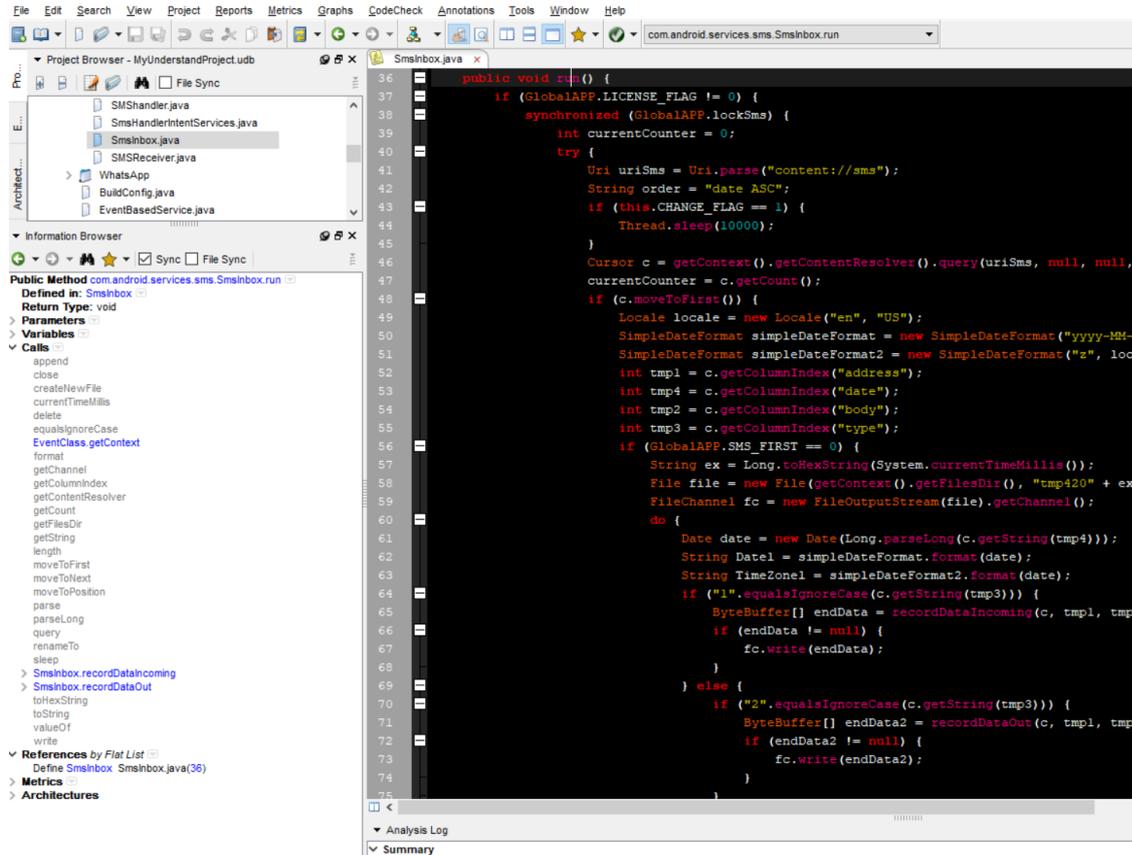


Abbildung 5. Sample 421and: com\android\services\sms

```

66 public final void run() {
67     if (a.k != 0) {
68         synchronized (a.aG) {
69             try {
70                 Uri parse = Uri.parse("content://sms");
71                 String str = "date ASC";
72                 if (this.a == 1) {
73                     Thread.sleep(10000);
74                 }
75                 this.b = a.e.getContentResolver().query(parse, null, null, null, str);
76                 int count = this.b.getCount();
77                 if (this.b.moveToFirst()) {
78                     Locale locale = new Locale("en", "US");
79                     SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss", locale);
80                     SimpleDateFormat simpleDateFormat2 = new SimpleDateFormat("a", locale);
81                     int columnIndex = this.b.getColumnIndex("address");
82                     int columnIndex2 = this.b.getColumnIndex("date");
83                     int columnIndex3 = this.b.getColumnIndex("body");
84                     int columnIndex4 = this.b.getColumnIndex("type");
85                     if (a.o == 0) {
86                         String hexString = Long.toHexString(System.currentTimeMillis());
87                         File file = new File(a.e.getFilesDir(), "tmp420" + hexString + ".dat");
88                         FileChannel channel = new FileOutputStream(file).getChannel();
89                         do {
90                             String format = simpleDateFormat.format(new Date(Long.parseLong(this.b.getString(columnIndex2))));
91                             String a2 = org.customer.fu.a.e.a(Long.parseLong(this.b.getString(columnIndex2)));
92                             if ("1".equalsIgnoreCase(this.b.getString(columnIndex4))) {
93                                 ByteBuffer[] b2 = b(this.b, columnIndex, columnIndex3, format);
94                                 if (b2 != null) {
95                                     channel.write(b2);
96                                 }
97                             } else {
98                                 if ("2".equalsIgnoreCase(this.b.getString(columnIndex4))) {
99                                     ByteBuffer[] a3 = a(this.b, columnIndex, columnIndex3, format);
100                                    if (a3 != null) {
101                                        channel.write(a3);
102                                    }
103                                }
104                            }
105                        } while (this.b.moveToNext());
106                    } channel.close();
107                    if (file.length() > 0) {

```

Public Method org.customer.fu.s1ms.e.run() **Return Type:** void

Defined in: e

Parameters:

Variables:

Calls: a, close, createNewFile, currentTimeMillis, delete, e.a, e.b, equalsIgnoreCase, f, format, getChannel, getColumnIndex, getContentResolver, getCount, getFilesDir, getString, length, moveToFirst, moveToNext, moveToPosition, parse, parseLong, query, renameTo, sleep, toHexString, toString, write

References by Flat List: Define e.java(66)

Metrics: 97 (CountLine), 97 (CountLineCode), 1 (CountLineComment), 22 (Cyclomatic)

Architectures: Directory Structure/org/customer/fu/s1ms

Abbildung 6. Sample adalet: org\customer\fu\s1ms

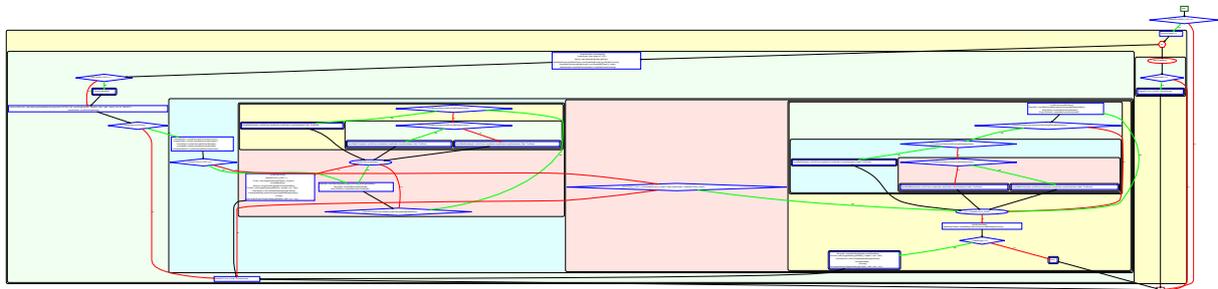


Abbildung 7. Sample 421and: ControlFlow com.android.services.CallLogs.run()

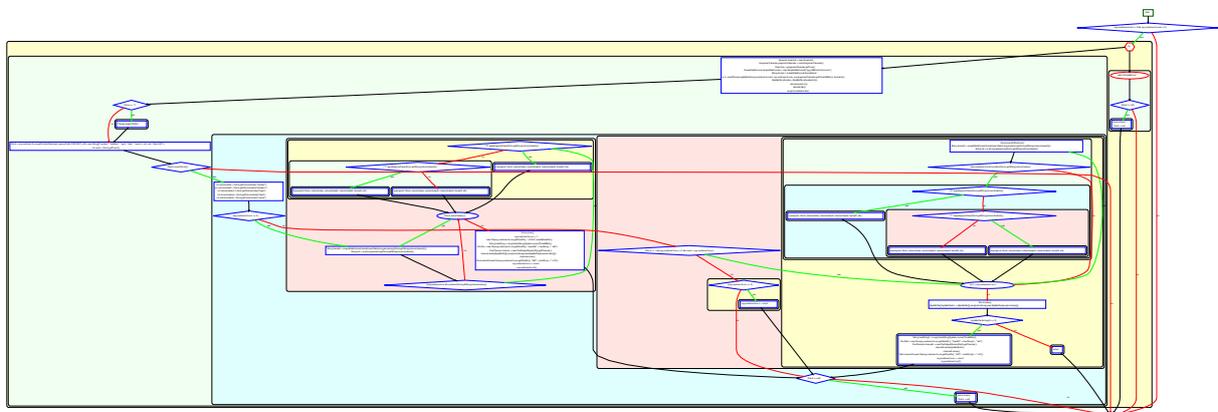


Abbildung 8. Sample adalet: ControlFlow org.customer.fu.e.a.run()

Abbildung 7 und Abbildung 8 finden sich in vergrößerter Darstellung im Appendix C & D, sowie als Einzeldateien im PDF-Format im Software Repository zum vorliegenden Bericht (siehe Abschnitt *Öffentliche Dokumentation von Untersuchungsgegenständen und -methoden*, Seite 33)

Bezüge zu Samples bekannter Herkunft

b) *Ähnlichkeit*

Im Abschnitt a) *SQLite-Version 3.13.0* auf Seite 13 wurden bereits die im Sample *adalet* verwendeten Libraries der Software *SQLite* beschrieben. Auch das Sample *flash28* verwendet die exakt gleichen Libraries, wie ein Vergleich der SHA-256-Prüfsummen zeigt.

adalet SHA-256 Prüfsummen:

| | |
|---|-----------------------------|
| 25c7ab9603506adb1e5ec475734763a519fbe19db94e2eeddf25604471541f21 | lib/arm64-v8a/library.so |
| 18689d9d4b76a011e410802204445980ff187065b120a1c7876a04e4633dbb89 | lib/arm64-v8a/libsqliteY.so |
| 2cc662cd13e5bd6720fff1217d77baf507558fea2b7469e9926f805f5a25f5d13 | lib/armeabi/library.so |
| e2340cfb97e7bcd2b938759291a2872d3dbe0b80b316922b7392a8b68f08d9e6 | lib/armeabi/libsqliteY.so |

flash28 SHA-256 Prüfsummen:

| | |
|---|-----------------------------|
| 25c7ab9603506adb1e5ec475734763a519fbe19db94e2eeddf25604471541f21 | lib/arm64-v8a/library.so |
| 18689d9d4b76a011e410802204445980ff187065b120a1c7876a04e4633dbb89 | lib/arm64-v8a/libsqliteY.so |
| 2cc662cd13e5bd6720fff1217d77baf507558fea2b7469e9926f805f5a25f5d13 | lib/armeabi/library.so |
| e2340cfb97e7bcd2b938759291a2872d3dbe0b80b316922b7392a8b68f08d9e6 | lib/armeabi/libsqliteY.so |

Die neueren, im Jahr 2019 öffentlich gemachten Samples verwenden ebenfalls das *Java Native Interface* und liefern im APK diverse Bibliotheken aus. In diesen Samples existiert meist nur eine Datei namens *libhelper.so*, die augenscheinliche Ähnlichkeiten mit den Dateien *library.so* und *libsqliteY.so* der älteren Versionen aufweist. Sollte eine Ähnlichkeit bei der Bewertung von wesentlichen Indizien eine Rolle spielen, empfehlen die Autoren, die Ähnlichkeiten detailliert zu verifizieren, indem inhaltliche Aspekte der unterschiedlichen *.so*-Dateien untersucht und verglichen werden.

Anhand der Ähnlichkeiten der nativen Bibliotheken in den APKs der Samples ab 2016 lässt sich folgern, dass diese eine gemeinsame Code-Basis verwenden, die der Firmengruppe FinFisher zugeordnet werden kann.

Nennenswerte Abweichung

Das Sample mit der SHA256-Summe

49c12654aaee1b089268931307f36a5d0d020325226328f780dc152b2f04b281 bildet eine Ausnahme: Es weist keine unmittelbaren Ähnlichkeiten zu den übrigen Samples auf. Dieses Sample beinhaltet im Verzeichnis *assets/* zahlreiche Programme und ein Shellscript:

| | |
|------------------|---|
| 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 |

Bei diesem Sample handelt es sich augenscheinlich um einen Container, der den als *dirtycow*⁴⁸ bekannten lokalen Kernel-Exploit beinhaltet. Dieser wird zur *Privilege Escalation*⁴⁹ verwendet: Unter Ausnutzung einer Schwachstelle kann ein nicht privilegierter Benutzer oder Prozess mittel *dirtycow* Administrationsrechte und somit die vollständige Kontrolle über das Gerät erlangen. Dies ist Voraussetzung, um dauerhafte Änderungen am System vorzunehmen, beispielsweise eine Schadsoftware zu installieren.

Ferner beinhaltet dieses Sample eine Tool-Suite zum sogenannten *rooten*⁵⁰ eines Android-Telefons. Diese Tools-Suite nennt sich *SuperSU*⁵¹ und befindet sich in der gleichnamigen ZIP-Datei. Das Sample bringt also alle Voraussetzungen mit, eine (weitere) Schadsoftware auf einem Zielgerät anzubringen.

Durch eine oberflächliche Analyse konnten wir feststellen, dass sich ein weiteres Sample in diesem APK befindet. Dieses Sample trägt den Dateinamen `sy.apk` und ist identisch mit dem ebenfalls analysierten Sample *PyawApp* `269227c4c4770e109e53c6cf87bd9bde367843c4806f5975c5aa317f318e28a9`. Dieses Sample wurde bereits von Experten des Schadsoftware-Spezialisten *Kaspersky* untersucht und der Software-Familie *FinSpy* zugeordnet⁵². Dieser Befund deckt sich mit unseren Beobachtungen. Der Name *PyawApp* deutet darauf hin, dass dieses Sample gegen burmesische Staatsbürger eingesetzt wurde, da *Pyaw* der Name eines populären burmesischen sozialen Netzwerks ist⁵³.

Die Autoren gehen davon aus, dass das Sample

`49c12654aaee1b089268931307f36a5d0d020325226328f780dc152b2f04b281` dem Zweck dient, *FinSpy* auf einem Zielgerät zu installieren. Eine detailliertere Analyse des Samples

`49c12654aaee1b089268931307f36a5d0d020325226328f780dc152b2f04b281` hat seitens des Chaos Computer Clubs aus zeitlichen Gründen und unter Berücksichtigung der Untersuchungsfragen nicht stattgefunden.

⁴⁸ Siehe <https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails>, zuletzt aufgerufen am 19. Dezember 2019

⁴⁹ Für eine Begriffsklärung siehe https://en.wikipedia.org/wiki/Privilege_escalation, zuletzt aufgerufen am 19. Dezember 2019

⁵⁰ Für eine Begriffsklärung siehe <https://www.heise.de/select/ct/2018/16/1533001012731723>, zuletzt aufgerufen am 19. Dezember 2019

⁵¹ Siehe <https://supersuroot.org/>, zuletzt aufgerufen am 19. Dezember 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/>, zuletzt abgerufen am 19. Dezember 2019

⁵³ Siehe https://play.google.com/store/apps/details?id=mm.com.pyaw&hl=en_US und <https://www.pyaw.com.mm/>, beide zuletzt abgerufen am 19. Dezember 2019

Tabelle 3. Übersicht der untersuchten Samples

| Konfiguration | TargetID | Initiale Meldung an VirusTotal | Datum | | Zertifikat | Charakteristika | Entwickler-Zertifikat | FinSpy? | |
|---------------|----------|--------------------------------|------------------------|------------|-----------------------------------|-----------------|-----------------------|--------------|------------------|
| | | | Datum des Kompilierens | Zertifikat | | | | Primärquelle | Analyse-Ergebnis |
| again | | 2012-09-05 | | 2011-12-06 | einmalig | ✓ | | | |
| JHANUK | | 2012-09-05 | | 2012-02-22 | 60:6D... | ✓ | ✓ | | |
| Andriod | | 2012-09-05 | | 2012-02-22 | 01:AB... | ✓ | ✓ | | |
| derise | | 2012-09-06 | | 2012-02-22 | 98:5D... | ✓ | ✓ | | |
| AKDemo | | 2014-08-06 | | 2012-10-10 | Textfragmente "Gamma" "Finfisher" | | | | |
| ANDDemo | | 2014-08-06 | | 2012-10-10 | Konfiguration in ZIP-Metadaten | ✓ | ✓ | | |
| 428 | | 2014-08-06 | | 2012-10-10 | JNI | ✓ | ✓ | | |
| tmWoot | | 2014-08-06 | | 2012-10-10 | DirtyCow Exploit | ✓ | ✓ | | |
| ANDR | | 2014-08-06 | | 2012-10-10 | | ✓ | ✓ | | |
| Android | | 2014-08-06 | | 2012-10-10 | | ✓ | ✓ | | |
| ANDXloe | | 2014-08-06 | | 2012-10-10 | | ✓ | ✓ | | |
| 421land | | 2014-08-06 | | 2012-02-22 | | ✓ | ✓ | | |
| trekki | | 2014-08-06 | | 2012-10-10 | | ✓ | ✓ | | |
| zefix | | 2014-08-06 | | 2012-10-10 | | ✓ | ✓ | | |
| testAD | | 2014-08-06 | | 2012-10-10 | | ✓ | ✓ | | |
| defs | | 2014-08-06 | | 2012-10-10 | | ✓ | ✓ | | |
| flash28 | | 2017-01-20 | 2016-05-18 | 2016-10-10 | | ✓ | ✓ | | |
| flash28 | | 2017-07-21 | 2016-05-18 | 2017-07-18 | | ✓ | ✓ | | |
| adalet | | 2017-07-27 | 2016-05-18 | 2016-10-10 | | ✓ | ✓ | | |
| cleaner | | 2019-03-19 | 2017-12-07 | 2018-12-21 | | ✓ | ✓ | | |
| whistel | | 2019-03-20 | 2017-12-07 | 2018-12-21 | | ✓ | ✓ | | |
| ŞcontainerŞ | | 2019-03-24 | 2016-03-08 | 2018-09-07 | | ✓ | ✓ | | |
| PyawApp | | 2019-03-24 | 2017-12-07 | 2018-06-20 | | ✓ | ✓ | | |
| network | | 2019-05-12 | 2017-12-07 | 2018-12-05 | | ✓ | ✓ | | |
| Disk | | 2019-05-14 | 2017-12-07 | 2018-12-05 | | ✓ | ✓ | | |
| FunVoic | | 2019-05-15 | 2017-12-07 | 2018-12-21 | | ✓ | ✓ | | |
| Diary | | 2019-05-21 | 2017-12-07 | | | ✓ | ✓ | | |
| myphone | | 2019-05-21 | 2017-12-07 | 2018-12-21 | | ✓ | ✓ | | |

Fazit

A. Feststellung des Herstellungszeitpunktes

1) Wann wurde das „adalet“-Sample produziert und eingesetzt?

Das „adalet“-Sample kann frühestens am 18. Mai 2016 hergestellt worden sein. Teile des „adalet“-Samples wurden augenscheinlich erst am 23. September 2016 erstellt. Es liegt weiterhin nah, dass das „adalet“-Sample erst nach dem 10. Oktober 2016 eingesetzt wurde.

2) Liegt der Zeitpunkt bzw. Zeitraum vor oder nach dem 18. Juli 2015?

Es konnte **bewiesen** werden, dass das von *der Gesellschaft für Freiheitsrechte* zur Untersuchung eingereichte Sample **frühestens am 18. Mai 2016** (Siehe *SQLite-Version 3.13.0*, Seite 13) hergestellt wurde. Somit ist **bewiesen**, dass das Sample **nach dem 18. Juli 2015** erstellt wurde.

Der vermutliche früheste Einsatz konnte darüber hinaus auf einen Zeitraum **nach dem 10. Oktober 2016 eingegrenzt** werden (Siehe 2. *Timestamps in Zertifikaten*, Seite 7).

B. Feststellung der Herkunft

1) Stammen die Samples aus unterschiedlichen Quellen, oder gibt es eindeutige Hinweise auf eine gemeinsame Urheberschaft?

Sämtliche im Rahmen dieser Untersuchung analysierten Samples teilen eindeutige Hinweise auf eine gemeinsame Urheberschaft.

2) Können die Urheber der Samples identifiziert werden?

Tabelle 3 zeigt die Entwicklung der untersuchten Samples über die Zeit. Samples, die bekanntermaßen von der Firmengruppe FinFisher stammen, sind rot hervorgehoben. Samples, die als Ergebnis der Analyse der Firmengruppe FinFisher zugeordnet werden können, sind gelb markiert. Das „adalet“-Sample und seine Ähnlichkeiten anderen Samples sind orange hervorgehoben.

Durch iterative Weiterentwicklungen über die Jahre konnte gezeigt werden, dass das zur Untersuchung eingereichte Sample `c2ce202e6e08c41e8f7a0b15e7d0781704e17f8ed52d1b2ad7212ac29926436e` („adalet“) eine Weiterentwicklung der Samples

`abcb11c4787c62ab90cecc262f6fc98bd7f73335ac631c2e9aec8734088e91aa,`
`2795e777d897857ab6fa19f85687a23a6071ab665299a47b8b952cb4e7056a07,`
`1935f2e52832df910edc1b7ef17b53d8c852fe66ec3afbe490ffc2ef057452b3,`
`045161094b9f6b98c4ef87d2324f4bb8a0d0fbf58e349a37d11622aef2e6b051,`
`587b110da2ef9c59b18f01e97e9b12628f3e7b2e88611f7cb28a6efccb0aaba2,`
`704d599fe51e7a0f982438c13983fb936dd1530f659e8036bee69752221ef7d7,`
`1ea335d1d5f99aeb1a516d6b267ba53c38438648874752eb0438edfffde380d,`
`1507ee069906d2a42216d77ef51d42a35efcc59b005b55d8ea771749057296db,`
`60dc08ab28db5ba3a56734097954861a525b4b384e8067e3eaac551c4cb8ece3,`
`84d39e5c6db75801a85cc5d2557ab536abe40496b23eba6b3e5c1722975d8f32,`
`84d231e6ea1e2e3283c3e9cbfcabeded0d7e5723852e378e0caf5bb001501938` und
`26c6205366e952bd9bc3f4c01983dae74ca589fda0205f8b2b387de512eafba1`

aus den Jahren 2012-2014 ist. Dieses Samples können eindeutig der Firmengruppe FinFisher zugewiesen werden.

Die Autoren sehen es als erwiesen an, dass das Sample **aus dieser Quelle** stammt.

Öffentliche Dokumentation von Untersuchungsgegenständen und -methoden

Der Chaos Computer Club e. V. (CCC) ist die größte europäische Hackervereinigung und seit über dreißig Jahren Vermittler im Spannungsfeld technischer und sozialer Entwicklungen. Die vorliegende Analyse wurde ehrenamtlich nach bestem Wissen und Gewissen von Thorsten Schröder und Linus Neumann durchgeführt.

Zu den Grundsätzen der Hacker-Ethik⁵⁴ gehört die Informationsfreiheit. Um dem wissenschaftlichen Anspruch dieses Gutachtens gerecht zu werden, haben die Autoren sämtliche im Rahmen dieser Bewertung analysierten Samples, deren Extrakte, sowie selbst entwickelte und verwendete Tools veröffentlicht.

Die vorliegende Analyse kann von Fachleuten und solchen, die es werden wollen, mit Hilfe der von uns veröffentlichten Tools und Dokumentationen vollständig nachvollzogen werden. Wir laden die deutsche und internationale Forschungsgemeinschaft ein, unsere Ergebnisse kritisch zu prüfen, zu ergänzen und – falls nötig– zu korrigieren.

Insbesondere fordern wir deutsche Ermittlungsbehörden, die ebenfalls Kunden der Firmengruppe FinFisher sind, dazu auf, unsere Analyseschritte an den ihnen vorliegenden Samples nachzuvollziehen.

Sämtliche zum Durchführen unserer Analyse benötigten Untersuchungsgegenstände und -methoden sind in folgenden Repositories verfügbar:

- **FinSpy-Tools:** Werkzeuge für die Analyse der hier genannten Samples und anderen Android-basierten FinSpy-Samples.
<https://github.com/devio/FinSpy-Tools>
- **FinSpy-Dokumentation:** Dokumentation der Analysen einzelner Komponenten der FinSpy-Schadsoftware, Extrakte, Samples und Helfer-Skripte.
<https://github.com/linuzifer/FinSpy-Dokumentation>

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

Appendix

A. Veröffentlichungszeitpunkt der SQLite-Version 3.13.0

In der Mitteilung zur Veröffentlichung ist, wie im Abbildung 9 illustriert, neben dem Zeitstempel der Veröffentlichung ebenfalls eine Prüfsumme veröffentlicht. Diese Kombination wird hier als `SQLITE_SOURCE_ID` bezeichnet.



SQLite Release 3.13.0 On 2016-05-18

Small. Fast. Reliable.
Choose any three.

Home About Documentation Download License Support Purchase Search

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 [dc6ebeda9396087](#)
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.

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

Die gleiche `SQLITE_SOURCE_ID` mit dem Wert `2016-05-18 10:57:30 fc49f556e48970561d7ab6a2f24fdd7d9eb81ff2` befindet sich ebenfalls in der im „adalet“-Sample enthaltenen Datei `arm64-v8a\libsqliteY.so` (Siehe Abbildung 10)

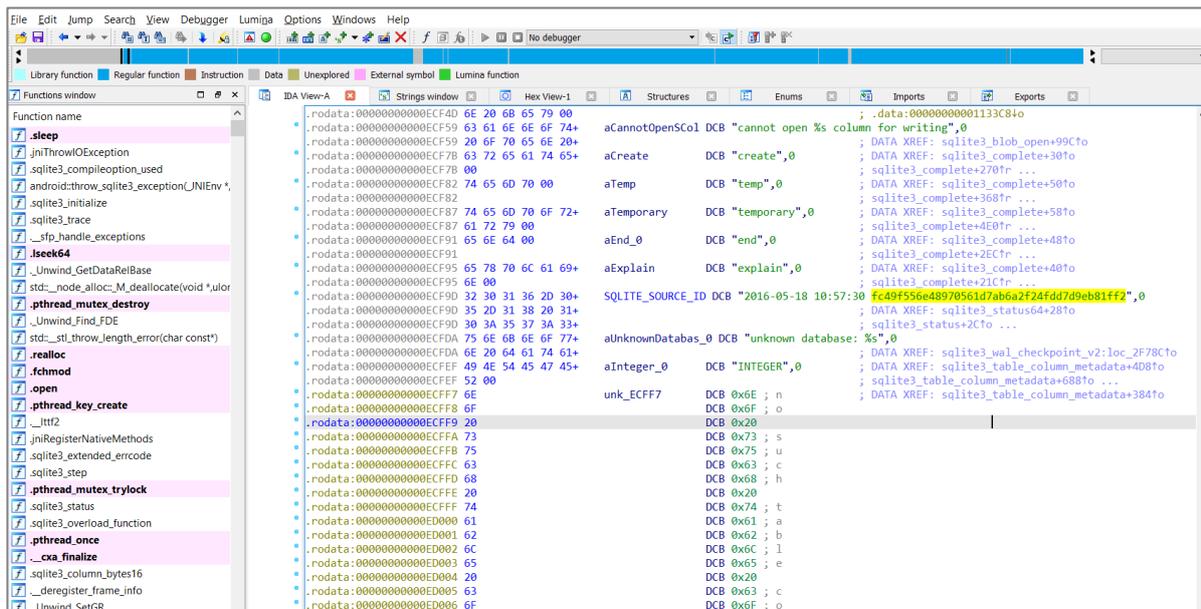


Abbildung 10 - Disassembly der Datei libsqliteY.so in IDA Pro 7.3

Die Binärdateien der Bibliothek wurden für 32-Bit und 64-Bit ARM-Prozessoren im „adalet“-Sample bereitgestellt. Eine Analyse der verwendeten Compiler-Versionen der einzelnen Bibliotheken erfolgte mit Hilfe des Werkzeugs 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 (pre
0020 656c6561 73652900 4f626675 73636174 elease).Obfuscat
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).Androi
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) .
```

Das Ergebnis für die 64bit-Version beinhaltet den gleichen Hinweis auf die Version des verwendeten Compilers.

```
$ 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 (pre
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) .
```

Es ist ersichtlich, dass Android clang version 3.8.256229 bei der Entwicklung zum Einsatz gekommen ist. Android clang version 3.8.256229 basiert auf LLVM 3.8.256229.

LLVM Version 3.8.0 wurde im März 2016 veröffentlicht, eine Version 3.8.256229 kann daher nicht bereits im Jahr 2015 zum Einsatz gekommen sein (Siehe Abbildung 11)

[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](#)

Abbildung 11 - Ankündigung der Veröffentlichung von LLVM v3.8

B. Konfiguration sämtlicher im Rahmen dieser Analyse untersuchten Samples

Zur vollständigen Dokumentation sind im Folgenden die Konfigurationen sämtlicher im Rahmen der vorliegenden Analyse untersuchten Samples aufgelistet. Zur Verbesserung der Lesbarkeit erfolgt die Darstellung im Querformat. Die Samples sind nach dem Datum ihrer ersten öffentlichen Entdeckung sortiert.

Die in den Konfigurationen enthaltenen Telefonnummern und IP-Adressen erlauben unter Umständen einen Rückschluss auf Hersteller oder Auftraggeber, sofern gesicherte Informationen über die Anschlussinhaber gewonnen werden können. Ein einfacher Rückschluss auf Basis einer Landesvorwahl oder der Geo-Lokation einer IP-Adresse ist jedoch nicht zulässig, da sowohl das Registrieren von IP-Adressen als auch das Schalten von Telefonanschlüssen weltweit mit geringem Aufwand unter Inanspruchnahme vielfältiger Verschleierungsmöglichkeiten gelingen kann.

Konfiguration des Samples 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     = 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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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\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)
```

Konfiguration des Samples 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\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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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\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)
```

Konfiguration des Samples 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\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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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 = 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)
```

Konfiguration des Samples 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\xe1\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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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= "0" (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= "0" (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= "0" (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= "0" (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= "0" (12)
```

```
TlvTypeMasterAgentUserPermissionValuePacket    =  
  b'\x0c\x00\x00\x00@\xcaq\x00\xf1\x03\x00\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\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\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\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)
```

Konfiguration des Samples 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)
```

Konfiguration des Samples 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) |

Konfiguration des Samples 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) |

Konfiguration des Samples 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) |

Konfiguration des Samples 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) |

Konfiguration des Samples 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) |

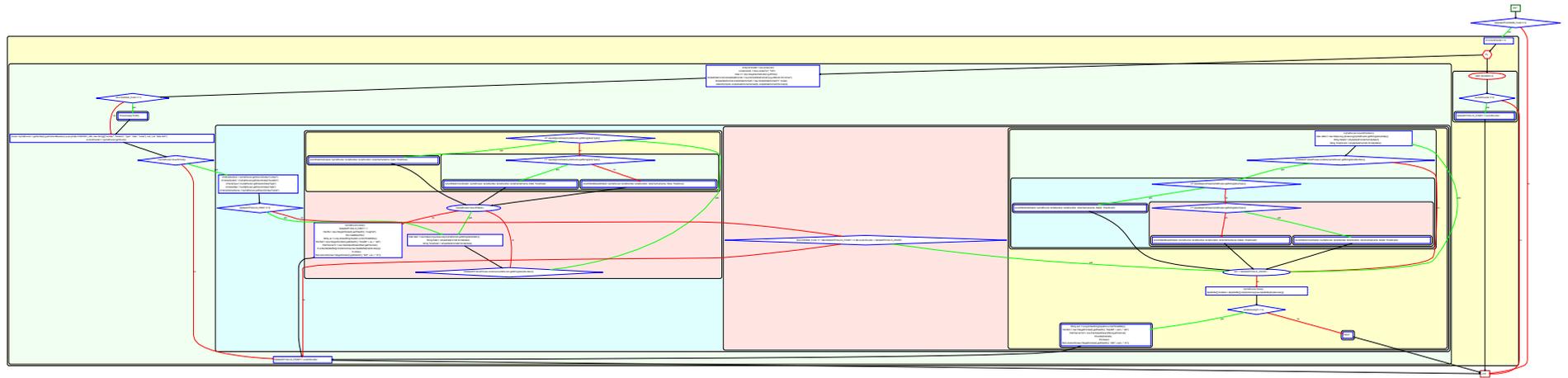
Konfiguration des Samples 3f8baeae01980e77fa905216e291b6478105295c8372a003d73e9086b0b3e964

| | |
|--------------------------------------|--|
| 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) |

Konfiguration des Samples 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()

