Technische Berichte in Digitaler Forensik

Herausgegeben vom Lehrstuhl für Informatik 1 der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) in Kooperation mit dem Masterstudiengang Digitale Forensik (Hochschule Albstadt-Sigmaringen, FAU, Goethe-Universität Frankfurt am Main)

Anwendungsanalyse des Messengers Telegram Desktop (Version 0.9.15) unter Windows 10

Christian Oertle

14.03.2016

Technischer Bericht Nr. 6

Zusammenfassung:
Sichere Messenger, wie die Anwendung Telegram Desktop, erfreuen sich zunehmender Beliebtheit, seit die Übernahme von WhatsApp durch Facebook bekannt wurde. Telegram Desktop wurde als Multi-Plattform-Messenger für den Austausch von Nachrichten und Medien entwickelt. Telegram Desktop nutzt hierfür die Cloud-Infrastruktur des Anbieters. Im Rahmen dieser Arbeit wurde die Applikation in Version 0.9.15 auf Erzeugung persistenter Spuren bei der Nutzung untersucht. Die Ergebnisse sollten im Wesentlichen auch auf alle aktuellen Folgeversionen übertragbar sein. Diese Arbeit entstand im Rahmen des Moduls Browser- und Anwendungsforensik des Studiengangs Digitale Forensik im Wintersemester 2015/2016 unter der Anleitung von Felix Freiling, Holger Morgenstern und Michael Gruhn.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Aufgabenstellung

Das Ziel dieser Hausarbeit ist die forensische Untersuchung einer Anwendung. Dies bedeutet konkret die auftretenden Spuren der Anwendung zu untersuchen und zu dokumentieren, sodass diese ggf. für die Arbeit anderer Forensiker oder für ein Gutachten verwendet werden können. Daher wird der Schwerpunkt und der Fokus dieser Arbeit auf die relevanten Spuren für eine praktische Arbeit bspw. für den täglichen Ablauf bei Ermittlungsbehörden gelegt.

1.2 Analyseumgebung

Als Analyseumgebung wurde ein Linux-System auf Ubuntu-Basis verwendet. Es handelte sich hierbei um ein Kubuntu 15.10 (Linux-Kernel-Version 4.2.0-22)[1]. Dieses System wurde als Hostsystem für die virtualisierte Testumgebung eingesetzt.

Als Virtualisierungsumgebung kam VirtualBox in Version 5.0.12 zum Einsatz.[2]

Für die Untersuchung der Datenübertragung der Software kam Wireshark in der Version 1.12.7 zum Einsatz.[3]

1.3 Virtuelle Maschine / System zum Anwendungsbetrieb

Als Gast-Betriebssystem zum Betrieb der zu analysierenden Anwendung wurde Windows 10 als 64 Bit Version in der virtualisierten Umgebung genutzt.

2 Untersuchte Anwendung

Bei der zu untersuchenden Applikation handelte es sich um den Instant-Messenger Telegram. Diese Applikation ist seit August 2013 unter iOS und Oktober 2013 unter Android verfügbar und wurde dem Grundgedanken entsprechend als Multiplattform-Instant-Messenger auf die am häufigsten genutzten Desktop- und Mobilgeräte-Betriebssysteme portiert. Derzeit werden Linux, Mac OSX, Windows, Apple iOS, Android und Windows Phone neben einem direkten Web Zugriff in die Cloud unterstützt. Wie Telegram auf der Webseite mitteilt, handelt es sich um eine cloud-basierte App für mobile und Desktop Systeme mit dem Schwerpunkt auf Geschwindigkeit und Sicherheit.[4]

[1] (Canonical, 2016)
[2] (Oracle, 2016)
[3] (Wireshark Foundation, 2015)
[4] (Durov & Durov, Telegram, 2016)

***Anmerkung**:*

Der Fokus dieser Arbeit liegt auf einer praktischen Orientierung, bspw. als Hilfe für einen Forensiker. Auf Grund der auf der Webseite von Telegram nachvollziehbaren raschen Updaterate, sowie der geringen praktischen Relevanz der Timestamps und der Hashwerte der Installation und Dateien im arbeitstäglichen Umfeld dieser Untersuchung bei Ermittlungsbehörden in Verbindung mit der Fragestellung, welche Spuren generell bei der Nutzung hinterlassen werden, haben den Verfasser dazu bewogen, auf diese Informationen im Rahmen dieser Arbeit zu verzichten. Für die Zeit der Untersuchung der Applikation wurde bewusst die Auto-Update-Funktion der Applikation deaktiviert, welche in den Standard-Einstellungen aktiv ist. Im Untersuchungszeitraum wären bereits so fünf verschiedene stable-Programmversionen entsprechend untersucht worden.

2.1 Quelle

Die Applikation sowie ihr Quellcode stehen online zur Verfügung.[5] Die Untersuchung beschränkt sich auf die Installationsversion, die als Windows Portable Executable (PE32 Gui) zum Download angeboten wurde.

Für die Installation wurde konkret die Version 0.9.15 von Telegram (tsetup.0.9.15.exe, md5-Hashwert 5605dbd2880f756ec9de9eb4c0166da5) genutzt. Es gab keine spezielle Auswahlmöglichkeit für eine Systemarchitektur (32 oder 64 Bit).

2.2 Installation

Das Ausführen der Installationsdatei und der Durchlauf der Installation mit den vorgeschlagenen Standardwerten führte zu einer Installation des Programms. Hier konnten mittels der Zustands- und Ereignismethode folgende Erkenntnisse gewonnen werden:

Lfd. Nr.	Typ	Beschreibung	Wert
001	Ordner	Installations-Ordner	C:\Users\m16user\AppData\Roaming\Telegram Desktop
002	Datei	Applikation	[001]\Telegram.exe
003	Datei	Logfile zur Deinstallation	[001]\unins000.dat
004	Datei	Applikation, Uninstaller	[001]\unins000.exe
005	Datei	Applikation	[001]\Updater.exe
006	Ordner	User Desktop	C:\Users\m16user\Desktop\
007	Datei	Link	[006]\Telegram.lnk
008	Ordner	Startmenü	C:\Users\m16user\Roaming\Microsoft\Windows\Start Menu\Programs\Telegram Desktop
009	Datei	Link	[008]\Telegram entfernen.lnk
010	Datei	Link	[008]\Telegram.lnk

Tabelle 1: Installationsauswertung

[5] (Durov & Durov, Telegram, 2016) und (Telegram, 2016)

Des Weiteren erfolgte eine Aktualisierung des Icon Cache (iconcache_16.db, iconcache_32.db und iconcache_idx.db) unter „C:\Users\m16user\AppData\Local\Microsoft\Windows\Explorer\“

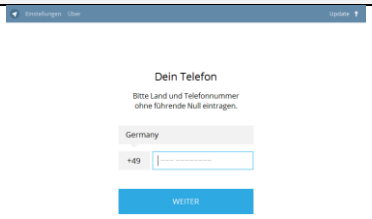
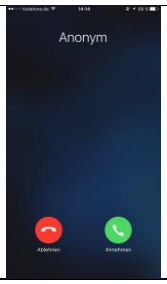
Darüber hinaus konnten Zugriffe auf die Registry nachvollzogen werden. Detaillierte Zugriffsaufzeichnungen finden sich bei Interesse hierzu in Anlage 1.

2.3 Ersteinrichtung

Dem Grundgedanken eines cloud-basierten Instant-Messenger folgend, stellen die Clients im Groben folglich nur Zugriffsmöglichkeiten auf die Daten der Cloud dar. Unabhängig davon ob zur Verbesserung der Geschwindigkeit oder anderer Ziele hierfür eine zusätzliche lokale Synchronisation der Daten erfolgt.

Dem entsprechend muss für die Nutzung der Client mit einem serverbasierten Benutzerkonto verknüpft werden. Da die Untersuchung für diese Arbeit auf der Nutzung des installationsbasierten Windows-Clients liegt, erfolgt die Beschreibung exemplarisch für diesen Client.

Die Verknüpfung wurde hierbei, ähnlich wie bei anderen Instant Messengern, bspw. Whats App, über die Zuordnung zu einer Telefonnummer realisiert.

Einrichtungsschritt	Beschreibung
	Zunächst ist es notwendig eine Telefonnummer für die Verifikation und Verknüpfung anzugeben. An diese Nummer wird, zumindest bei der ersten Einrichtung eines Benutzerkontos, ein 5-stelliger Verifikationscode per SMS gesandt.
	Der per SMS erhaltene Code.
	Der per SMS erhaltene Code musste zur Aktivierung des Accounts eingegeben werden.
	Alternativ war es möglich, sich seitens Telegram von einer anonymen Nummer anrufen zu lassen, dies ermöglicht ggf. auch die Nutzung einer Festnetznummer, statt eines Mobilfunkgerätes als „Authentifizierung“.

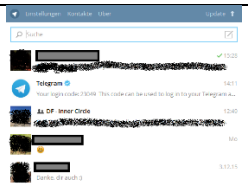
	Sofern bereits eine Anmeldung auf einem anderen Gerät erfolgte, wurde bei weiteren Client-Account-Verknüpfungen der Code auch zunächst an den Telegram-Messenger gesandt.
	Bei korrekter Eingabe wurde die Verknüpfung mit dem Konto hergestellt.

Tabelle 2: Einrichtung der Installation

Da Telegram als Multi-Plattform-Messenger konzipiert ist und die Daten cloud-basiert gespeichert sind, erfolgt eine Synchronisation mit den Daten zwischen allen Instanzen auf den Geräten.

Ebenfalls wurden bei späterer Einrichtung eines weiteren Clients die bisherigen Chat-Verläufe übernommen, da ja serverbasiert gespeichert.

2.4 EXKURS „geheimer Chat“

Auffallend war hierbei, dass geheime Chats – welche nur von einer Mobile App gestartet werden können – lediglich auf dem initiierenden mobilen Gerät angezeigt wurden. Bei der Recherche[6] hierzu zeigte sich, dass diese geheimen Chat-Kanäle nicht auf dem Server zwischengespeichert werden und mit einer Ende-zu-Ende-Verschlüsselung versehen sind. Ebenfalls ist eine Weiterleitung nicht möglich und es kann zusätzlich noch ein Selbstzerstörungstimer für Nachrichten gesetzt werden. Die zuletzt genannten Funktionen sind jedoch eher als technische Spielerei zu betrachten, denn als echte Sicherheitsfunktion, da mittels einfacher Screenshots diese Funktionen umgangen werden können.

Da Telegram teilweise im Quellcode vorliegt und die Anbieter-Webseite zur Entwicklung eigener Applikationen mit Zugriff auf das Telegramnetzwerk ermuntert, konnte in einer Recherche zu Applikationen herausgefunden werden, dass es zwischenzeitlich auch Desktop-Applikationen (für Linux) gibt, die ebenfalls den geheimen Chat vermeintlich unterstützen.[7]

2.5 Laufzeitanalyse

2.5.1 Datenspeicherung

2.5.1.1 Ordner

Die in 2.2 aufgezeigte Struktur (C:\Users\m16user\Roaming\Microsoft\Windows\Start Menu\Programs\Telegram Desktop) wurde während der Einrichtung um weitere Ordner und Daten ergänzt.

Die Struktur der Installation ist Abbildung 1: Datenstruktur nach Einrichtung zu entnehmen. Es zeigte sich nach der Einrichtung, dass neben einem Update-Verzeichnis (tupdates) – als Zwischenablage für

[6] (Durov & Durov, Telegram, 2016)
[7] (Ubuntuusers, 2016)

Updates bis zum nächsten Programmstart – und einem Logfile (log.txt) – Fehlermeldungen und allgemeine Statusinformationen – vor allem ein neues Datenverzeichnis (tdata) angelegt wurde.

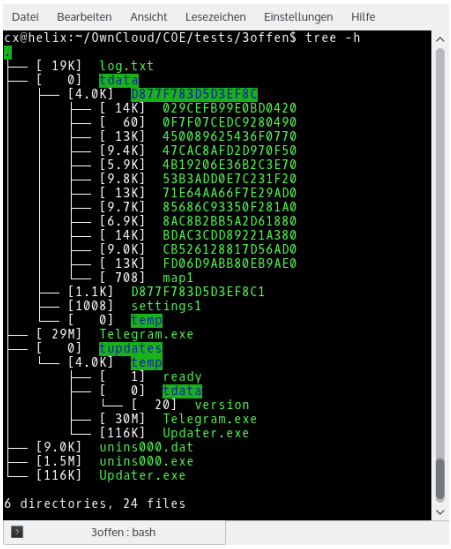


Abbildung 1: Datenstruktur nach Einrichtung (Ordner temp wurde durch eine Aktion erstellt. Details in Kapitel 2.5.1.5)

Der Vollständigkeit halber sei an dieser Stelle erwähnt, dass das Update-Verzeichnis in dieser Aufnahme auch schon die bereits für den nächsten Applikationsstart heruntergeladenen Updates (Telegram.exe, Updater.exe, usw.) anzeigt. Dies ist der zunächst noch aktiven Auto-Update-Funktion geschuldet, welche vor einem Neustart der Applikation deaktiviert wurde, um ein entsprechendes Update der Applikation zu verhindern.

Das vor allem interessante Verzeichnis tdata beinhaltete unter anderem einen Ordner D877F783D5D3EF8C (16 Zeichen) dessen Name über sämtliche Installationen auf verschiedenen Testumgebungen (Windows, Linux und Mac) identisch war.

2.5.1.2 Allgemeine Dateien

Im Folgenden werden lediglich die interessanten Daten der Applikation weiter beschrieben.

Der unter 2.5.1.1 benannte Ordner D877F783D5D3EF8C beinhaltete Dateien deren Namen nach einem ähnlichen Muster aufgebaut sind (17 Zeichen), sowie eine Datei „map0“ bzw. „map1“ (künftig durch map[0|1] dargestellt)[8].

Die Dateinamen der Inhaltsdateien (bspw. 7F5072981764E0640) setzen sich aus 16 Zeichen ähnlich dem Ordnernamen zusammen und einer abschließenden 0 oder 1. Für die Benennung des genannten Ordners und der Dateien kamen lediglich Zeichen und Zahlen zur Verwendung, die aus dem hexadezimalen Zeichenraum stammen.

Die weitere Untersuchung der Dateien selbst zeigte TDF$ als Magic-Byte-Kennung für die Inhaltsdateien, wie auch für die Datei map[0|1] bzw. settings[0|1]. Die Inhalte der Dateien waren auf Grund einer Verschlüsselung nicht lesbar.

[8] Gleiches gilt für die Darstellung bei ähnlichen Namensmustern anderer Dateinamen.

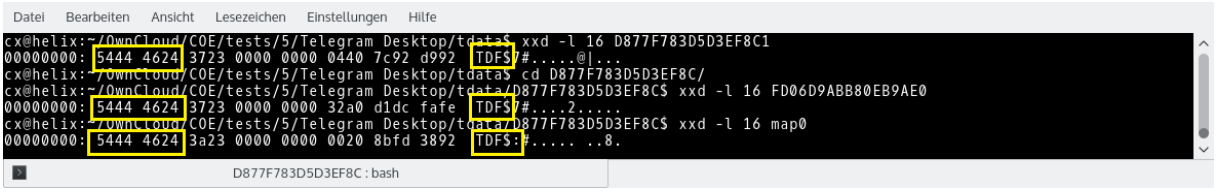


Abbildung 2: Magic Byte

Die Untersuchung des Applikationsverhaltens bei Nutzung gab darüber hinaus nachfolgende Erkenntnisse.

2.5.1.3 Schreiben und Versand

Während dem Schreiben und dem Versand einer Nachricht in einem Chat-Kanal konnte die Erstellung und Modifizierung von Dateien im zuvor genannten Inhalts-Datenordner tdata beobachtet werden.

Erläuterungen: siehe hierzu die nachfolgende Abbildung 3: Schreiben und Versand einer Nachricht und Tabelle 3: Phasen des Schreibens und Versandes.

08. Jan 16	16:01:43	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\878986D75BC8DD880
08. Jan 16	16:01:43	Modified	76 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\878986D75BC8DD880
08. Jan 16	16:01:43	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:01:43	Modified	76 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\878986D75BC8DD880
08. Jan 16	16:01:43	Created	76 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\056BEEC9BE8CAAA20
08. Jan 16	16:01:43	Modified	76 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\056BEEC9BE8CAAA20
08. Jan 16	16:01:43	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:01:43	Modified	76 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\056BEEC9BE8CAAA20
08. Jan 16	16:01:43	Created	2.29 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
08. Jan 16	16:01:43	Modified	2.29 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
08. Jan 16	16:01:43	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:01:43	Modified	2.29 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
08. Jan 16	16:01:43	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
08. Jan 16	16:01:43	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:01:44	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\878986D75BC8DD880
08. Jan 16	16:01:44	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:01:44	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\056BEEC9BE8CAAA20
08. Jan 16	16:01:44	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:01:45	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
08. Jan 16	16:01:45	Modified	2.24 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
08. Jan 16	16:01:45	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:01:45	Modified	2.24 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
08. Jan 16	16:01:45	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1

Abbildung 3: Schreiben und Versand einer Nachricht

	Phase	Beschreibung
Erstellen/Schreiben der Nachricht	Gelb	Die erste Datei wurde angelegt und im Anschluss verändert. Die Benennung der Daten erfolgt nach bekannten Muster mit 17 Stellen (16 Stellen aus dem „hexadezimalen Zeichenraum“ + abschließende 0). Bei mehrmaligen Testdurchläufen konnte festgestellt werden, dass die erzeugten Dateien jeweils anders benannt waren und es zu keiner Namenswiederholung, selbst bei gleichem Nachrichteninhalt, kam.
	Orange	Das Gleiche geschah mit der zweiten Datei.
	Grün	Im Anschluss an die Dateioperationen konnte das Anlegen und Verändern der Datei map1 beobachtet werden. Nach Abschluss der Modifikationen der Datei map1 wurde die Datei map0 gelöscht.

Versand der Nachricht	Blau	Der Versand (Klick auf Button „Senden“) der Nachricht an den Chat-Kanal löste die Löschung der zwei zuvor erstellten Dateien aus.
	Rot	Im Anschluss an die Löschung der Dateien, wurde die Datei map0 erstellt und modifiziert. Nach Abschluss der Modifikation wurde die bestehenden Datei map1 gelöscht.

Tabelle 3: Phasen des Schreibens und Versandes

Zwischen den Datei-Operationen kommt es, wie aus dem Verlauf in Abbildung 3: Schreiben und Versand einer Nachricht ersichtlich, auch zu Updates an den Ordnerinformationen.

Der erstmalige Versand eines Bildes bzw. einer Datei im Chat führte zu der Erstellung des Ordners C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\tdummy. Dieser Ordner wurde jedoch für den eigentlichen Versand nicht genutzt und es erfolgte auch keine Zwischenspeicherung der Datei bzw. des Bildes. Er wurde lediglich angelegt und blieb auch nach Versand leer bestehen.

2.5.1.4 Schreiben von Nachrichten auf einem anderen eigenen Gerät

Wird eine Nachricht auf einem eigenen anderen Gerät geschrieben (bspw. unter Linux oder iOS) und durch die Synchronisation der Nachrichten über die Cloud in dem Windows Client angezeigt, erfolgt keine Schreiboperation an einer der Dateien im Telegram Desktop Ordner. Dies legt die Vermutung nahe, dass Textnachrichten lediglich online vorgehalten werden und im flüchtigen Speicher vorliegen.

Der Versand von Bildern auf eigenen anderen Endgeräten, hier unter einem Linux-System, führte zu einer Schreiboperation unter Windows im tdata-Verzeichnis.

09. Jan 16	15:56:43	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\6132D34F940873A70
09. Jan 16	15:56:43	Modified	15.62 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\6132D34F940873A70
09. Jan 16	15:56:43	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	15:56:43	Modified	15.62 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\6132D34F940873A70
09. Jan 16	15:56:43	Created	492 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\E6280A3C5ABEB3190
09. Jan 16	15:56:43	Modified	492 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\E6280A3C5ABEB3190
09. Jan 16	15:56:43	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	15:56:44	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
09. Jan 16	15:56:44	Modified	2.22 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
09. Jan 16	15:56:44	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	15:56:44	Modified	2.22 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
09. Jan 16	15:56:44	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
09. Jan 16	15:56:44	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C

Abbildung 4: Schreiben auf einem anderen Gerät

	Phase	**Beschreibung**
Bild-Synchronisation	Gelb	Die Synchronisation des Bildes aus der Cloud in den lokalen Windowsclient führte zu Schreiboperationen in zwei Dateien, analog der Beschreibung unter 2.5.1.3. Ebenfalls kam es zu Updates der Ordnerinformation.

	Orange	Im Anschluss an die Dateioperationen konnte das Anlegen und Verändern der Datei map1 beobachtet werden. Nach Abschluss der Modifikationen der Datei map1 wurde die Datei map0 gelöscht. Ebenfalls kam es zu Updates der Ordnerinformation.

Tabelle 4: Bildsynchronisation

2.5.1.5 Empfang von Nachrichten von Kommunikationspartnern

Der Empfang von Nachrichten anderer Kommunikationspartner, also „Fremdgeräten“ verhielt sich analog zu dem Empfang von Nachrichten von anderen eigenen Geräten.

Der Empfang von Textnachrichten führte wie unter 2.5.1.4 beschrieben zu keinen Schreiboperationen an Inhaltsdateien.

Interessant war hierbei jedoch, dass Schreiboperationen beobachtet werden konnten, abhängig davon, ob die Nachricht in einem Einzelchat (1:1 zwischen zwei Personen) ankam, oder ob es sich um einen Gruppenchat handelte. Kam die Nachricht in einem Einzelchat, wurde das Profilbild des Kommunikationspartners geladen (sofern nicht mit einer vorhergehenden Nachricht gerade geschehen) und im Ordner temp für 1 Minute und 55 Sekunden zwischengespeichert, wie Abbildung 5: Profilbild entnommen werden kann. Der Ordner temp wurde hierfür, sofern er noch nicht bestand, neu angelegt und blieb auch nach Löschung des Bildes weiterbestehen.

08. Jan 16	17:17:15	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\temp
08. Jan 16	17:17:15	Created	41.21 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\temp\95a7a962431e49f8.png
08. Jan 16	17:17:15	Modified	41.21 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\temp\95a7a962431e49f8.png
08. Jan 16	17:17:15	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\temp
08. Jan 16	17:17:15	Modified	41.21 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\temp\95a7a962431e49f8.png
08. Jan 16	17:19:10	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\temp\95a7a962431e49f8.png
08. Jan 16	17:19:10	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\temp

Abbildung 5: Profilbild

Der Empfang eines Bildes im regulären Chat führte zu Schreiboperationen, die analog dem unter 2.5.1.4 gezeigten Muster verliefen. Daher wird hier auf eine erneute tabellarische Beschreibung verzichtet, da diese dem o.g. Kapitel entnommen werden kann.

09. Jan 16	15:47:55	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\8B6125374B19E1710
09. Jan 16	15:47:55	Modified	1.40 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\8B6125374B19E1710
09. Jan 16	15:47:55	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	15:47:55	Modified	1.40 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\8B6125374B19E1710
09. Jan 16	15:47:55	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\3B28AD12B13C17350
09. Jan 16	15:47:55	Modified	148.65 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\3B28AD12B13C17350
09. Jan 16	15:47:55	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	15:47:55	Modified	148.65 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\3B28AD12B13C17350
09. Jan 16	15:47:56	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
09. Jan 16	15:47:56	Modified	2.11 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
09. Jan 16	15:47:56	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	15:47:56	Modified	2.11 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
09. Jan 16	15:47:56	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
09. Jan 16	15:47:56	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C

Abbildung 6: Bildempfang im Chat

2.5.1.6 Löschen von Nachrichten

Das Löschen einzelner Nachrichten in Chats führte zu keiner Veränderung an Daten.

2.5.1.7 Löschen eines Chatverlaufs

Da Löschen von Chatverläufen führte zu Änderungen. Es wurde eine Inhaltsdatei angelegt (16 Stellen + „0“), Schreiboperationen durchgeführt, die bestehende Inhaltsdatei (erste 16 Stellen gleich + „1“) wurde gelöscht. Anschließend vice-versa.

09. Jan 16	17:38:17	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB0
09. Jan 16	17:38:17	Modified	92 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB0
09. Jan 16	17:38:17	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	17:38:17	Modified	92 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB0
09. Jan 16	17:38:17	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB1
09. Jan 16	17:38:17	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	17:38:18	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB1
09. Jan 16	17:38:18	Modified	108 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB1
09. Jan 16	17:38:18	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	17:38:18	Modified	108 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB1
09. Jan 16	17:38:18	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB0
09. Jan 16	17:38:18	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C

Abbildung 7: Löschen eines Chatverlaufs

2.5.1.8 Löschen eines Chats-Kanals

Das Löschen eines ganzen Chat-Kanals führte ebenfalls zu Veränderungen.

09. Jan 16	17:39:46	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\D09D69BE166617990
09. Jan 16	17:39:46	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	17:39:46	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB0
09. Jan 16	17:39:46	Modified	92 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB0
09. Jan 16	17:39:46	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	17:39:46	Modified	92 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB0
09. Jan 16	17:39:46	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\79060EA1A200F1EB1
09. Jan 16	17:39:46	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	17:39:47	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
09. Jan 16	17:39:47	Modified	2.63 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
09. Jan 16	17:39:47	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
09. Jan 16	17:39:47	Modified	2.63 KB	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
09. Jan 16	17:39:47	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
09. Jan 16	17:39:47	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C

Abbildung 8: Löschen Chat-Kanal

Phase	Beschreibung
Gelb	Hier zeigte sich, dass zunächst eine Inhaltsdatei gelöscht wurde.
Orange	Anschließend wurden Änderungen an einer Inhaltsdatei hierzu vorgenommen, die zu einem Wechsel der letzten Stelle 0 → 1 im Namen gemäß bekanntem Schema von map[0|1] führten.
Grün	Abschließend erfolgten Änderungen an der Datei map[0|1]

Tabelle 5: Phasen der Chat-Kanal-Löschung

2.5.1.9 Löschen des gesamten Nachrichten-Caches

Das Löschen des gesamten Daten-Caches über „Einstellungen > lokaler Cache > leeren“ führte zu Löschoperationen der Inhaltsdateien.

08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\4B19206E36B2C3E70
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\9E59E392B25DBDF10
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\C137CE9A1F6D2CD30
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\527831B058C743080
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\CBFA3E709535C7120
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\8EC2F2E30CCD30E80
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\CB526128817D56AD0
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\266CCC8657265B3D0
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\1DFCDD9109FFBAD90
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\2F9286C4496201820
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\706EDAD3195E7CD30
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:13:59	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\F8C2F8A4057309D80
08. Jan 16	16:13:59	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:14:00	Created	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
08. Jan 16	16:14:00	Modified	452 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
08. Jan 16	16:14:00	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
08. Jan 16	16:14:00	Modified	452 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map1
08. Jan 16	16:14:00	Deleted	0 Bytes	---	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C\map0
08. Jan 16	16:14:00	Modified	0 Bytes	m16user	C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C

Abbildung 9: löschen des lokalen Chats

Die Überprüfung des Inhaltsdatenordners in tdata zeigte nun folgenden Inhalt.

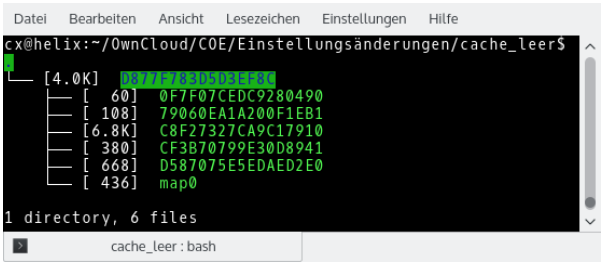


Abbildung 10: Inhalt des tdata-Ordners nach Cache-Löschung

***Anmerkung**:*

Diese Informationen und der Vergleich bspw. mit den Daten der Schreiboperationen aus 2.5.1.8 zeigt, dass bspw. die Datei 79060EA1A200F1EB[0|1] noch vorhanden ist und legt die Vermutung nahe, dass es sich bei den Verbliebenen um Dateien handelt, die Einstellungen und Metadaten speichern. Diese These konnte im Rahmen der Aufgabenstellung nicht für alle Dateien nachgewiesen werden (Ausnahme siehe 2.5.1.11). Hierfür ist die Analyse des Quellcodes oder Reverse Engineering der Applikation notwendig.

2.5.1.10 Map[0|1]

Nach der Löschung des lokalen Caches und den Beobachtungen aus den vorherigen Unterkapiteln in 2.5.1 Datenspeicherung zeigte sich, dass die Datei map[0|1] nicht zu den Chat-Inhaltsdateien gehört, sondern eine andere Funktion wahrnimmt. Bei Schreiboperationen auf der Datei map[0|1] änderte sich der Hashwert, die Größe variierte zwischen 2020 und 2068 Byte, wuchs jedoch nicht weiter.

Für eine Änderung an der Datei map[0|1] wurde bspw. bei Vorliegen der Datei map0 zunächst eine Datei map1 erzeugt und die Änderungen dort hineingeschrieben. Anschließend wurde die Datei map0 gelöscht.

Bei einer erneuten Änderung, die eine Aktualisierung der Datei map1 erforderlich machte, wurde nun zunächst eine Datei map0 angelegt, die Daten dort hineingeschrieben und abschließend wurde die Datei map1 gelöscht.

Der beschriebene Verlauf kann bspw. Abbildung 3: Schreiben und Versand einer Nachricht entnommen werden.

Anmerkung:

Die Einbeziehung der Datei map[0|1] in einer Schreiboperation bei nahezu jeder Aktion innerhalb der Applikation führte zu der These, dass die Datei map[0|1] für die Speicherung von Metadaten, bspw. analog einem Index für die Zuordnung von Inhaltsdateien, Einstellungen usw. zu Dateien genutzt wird. Diese These konnte im Rahmen der Aufgabenstellung jedoch nicht bewiesen werden. Hierfür ist die Analyse des Quellcodes oder Reverse Engineering der Applikation notwendig.

2.5.1.11 Ändern von Einstellungen

Das Ändern von Einstellungen im Menü Einstellungen führte, abhängig vom Einstellungsabschnitt, zu verschiedenen Speicherorten.

Einstellungen die in den Abschnitten **Allgemein** (bspw. Sprache, Auto-Updates, Symbol im System-Tray, usw.), **Größe ändern** oder **Erweitert > Verbindungsart** (bspw. IPv6) wurden in die Datei C:\Users\m16user\AppData\Roaming\Telegram Desktop\tdata\settings[0|1] geschrieben, analog dem Vorgehen bei Schreiboperationen in map[0|1]. Ebenso wurde die Positionierung des Fensters auf dem Bildschirm und die Größe für den nächsten Systemstart in der Datei settings[0|1] gespeichert, wie anhand von Verschieben und Größenänderung während des Monitoring beobachtet werden konnte.

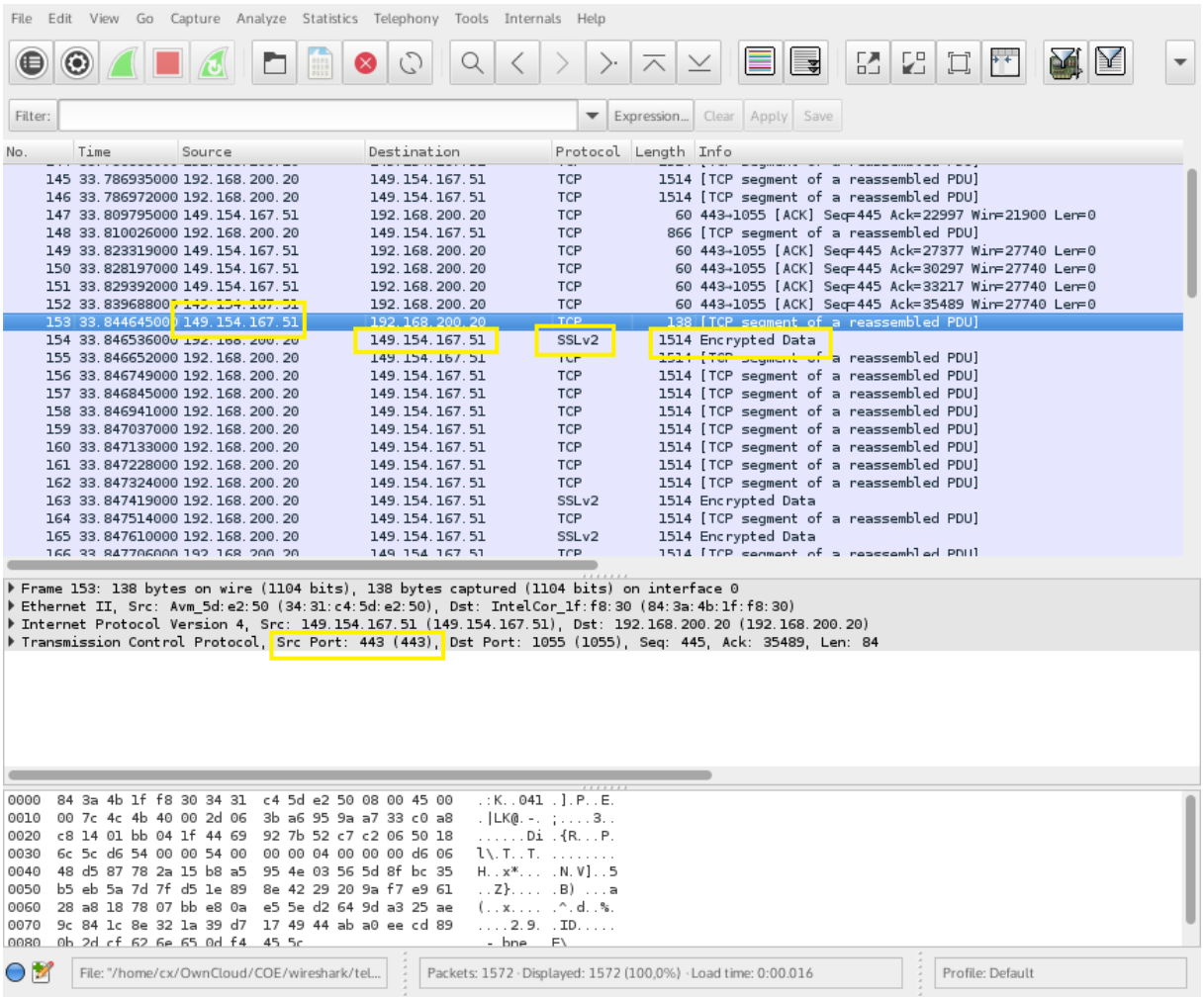
Die restlichen Einstellungen wurden in einer Inhaltsdatei im Ordner D877F783D5D3EF8C, wie sie unter Abbildung 10: Inhalt des tdata-Ordners nach Cache-Löschung in 2.5.1.9 gelistet sind, geschrieben. Im konkreten Fall wurden Änderungen aus den anderen Abschnitten der Einstellungen in der Datei D587075E5EDAED2E[0|1] gespeichert. Diese Datei wurde jedoch bei unterschiedlichen Test-Installationen anders benannt und ist daher hier nur exemplarisch genannt.

2.5.2 Datenübertragung

Auffällig war, dass eine Nutzung der Applikation ohne bestehende Internetverbindung nicht möglich war. Die Applikation bot einen leeren Startbildschirm, der in wachsenden zeitlichen Abständen (jeweils Verlängerung des Zeitraums um Faktor 2) versuchte einen Connect zu der Webseite herzustellen.

Bei der Analyse der Datenübertragung bei Aufruf und Nutzung des Programms zeigte sich, dass ein Aufbau der Datenkommunikation zu der IP-Addresse 149.154.167.51 (RIPE-Zuordnung: Telegram Messenger Amsterdam Network) stattfand[9]. Die Zuordnung des Messenger erfolgt zu dem nächstgelegenen Datacenter, dass die Applikation selbst ermittelt. Weitere sind weltweit verteilt.[10]

Eine Analyse des Verkehrs mittels Wireshark zeigte schnell, dass die Übertragung der Daten mittels SSL/TLS-gesicherter Verbindung stattfand.



2.6 Szenarien zur Inhalts-Analyse / Angriffe

Diese Überlegungen wurden angestellt, da es sicherlich im Rahmen der forensischen Arbeit – über den Nachweis von Spuren der Nutzung – auch von Interesse für Ermittlungsbehörden sein dürfte, an die Inhaltsdaten des Messengers zu gelangen. Der Fokus liegt, gemäß der Aufgabenstellung eng begrenzt auf den Windows-Client.

[9] Siehe Anlage 2
[10] (Durov & Durov, Datacenter, 2016)

2.6.1 Auslesen von Textnachrichten

Das Auslesen von Textnachrichten eines Chats ist auf Basis der lokal gespeicherten Daten nicht möglich, da diese Daten, wie bspw. Kapitel 2.5.1.5 zu entnehmen, nicht auf das Windowssystem synchronisiert werden.

Das Erstellen von Nachrichten führte, wie unter Kapitel 2.5.1.3 ersichtlich, lediglich zu einer kurzen Zwischenspeicherung der Informationen in einer Inhaltsdatei. Diese wurde aber nach Versand der Nachricht wieder gelöscht. Es wäre zwar grundsätzlich möglich (sofern die Dateien nicht überschrieben wurden), diese im Rahmen einer forensischen Festplattenuntersuchung wiederherzustellen, aber dann würden sie zunächst auch nur in Ihrer verschlüsselten Form vorliegen.

Die Untersuchung der Dateien beschränkt sich daher lediglich auf die lokal zwischengespeicherten Bilder bzw. Dateien in ebenfalls verschlüsselter Form, die demzufolge in den Inhaltsdateien abgelegt werden. Hierfür ist jedoch zunächst die Analyse des Quellcodes bzw. ein Reverse-Engineering der Applikation notwendig, auf welches jedoch im Rahmen der Arbeit verzichtet wurde, da dies nicht primär im Aufgabenfokus stand.

Bei der Recherche im Internet zeigte sich kein kommerzielles Produkt[11], dass eine Entschlüsselung der Inhaltsdaten des Messengers beherrschte.

2.6.2 Analyse der Cloud-Daten

Da, wie zuvor geschrieben, keine lokale Synchronisation der Textnachrichten auf den Endgeräten unter Windows erfolgt, wäre eine Möglichkeit die Beschaffung der Daten, die auf dem Server gespeichert wurden.

Mittels webbasierten Zugriff auf die Cloud-Daten wäre es ohne weiteres möglich, die entsprechenden Chatverläufe zu sichten. Schwierig gestaltet sich in diesem Rahmen jedoch die Sicherung der Chatverläufe, da diese nicht auf einfache Weise gesichert werden können. Möglich wäre ein entsprechendes „Herauskopieren“ bzw. Screenshots. Wichtig hierbei ist die Dokumentation des aktuellen Status für Nachuntersuchungen, falls zur Zeit der Untersuchung oder danach weitere Nachrichten eingehen bzw. diese geändert werden.

Darüber hinaus müsste sichergestellt werden, dass der Benutzer zwischenzeitlich nicht noch die Möglichkeit eines anderen Zugriffsclients bspw. über ein Mobilfunkgerät besitzt, da er sonst Einfluss auf die Chatverläufe (bspw. löschen von Nachrichten) nehmen könnte.

Ebenso ist zu beachten, dass, wie unter 2.4 EXKURS „geheimer Chat“ betrachtet, geheime Chats nicht auf dem Server gespeichert werden.

2.6.3 Quellcode-Analyse und Reverse Engineering

Im Rahmen der Untersuchung des Windows-Clients bestünde noch die Möglichkeit, die Programmfunktionen, insbesondere die Verschlüsselung der Daten auf der lokalen Festplatte mittels Quellcode-

[11] Hier beispielhaft zwei der in Deutschland am weitesten verbreiteten Systeme (X-Ways Software Technology AG, 2016) und (Guidance Software, Inc., 2016) angegeben.

Analyse oder Reverse Engineering zu untersuchen. Dies hätte aber in einer ausreichenden Qualität, Vollständigkeit und Güte den Rahmen dieser Arbeit und der Aufgabenstellung gesprengt[12].

3 Zusammenfassung Ergebnisse

Im Rahmen der Analyse zeigte sich, dass die relevanten Daten der Applikation – neben den üblichen Einträgen im Windows-Startmenü und der Liste der installierten „Apps“ unter Windows 10 – in einem Ordner des Benutzers abgelegt wurden. Dort befanden sich sowohl die Einstellungs-/Konfigurationsdateien, die Applikation selbst, sowie die Inhaltsdaten der Informationen, die auf das lokale Gerät synchronisiert wurden.

Die Benennung der Ordner und der meisten Metadaten- und Inhaltsdatendateien mit 16 Stellen aus dem hexadezimalen Zeichenraum und einer abschließenden 0 bzw. 1 können als erste einfache Obfuscation-Maßnahmen, neben der Verschlüsselung, gewertet werden. Außer dem Ordner D877F783D5D3EF8C änderten sich die Bezeichnung der Metadaten- und Inhaltsdatendateien zwischen verschiedenen Installationen und lassen so zunächst keine Rückschlüsse auf Ihre Funktion zu.

Die Nutzung des Clients zu Testzwecken auf einem Linux-System des Verfassers zeigte dort einen ähnlichen Aufbau der Struktur und legt daher die Vermutung nahe, dass die gewonnenen Erkenntnisse in leicht abgewandelter Form auf den für Telegram zur Verfügung stehenden Plattformen in der Praxis angewandt werden können.

Tiefere Kenntnisse der Applikation bringt nun nur noch die Analyse des Quell-Codes bzw. ein Reverse-Engineering der Applikation.

[12] (Telegram, 2016), mehr als 100.000 lines of code.

4 Quellenverzeichnis

Canonical. (2016). *Kubuntu*. Abgerufen am 02. Januar 2016 von http://www.kubuntu.org/

Durov, P., & Durov, N. (2016). *Datacenter*. Abgerufen am 06. Januar 2016 von https://core.telegram.org/api/datacenter

Durov, P., & Durov, N. (2016). *Telegram*. Abgerufen am 04. Januar 2016 von https://telegram.org/

Guidance Software, Inc. (2016). *EnCase® Forensic*. Abgerufen am 08. Januar 2016 von https://www.guidancesoftware.com/encase-forensic?cmpid=nav_r

Oracle. (2016). *Virtual Box*. Abgerufen am 03. Januar 2016 von https://www.virtualbox.org/

Telegram. (2016). *GitHub*. Abgerufen am 08. 12 2015 von https://github.com/telegramdesktop/tdesktop

Ubuntuusers. (2016). *Telegram*. Abgerufen am 08. Januar 2016 von https://wiki.ubuntuusers.de/Telegram/

WhatsApp Inc . (2016). *WhatsApp*. Abgerufen am 03. Januar 2016 von https://www.whatsapp.com/

Wireshark Foundation. (2015). *Wireshark*. Abgerufen am 17. 12 2015 von https://www.wireshark.org/

X-Ways Software Technology AG. (2016). Abgerufen am 09. Januar 2016 von http://www.x-ways.net/