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)

Forensische Untersuchung der Anwendung „ownCloud Client 2.2.4 (6408)“ unter Microsoft Windows

Daniel Bläser

12.02.2017

Technischer Bericht Nr. 11

Zusammenfassung

Synchronisationsdienste für Dateien erfreuen sich großer Beliebtheit und sind im privaten wie im beruflichen Umfeld im Einsatz. Da Anwender zunehmend Wert auf Datenschutz legen und selbst entscheiden möchten, wo die Daten gehostet werden, nimmt die Zahl der ownCloud Installationen, sei es bei einem Service Provider oder privat zu Hause, stetig zu. Die unter Laborbedingungen durchgeführte Analyse soll darstellen, welche Artefakte bei der Nutzung des ownCloud Client unter Windows erzeugt werden und auch nach der Deinstallation bestehen bleiben.

Entstanden im Rahmen des Moduls Browser- und Anwendungsforensik des Studiengangs Digitale Forensik im Wintersemester 2016/2017 unter der Anleitung von Felix Freiling, Holger Morgenstern und Michael Gruhn.

Inhaltsverzeichnis

1 Einführung

Der vorliegende Technische Bericht beschreibt die Ergebnisse einer im Rahmen des Studienganges Digitale Forensik im Modul Browser und Anwendungsforensik durchgeführten Analyse der Anwendung ownCloud Client für das Betriebssystem Microsoft Windows. Analysiert wurde die Version 2.2.4.6408, die von folgender URL heruntergeladen wurde: https://download.owncloud.com/desktop/stable/ownCloud-2.2.4.6408-setup.exe

1.1 Aufgabenstellung

Die Ergebnisse dieser Untersuchung sollen anderen forensisch interessierten Personen eine Hilfestellung bei der Analyse dieser Applikation sein. Dabei sollen im Wesentlichen zwei Fragestellungen untersucht werden:

1) Welche persistenten Spuren hinterlässt die Anwendung im Dateisystem?
2) Wie kann man diese Spuren auswerten?

1.2 Aufbau

Die vorliegende Arbeit gliedert sich in vier Kapitel. In diesem ersten Abschnitt wird die Aufgabenstellung und der Kontext erläutert, sowie die verwendete Arbeitsumgebung beschrieben. Im zweiten Kapitel wird die im Rahmen der Arbeit analysierte Anwendung ownCloud Client veranschaulicht. Hierfür werden Darstellungen der Oberfläche verwendet und beispielhaft ein Nutzungsfall aufgezeigt. Im dritten Kapitel werden die Ergebnisse der technischen Analyse beschrieben, wobei hierzu die verwendeten Tools und Techniken näher erläutert werden, um die Nachvollziehbarkeit zu gewährleisten. Im letzten Kapitel werden die Analyseergebnisse abschließend zusammengefasst.

1.3 Arbeitsumgebung

Als Arbeitsumgebung wird als Hostsystem ein Lenovo T450s mit Windows 10 Education 64bit verwendet. Darauf laufen verschiedene Virtuelle Maschinen um die Analyse durchzuführen:

Lfd. Nr.	Name	Betriebssystem	Beschreibung
1	m117_win7	Windows 7 64bit	In dieser VM wird die Anwendung installiert und alle Aktionen ausgeführt, deren Spuren später analysiert werden.
2	m117_fiwalk	Ubuntu 14.04 LTS 32bit	Über diese VM werden mittels idifference2.py Datenträgerabbilder verglichen

3	owncloudjail[1]	FreeNAS 9.10.2-STABLE	In dieser VM, die auf einem FreeNAS[2] Hostsystem des Autors gehostet wird, läuft die ownCloud Instanz zu der die Verbindung aus der m117_win7 VM aufgebaut wird.

Tabelle 1 - Übersicht der verwendeten virtuellen Maschinen

Für die Analyse der durch die Anwendung in der VM hervorgerufenen Veränderungen werden die beiden Programme RegShot v1.9.0[3] und ProcMon v3.31[4] eingesetzt. RegShot erlaubt es, ein zwei Abbilder der Windows Registry nach der Zustandsmethode zu erzeugen, diese miteinander zu vergleichen und das Ergebnis als Textdatei zu exportieren. ProcMon verwendet die Ereignismethode und „hängt sich“ an die zu überwachende Anwendung an. Dabei werden alle von der Anwendung erzeugten, veränderten, gelöschten oder gelesenen Dateien protokolliert. Das Protokoll kann sehr umfangreich werden, daher muss hier geschickt gefiltert werden.

Für die Analyse der im Dateisystem stattfindenden Veränderungen wird vor einer Aktion ein Snapshot der VM erstellt. Nach Ausführung der Aktion wird ein weiterer Snapshot erstellt und die von beiden Snapshots extrahierten RAW-Festplattenabbilder werden anschließend mittels idifference2.py verglichen. Dieser Vergleich findet in der zweiten VM, „m117_fiwalk“ per automatisiertem Skript statt.

[1] Die ownCloud Serverinstanz läuft nicht auf dem Analyserechner, sondern auf dem FreeNAS Server des Autors, einer dedizierten Maschine, die aus dem Internet per HTTPS erreichbar ist.
[2] http://www.freenas.org/
[3] https://sourceforge.net/projects/regshot/
[4] https://technet.microsoft.com/en-us/sysinternals/bb896645

2 Überblick OwnCloud Client

Die Software „ownCloud Client“ ist eine Anwendung zum Zugriff auf eine meist privat gehostete Cloud-Datenspeicher-Lösung für sicherheitsbewusste Anwender. Durch die vollständige Kontrolle über den Server ist es dem Anwender möglich zu bestimmen, wer Zugriff auf die gespeicherten Daten hat, die Verschlüsselung selbst zu wählen und sich somit unabhängig von den Diensten und der Kontrolle kommerzieller Anbieter zu machen.

Die Clientanwendung wird dabei zum Zugriff mittels der Benutzerdaten auf die Dateien des Benutzers verwendet, analog zu bekannten Speicherdiensten wie Dropbox oder OneDrive. Die Verbindung zum Server erfolgt per verschlüsselt per HTTPS, das Server-Zertifikat wird dabei geprüft. Die Daten werden erst nach dem Download auf den Rechner des Benutzers entschlüsselt, wenn sie auf dem Server verschlüsselt vorliegen.

Nach der Installation des ownCloud Clients steht dem Benutzer ein neues Icon im System Tray zur Verfügung

Abbildung 1 - ownCloud Client Tray Icon nach Installation

Für die initiale Einrichtung wird ein Benutzer auf dem ownCloud Server benötigt, der vom Admin angelegt wird. In unserem Fall heißt der Benutzer „m117“. Der Server ist unter https://192.168.178.102 erreichbar:

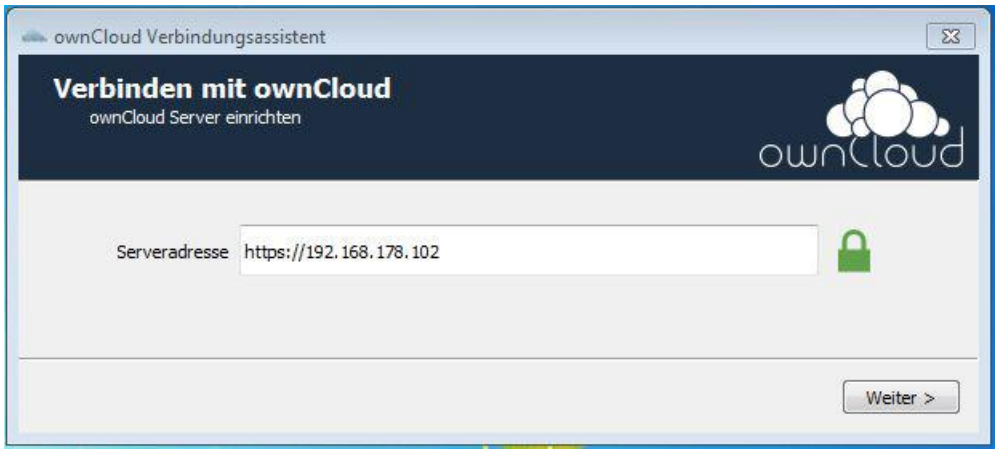

Abbildung 2 - ownCloud Client Installation - Serveradresse

Die nun folgende Zertifikatswarnung ist nur in diesem Testfall vorhanden da normalerweise die korrekte im Zertifikat hinterlegte URL verwendet wird:

Abbildung 3 - ownCloud Client Installation - SSL Zertifikatsfehler

In diesem Fall wird dem Zertifikat trotzdem vertraut. Anschließend folgt die Eingabe der Zugangsdaten:

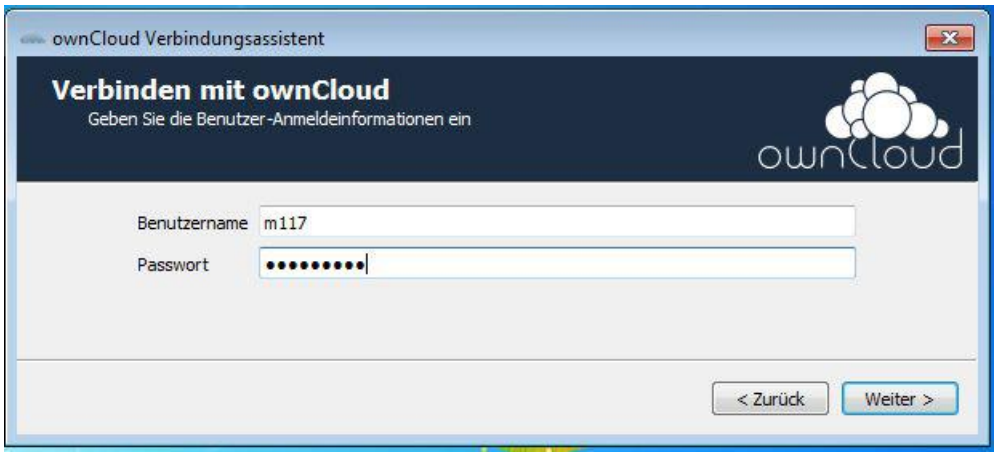

Abbildung 4 - ownCloud Client Installation - Zugangsdaten

Waren die Daten gültig kann nun der zu synchronisierende Ordner gewählt werden, danach ist die Installation abgeschlossen:

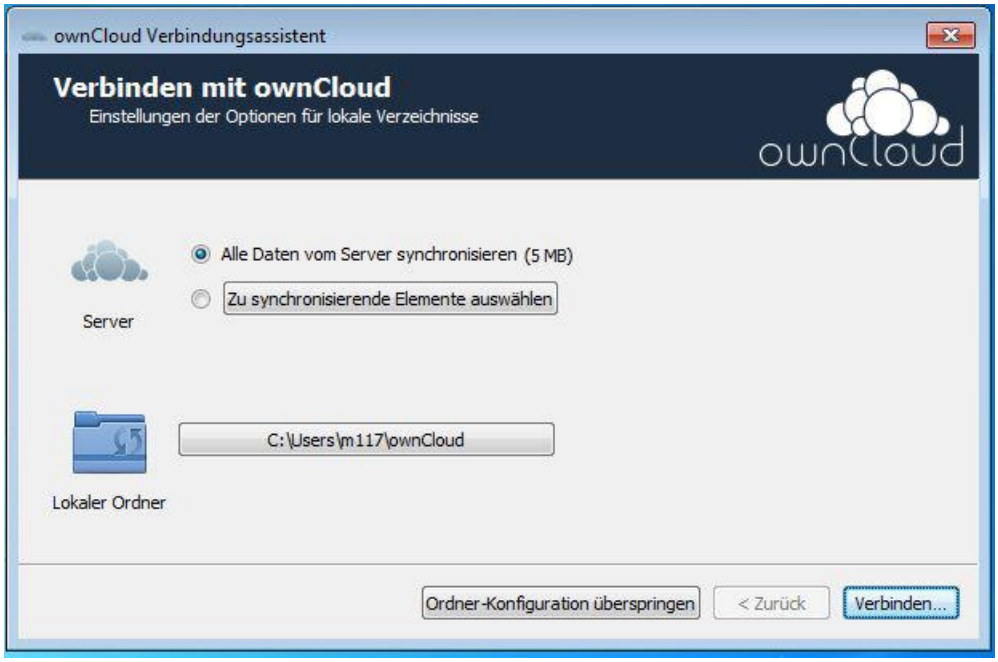

Abbildung 5 - ownCloud Client Installation - Ordnerauswahl

Nach Abschluss der initialen Synchronisierung stehen dem Nutzer nun folgende Möglichkeiten im Kontextmenü zur Verfügung:

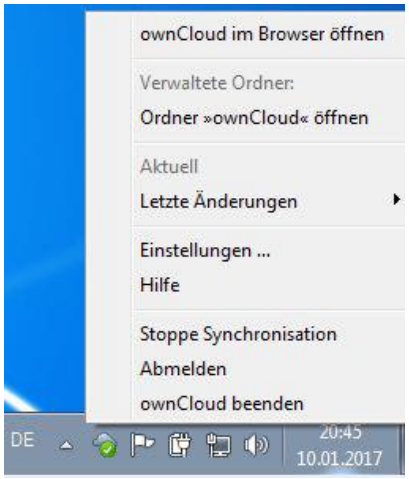

Abbildung 6 - ownCloud Client Trayicon Kontextmenü nach Installation

Die Bedienoberfläche, welche durch einen Linksklick auf das Trayicon geöffnet werden kann, bietet folgende Interaktionsmöglichkeiten:

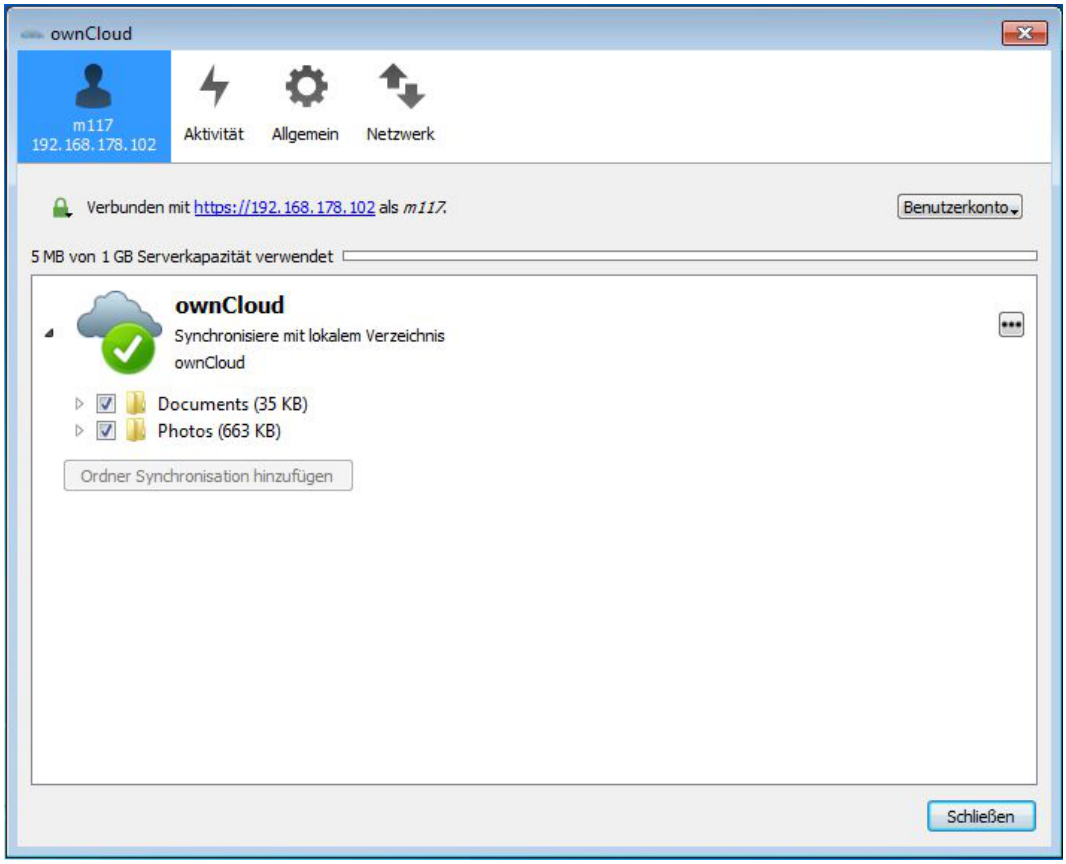

Abbildung 7 - ownCloud Client Oberfläche - Übersicht

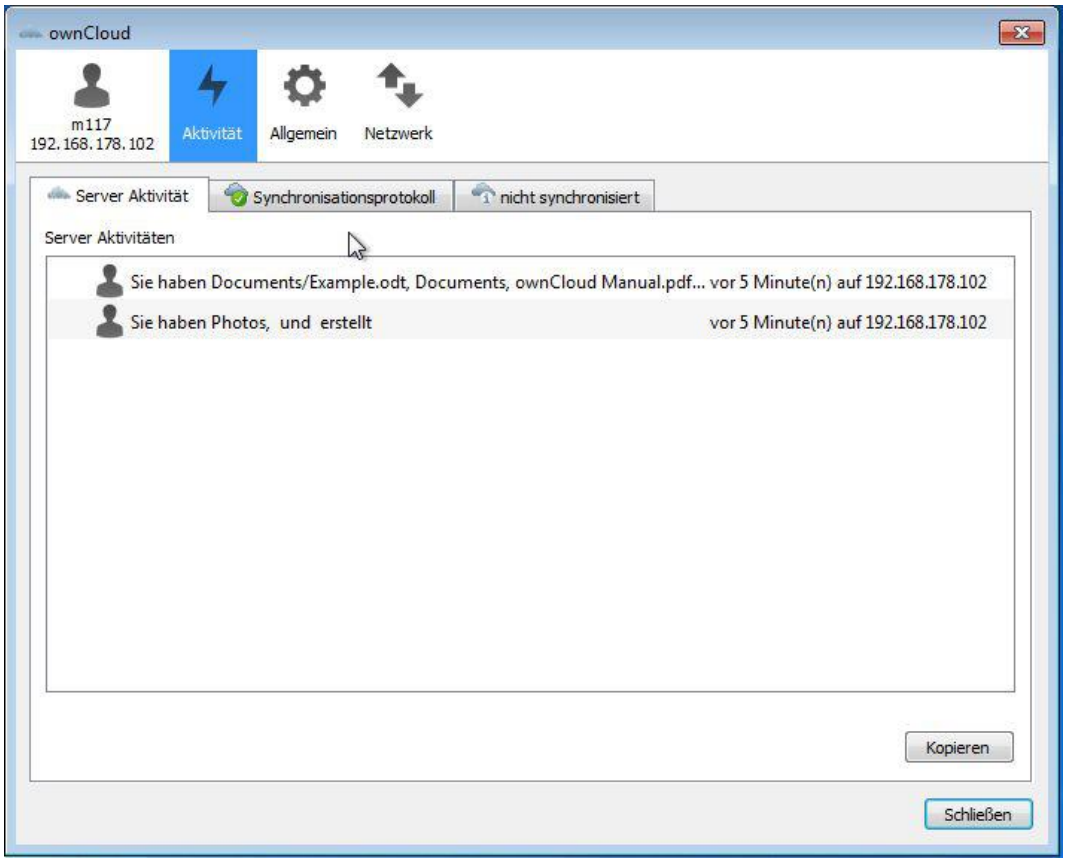

Abbildung 8 - ownCloud Client Oberfläche - Aktivität / Server Aktivität

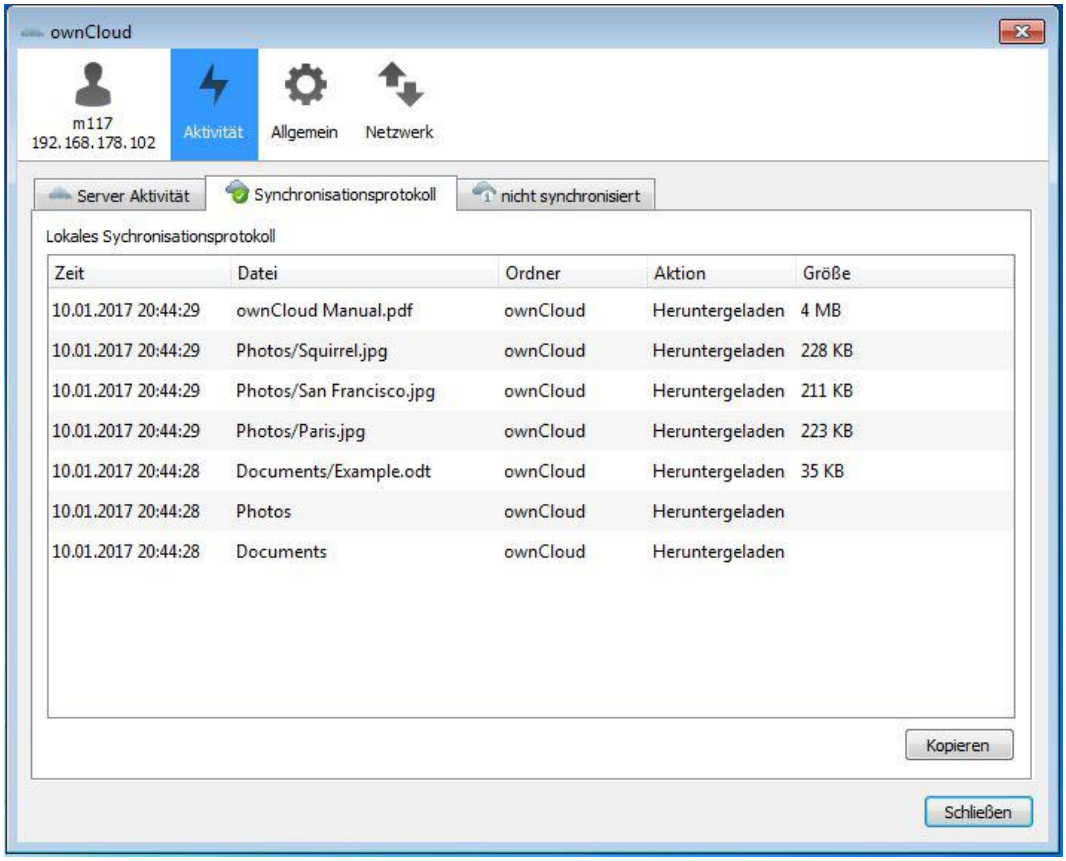

Abbildung 9 - ownCloud Client Oberfläche - Aktivität / Synchronisationsprotokoll

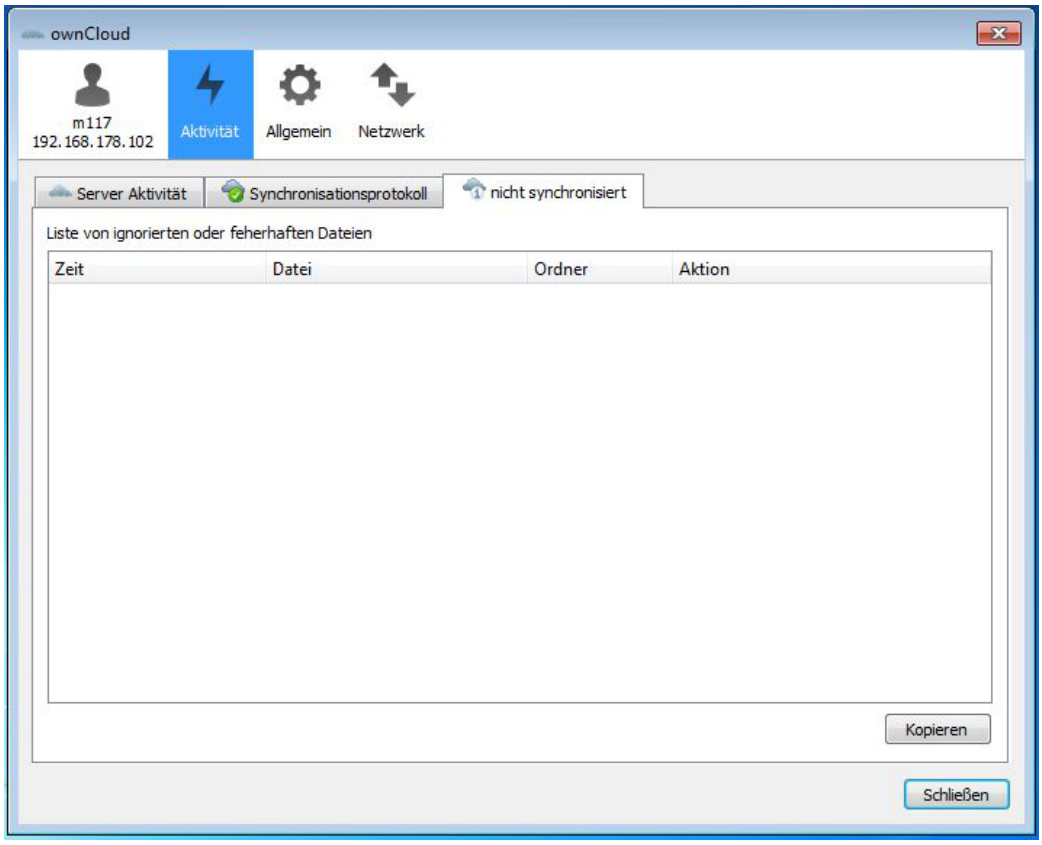

Abbildung 10 - ownCloud Client Oberfläche - Aktivität / nicht synchronisierte Dateien

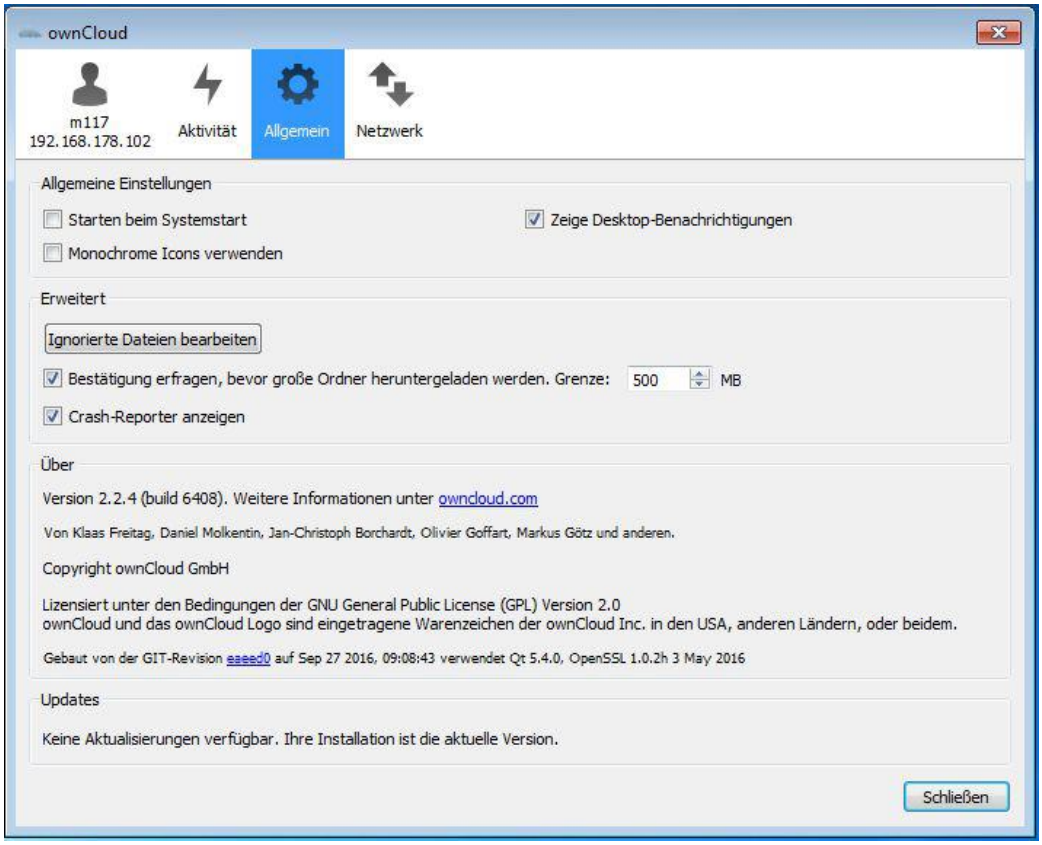

Abbildung 11 - ownCloud Client Oberfläche - Allgemein

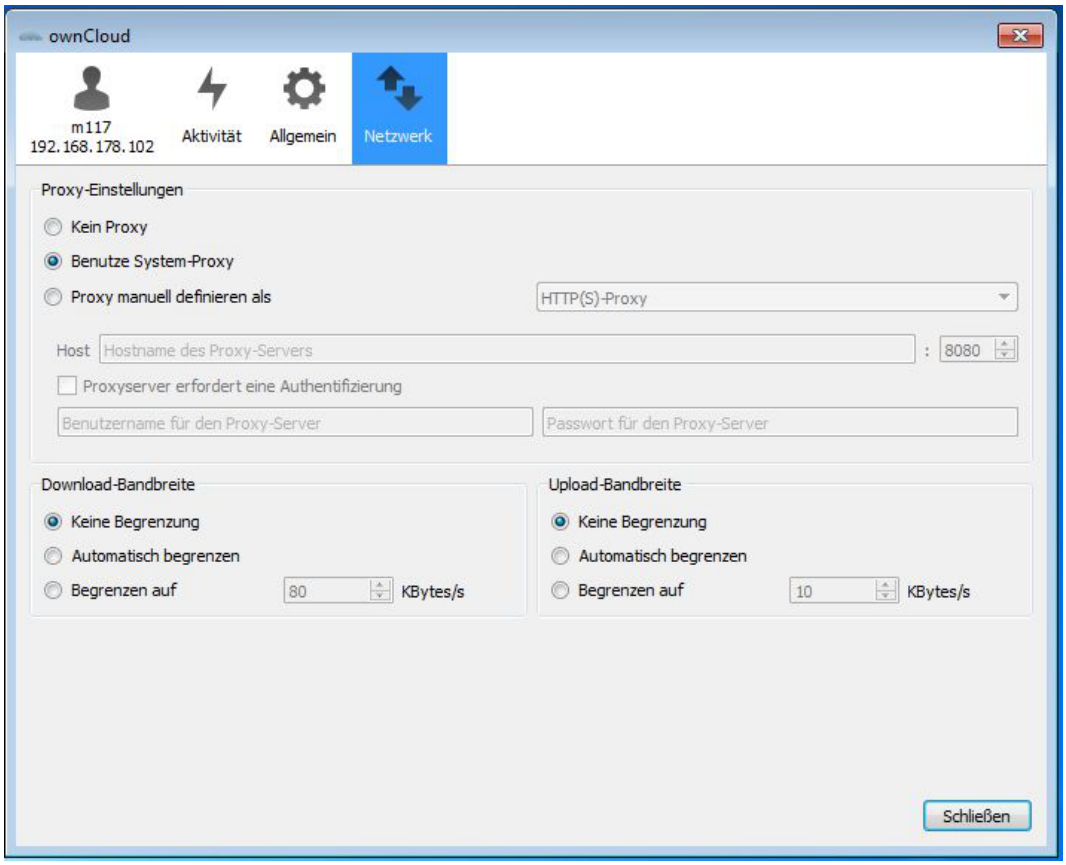

Abbildung 12 - ownCloud Client Oberfläche – Netzwerk

Wird eine Datei synchronisiert, so wird dies in der ownCloud Oberfläche dargestellt:

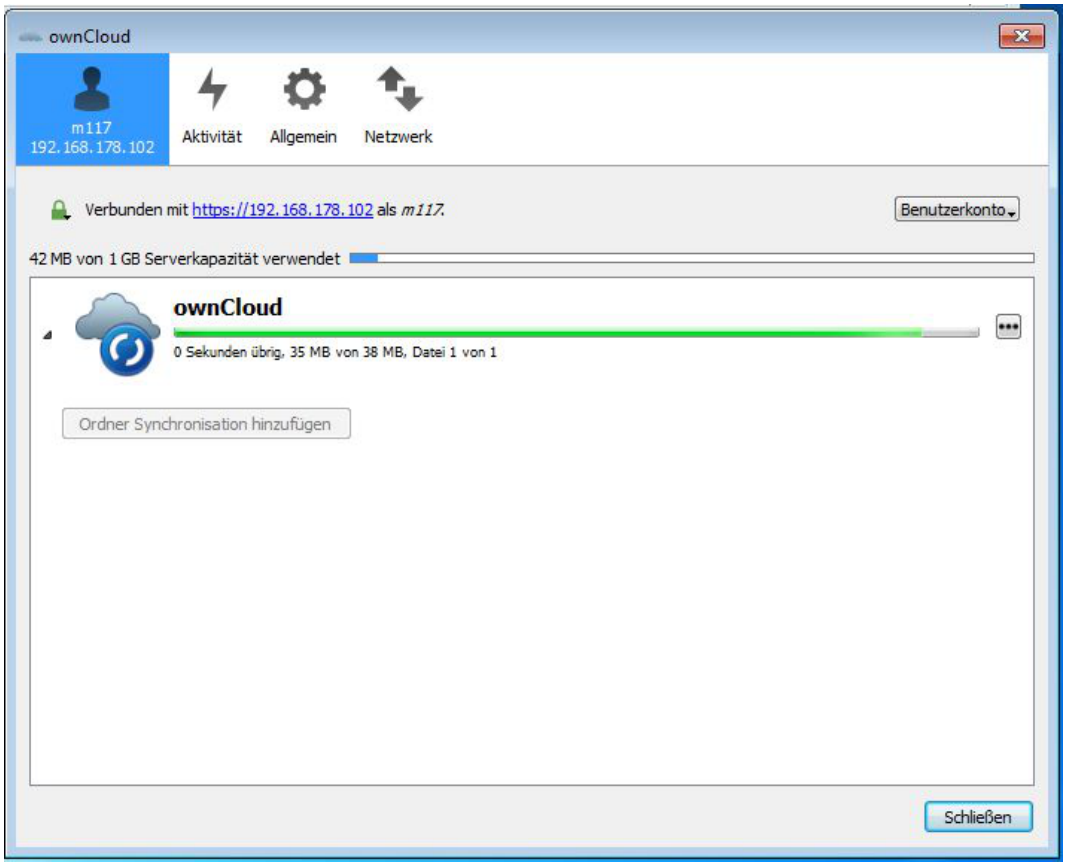

Abbildung 13 - ownCloud Client Oberfläche - Synchronisation einer Datei

Das Tray Icon stellt dies ebenfalls durch einen blauen Doppelpfeil dar:

Abbildung 14 - ownCloud Client Trayicon - Synchronisation einer Datei

3 Technische Analyse

In diesem Kapitel wird die Anwendung technisch analysiert. Um die Nachvollziehbarkeit sicherzustellen wird dafür zunächst das Vorgehen beschrieben. Im Anschluss folgt dann ein Abschnitt in dem die Spuren, die durch die Nutzung der Software entstehen, dargestellt werden.

3.1 Vorgehen

Für die Analyse des ownCloud Clients wird die in Kapitel 1.3 beschriebene Arbeitsumgebung verwendet. Die Untersuchung ist dabei in drei Phasen aufgeteilt:

1. Installation des ownCloud Clients sowie initiale Konfiguration
2. Up- und Download je einer Beispieldatei
3. Deinstallation des Programmes

Jede dieser Phasen wird je Untersuchungsmethode (Procmon, RegShot und idifference) gesondert durchlaufen.

In den Phasen werden folgende Aktionen ausgeführt:

Lfd. Nr.	Phase	Aktionen
1	Installation und initiale Konfiguration	- Installation mithilfe des Installationsprogrammes ownCloud-2.2.4.6408-setup.exe, Standardeinstellungen - Eingeben der Anmeldedaten und der Adresse des Servers[5] - Initiale Konfiguration eines zu synchronisierenden Ordners (Standardeinstellungen)
2	Up- und Download je einer Beispieldaten	- Kopieren einer Beispieldatei in das in Phase 1 eingerichtete Verzeichnis, dadurch Upload der Datei auf den Server[6] - Hinzufügen einer Datei zum Benutzerkonto des hier verwendeten Benutzers über die Weboberfläche der ownCloud, dadurch Download in der VM in das in Phase 1 eingerichtete Verzeichnis.[7]
3	Deinstallation	- Deinstallation des Programmes ownCloud Client über die Systemsteuerung (Programme deinstallieren)

Tabelle 2 - Phasen der technischen Analyse

[5] Hier: Server = 192.168.178.102, Benutzer m117
[6] 30_sync_upload.txt
[7] 30_sync_download.txt

Der Ablaufplan sie wie folgt aus:

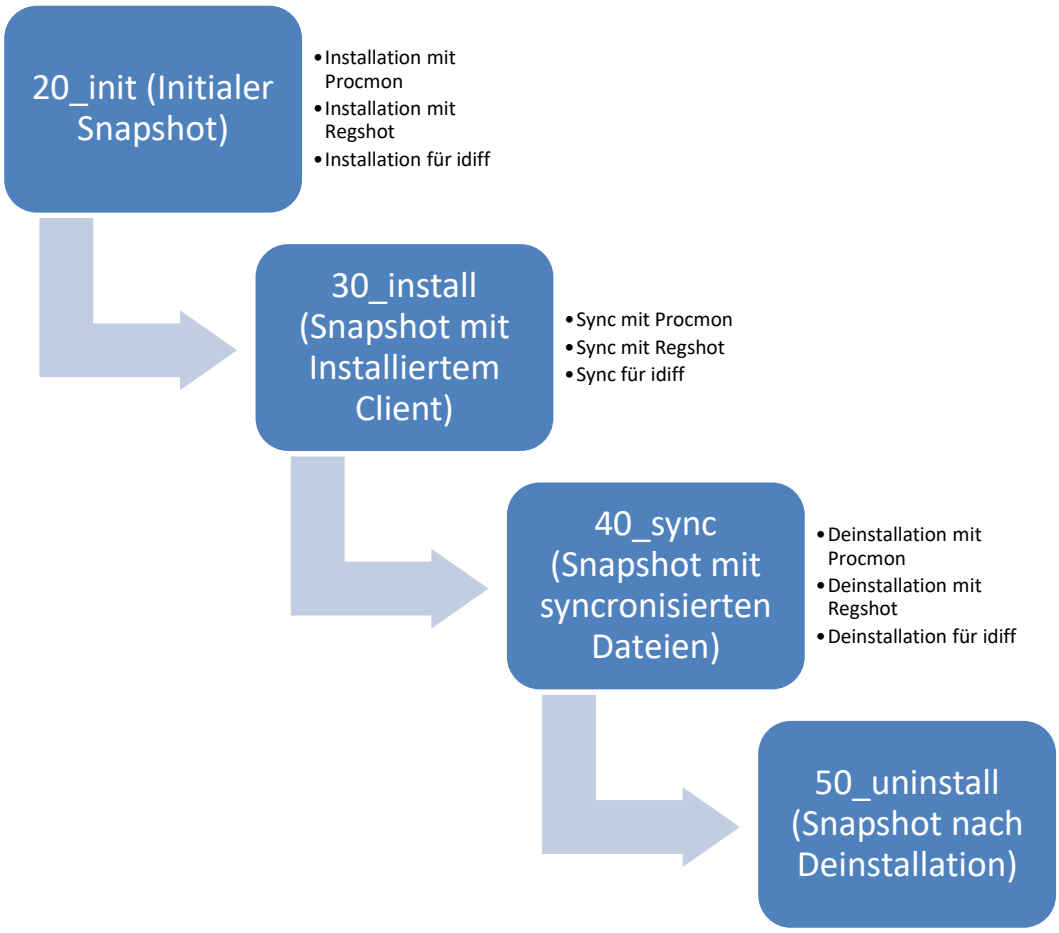

Abbildung 15 - Ablaufplan für die technische Analyse[8]

Nach jeder Ausführung wird die VM auf den Snapshot zurückgesetzt, um gleiche Ausgangsbedingungen sicherzustellen. Nach der dritten Ausführung wird dann ein neuer Snapshot erzeugt, der als Grundlage für die nächste Aktion dient. Durch die Snapshots kann im Falle eines Fehlers einfach auf den vorherigen Zustand zurückgesprungen werden, ohne alle vorherigen Schritte wiederholen zu müssen.

Bei der Analyse mit ProcMon kommt die Ereignismethode zum Einsatz. Dazu wird ProcMon vor der durchzuführenden Aktion gestartet und ein entsprechender Filter auf die Anwendung gesetzt, die nachverfolgt werden soll. Die Aufzeichnung startet automatisch mit dem Start von ProcMon. Nach Ende der Aktion wird die Aufzeichnung gestoppt und das Ergebnis als native PML-Datei exportiert, um darauf spätere Auswertungen zu ermöglichen. Die so generierte Datei wird aus der VM extrahiert

[8] (eigene Darstellung)

und gesichert. Bei der Analyse mit ProcMon wird der Fokus auf Dateisystemoperation wie das Erstellen, Schreiben und Lesen von Dateien gelegt, um die persistente Spurenmenge zu erhalten.

Um die Spuren in der Registry aufzuzeichnen wird die Zustandsmethode verwendet. Hierzu wird mittels der Software RegShot vor Ausführung der Aktion ein Abbild der Windows Registry erstellt. Anschließend wird die Aktion durchgeführt, danach wird das zweite Registryabbild erzeugt und mit dem ersten verglichen. Die Differenz zwischen beiden Abbildern wird als Textdatei gespeichert und ebenfalls aus der VM extrahiert.

Zusätzlich wird nach jeder Aktion ein Datenträgerabbild erstellt, welches im Anschluss mit dem jeweils vorherigen Datenträgerabbild mithilfe des Programmes idifference2.py verglichen wird. Die so erzeugten .idiff Dateien werden später aufbereitet und zeigen jegliche Veränderungen am Dateisystem auf. Die so gewonnenen Spuren werden mittels der durch Promon erlangten Spuren verifiziert.

3.2 Persistente Spurenmenge

Anhand des zuletzt beschriebenen Vorgehens werden in den nächsten Abschnitten die im Dateisystem, der Registry und in den Prefetch Dateien persistierten Spuren ermittelt und beschrieben.

Die gesonderte Betrachtung der Registrierungsdatenbank ist notwendig, weil auf Dateisystemebene nur erkannt werden kann, dass Änderungen stattgefunden haben – jedoch nicht welche.

3.2.1 Dateisystem

Durch die Auswertung der .idiff Dateien lässt sich erkennen, welche Dateisystemoperationen durch eine Aktion auf dem Datenträger verursacht wurden. Da von jeder Aktion ein .idiff besteht, kann die gesamte Kette von Installation über Synchronisation von Dateien und anschließender Deinstallation betrachtet werden. Ebenso ist ein direkter Vergleich der VM vor der Installation mit dem Zustand nach der Deinstallation möglich. Daraus lässt sich ableiten, welche Dateien durch die Deinstallation nicht automatisch entfernt werden – dies sind wichtige Spuren, um eine (ehemalige) Installation der Software nachweisen zu können.

In den folgenden Tabellen werden, nach Phasen gegliedert, die durch den ownCloud Client veränderten Dateien bzw. Pfade aufgezeigt. Hieraus lassen sich Anlaufpunkte für eine manuelle Analyse entnehmen.

3.2.1.1 Spuren aus Phase 1, Installation:

In dieser Phase wird der ownCloud Client in der VM installiert sowie die initiale Konfiguration und die erste Verbindung zum Server durchgeführt:

Lfd. Nr.	Pfad / Datei[9]	Beschreibung	Operation[10]
1	Program Files (x86)/ownCloud/*	Programmverzeichnis des Clients, Standardpfad	CR
2	Program Files (x86)/ownCloud/owncloud.exe	Programmdatei des Clients	CR
3	Users/m117/AppData/Local/ownCloud	Enthält Konfigurationsdateien zum Client	CR
4	Users/m117/AppData/Local/ownCloud/cookies.db	Enthält unter anderem die URL des Servers	CR
5	Users/m117/AppData/Local/ownCloud/owncloud.cfg	Enthält Konfigurationsparameter des lokalen ownCloud Client Installation, u.a. die URL/IP, den User, ein SSL Zertifikat, die Version des Servers, den lokalen Pfad der zu synchronisierenden Ordner, den Status der Synchronisation (pausiert oder laufend).	CR
6	Users/m117/Links/ownCloud.lnk	Verknüpfung, um aus dem Arbeitsplatz auf den lokalen ownCloud Ordner zu gelangen	CR
7	Users/m117/ownCloud	Pfad des lokalen ownCloud Ordners, in den Dateien synchronisiert werden sowie die Logfiles (versteckt) geschrieben werden	CR
8	Users/m117/ownCloud/.csync_journal.db	SQLite Datei über den Inhalt des lokalen Synchronisationsordners. Nach der Aktion Installation sind hier nur Beispieldateien in den Tabellen enthalten	CR
9	Users/m117/ownCloud/.csync_journal.db-wal	Temporäre Datei der SQLite Datenbank Datei Users/m117/ownCloud/.csync_journal.db	CR
10	Users/m117/ownCloud/.csync_journal.db-shm	Temporäre Datei der SQLite Datenbank Datei Users/m117/ownCloud/.csync_journal.db	CR
11	Users/m117/ownCloud/.owncloudsync.log	Logfile über alle Synchronisationen mit dem Server, als lesbarer Plaintext	CR

[9] Alle Pfadangaben beziehen sich auf Laufwerk C:/…
[10] CR = Created, C = Changed, D = Deleted, A = Accessed

12	Users/m117/ownCloud/*	Ordner und Dateien, die synchronisiert wurden, also Dateien des Benutzers	CR
13	Users/Public/Desktop/own-Cloud.lnk	Verknüpfung zum Client auf dem Desktop für alle Windowsbenutzer	CR
14	Windows/Prefetch/OWN-CLOUD.EXE-6ABE8EBF.pf	Prefetch Datei[11] des Clients, wird von Windows angelegt	CR
15	Windows/Prefetch/OWN-CLOUD-2.2.4.6408-SETUP.EXE-6D9FF2D2.pf	Prefetch Datei der Installationsroutine, wird von Windows angelegt	CR

Tabelle 3 - Dateisystemoperationen in Phase 1 - Installation

Die Auswertung der durch ProcMon gefundenen Spuren bestätigt das in Tabelle 3 beschriebene Verhalten.

Weiterhin lässt sich beobachten, dass während der Installation durch das Setup der Software eine weitere Software über den Befehl „"C:\Program Files (x86)\ownCloud\vcredist_x64.exe" /install /quiet“ installiert wird, die zum Betrieb benötigt wird. Von dieser Installation bemerkt der Endanwender in aller Regel nichts, da der Schalter „/quiet“ für eine Ausführung im Hintergrund sorgt. Die Spuren dieser Installation ließen sich auch im idifference Report sehen, waren jedoch nicht eindeutig zuzuordnen wie mit Hilfe von ProcMon.

3.2.1.2 Spuren aus Phase 2, Synchronisation:

Nach der Installation folgt Phase 2, in der zwei Dateien synchronisiert werden. Es Erfolgt jeweils ein Up- sowie Download[12]. Die folgende Tabelle stellt die dabei betroffenen Dateien dar:

Lfd. Nr.	**Pfad / Datei**	**Beschreibung**	**Operation**
1	Users/m117/own-Cloud/.owncloud-sync.log	Logfile der Synchronisationen, Änderungen zur vorherigen Aktion, Inhalt (Auszug): #=#=#=# Syncrun started until (0 msec)	C

[11] Vgl. Kapitel 0
[12] Die Dateinamen lauten: 30_sync_upload.txt und 30_sync_download.txt

		12:46:35|1609|30_sync_up-load.txt|INST_NEW|Up|1484480921||591||4||201|0|0|||INST_NONE| #=#=#=# Syncrun started until (0 msec) 12:47:28|538|30_sync_down-load.txt|INST_NEW|Down|1484484413|443343ea3205181da0a846ec0a30de38|1287|00292471ocllqc8eeagn|4||0|0|0|||INST_NONE|	
2	Users/m117/own-Cloud/30_sync_upload.txt	Hochgeladene Datei, die lokal in den ownCloud Synchronisations-ordner kopiert wurde	CR
3	Users/m117/own-Cloud/30_sync_download.txt	Heruntergeladene Datei, die online in das Benutzerkonto hochgela-den und durch den ownCloud Client heruntergeladen wurde.	CR
4	Users/m117/own-Cloud/.csync_journal.db	SQLite Datenbank des Synchronisationsordners	C
5	Users/m117/own-Cloud/.csync_journal.db-wal	Temporäre Datei der SQLite Datenbank Datei Users/m117/own-Cloud/.csync_journal.db	C
6	Users/m117/own-Cloud/.csync_journal.db-shm	Temporäre Datei der SQLite Datenbank Datei Users/m117/own-Cloud/.csync_journal.db	C
7	Users/m117/App-Data/Local/own-Cloud/own-cloud.cfg	Konfigurationsdatei, auf die während der Synchronisation mehrfach zugegriffen wird	A

Tabelle 4 - Dateisystemoperationen in Phase 2 - Synchronisation

Die Auswertung der durch ProcMon gefundenen Spuren bestätigt das in Tabelle 4 beschriebene Verhalten.

Aus forensischer Sicht ist die .owncloudsync.log Datei die interessanteste Spur. Hier werden sämtliche Synchronisationen der Dateien von und zum Server festgehalten. Dabei wird die Beschreibung, welche Bedeutung die einzelnen Einträge jeder Log Zeile haben, als Titelzeile mitgeliefert. So lässt sich exakt erkennen, wann welche Dateien hoch- und runtergeladen wurden, welche Größe diese hatten, wie lange die Übertragung gedauert hat, wann die Datei zuletzt geändert wurde und ob die Übertragung erfolgreich war oder nicht.

Die im gleichen Verzeichnis liegenden Anwenderdaten (der Inhalt des zu synchronisierenden Ordners) werden durch die Deinstallation nicht gelöscht (siehe Kapitel 3.2.1.3) und sind, falls nicht vom Anwender entfernt, ebenfalls wertvolle Spuren.

3.2.1.3 Spuren aus Phase 3, Deinstallation:

In dieser Phase wird die Clientsoftware wieder deinstalliert:

Lfd. Nr.	Pfad / Datei	Beschreibung	Operation
1	Program Files (x86)/ownCloud/*	ownCloud Client Programmdateien	D
2	ProgramData/Microsoft/Windows/Start Menu/Programs/ownCloud.lnk	Startmenüverknüpfung des Clients	D
3	Users/m117/AppData/Local/ownCloud/owncloud.cfg	Konfigurationsdaten, s.o. Abweichend zur .idiff Analyse ist diese Datei trotz angeblicher Löschung noch vorhanden und kann somit ausgewertet werden! [13]	D
4	Users/Public/Desktop/ownCloud.lnk	Desktopverknüpfung	D
5	Windows/Prefetch/AU_.EXE-28D4D6B8.pf	Prefetchdatei der Deinstallationsroutine	CR

Tabelle 5 - Dateisystemoperationen in Phase 3 - Deinstallation

Nicht gelöscht werden alle Dateien im lokalen Synchronisationsordner. Diese bleiben unverändert bestehen und enthalten somit weiterhin wichtige Spuren, falls der Nutzer diesen Ordner nicht manuell löscht.

Die Auswertung der durch ProcMon gefundenen Spuren bestätigt das in Tabelle 5 beschriebene Verhalten. Ebenso lässt sich eindeutig die Deinstallationsroutine zuordnen:

[13] Dies wurde durch einen erneuten Test verifiziert, auch dabei blieb die .cfg Datei bestehen und wurde nicht entfernt.

Command line: "C:\Users\m117\AppData\Local\Temp\~nsuA.tmp\Au_.exe" _?=C:\Program Files (x86)\ownCloud

Dadurch lässt sich auch Eintrag Nr. 5 erklären, der in den Prefetch Dateien erzeugt wurde.

Forensisch wertvoll ist die Konfigurationsdatei C:/Users/m117[14]/AppData/Local/ownCloud/owncloud.cfg: Hierin befinden sich bspw. folgende Informationen:

[General]

optionalDesktopNotifications=true

[Accounts]

0**Folders\1\localPath=C:/Users/m117/ownCloud/**

version=2

0\Folders\1\targetPath=/

0**url=https://192.168.178.102**

0\Folders\1\paused=false

0**serverVersion=9.1.3.1**

0\Folders\1\ignoreHiddenFiles=true

0\http_certificatePasswd=

0\http_certificatePath=

0**http_user=m117**

0\authType=http

0\user=m117

3.2.2 Registry

In der Windows Registry werden ebenfalls persistente Spuren durch die Installation und die Deinstallation des Clients erzeugt. Diese Spuren werden hier als Überblick dargestellt, wie im vorherigen Abschnitt gegliedert nach Phasen.

Die gefundenen Spuren werden im Vergleich mit den durch ProcMon aufgezeichneten Veränderungen der Registry bestätigt.

[14] Ist durch den Username auf dem analysierten Rechner zu ersetzen

3.2.2.1 Spuren aus Phase 1, Installation:

Hinzugefügte Schlüssel:

- HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\ownCloud
- HKLM\SOFTWARE\Wow6432Node\ownCloud
- HKLM\SOFTWARE\Wow6432Node\ownCloud\ownCloud

Hinzugefügte Werte:

- HKLM\SOFTWARE\Classes\CLSID\{0960F090-F328-48A3-B746-276B1E3C3722}\InprocServer32\: "C:\Program Files (x86)\ownCloud\shellext\...
    - OCOverlays_x64.dll"
    - OCOverlays_x64.dll"
    - OCOverlays_x64.dll"
    - OCOverlays_x64.dll"
    - OCOverlays_x64.dll"
    - OCContextMenu_x64.dll"
- HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\ownCloud\...
    - UninstallString: ""C:\Program Files (x86)\ownCloud\Uninstall.exe""
    - InstallLocation: "C:\Program Files (x86)\ownCloud"
    - DisplayName: "ownCloud"
    - Publisher: "ownCloud"
    - DisplayIcon: "C:\Program Files (x86)\ownCloud\Uninstall.exe,0"
    - DisplayVersion: "2.2.4.6408"
    - VersionMajor: 0x00000002
    - VersionMinor: 0x00000002
    - URLInfoAbout: "http://owncloud.com/"
    - HelpLink: "http://owncloud.com/"
    - NoModify: 0x00000001
    - NoRepair: 0x00000001
    - MementoSectionUsed: ""
    - MementoSection_SEC_SHELL_EXT: 0x00000001
    - MementoSection_SEC_START_MENU: 0x00000001
    - MementoSection_SEC_DESKTOP: 0x00000001
    - MementoSection_SEC_QUICK_LAUNCH: 0x00000000
- HKLM\SOFTWARE\Wow6432Node\ownCloud\ownCloud\: "C:\Program Files (x86)\ownCloud"
- HKLM\SOFTWARE\Wow6432Node\ownCloud\ownCloud\...

  - VersionMajor: 0x00000002
  - VersionMinor: 0x00000002
  - VersionRevision: 0x00000004
  - VersionBuild: 0x00001908
- HKU\S-1-5-21-1479223796-608957520-1148870536-1000\Software\Microsoft\
  - Windows\CurrentVersion\Explorer\StartPage\NewShortcuts\C:\Users\m117\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\ownCloud.lnk: 0x00000001
  - Windows\CurrentVersion\Explorer\StartPage\NewShortcuts\C:\ProgramData\Microsoft\Windows\Start Menu\Programs\ownCloud.lnk: 0x00000001
  - Windows\CurrentVersion\Run\ownCloud: "C:\Program Files (x86)\ownCloud\owncloud.exe"
  - Direct3D\MostRecentApplication\Name: "owncloud.exe"

In der Windows Registry finden sich keine Spuren der Konfiguration des Clients, diese werden ausschließlich in der .cfg Datei gesichert. (vgl. Kapitel 3.2.1.1)

3.2.2.2 Spuren aus Phase 2, Synchronisation

Bei dieser Aktion wurden keine Registryeinträge generiert oder verändert, die auf die Verwendung des ownCloud Clients schließen lassen.

3.2.2.3 Spuren aus Phase 3, Deinstallation

Die entfernten Schlüssel entsprechend weitestgehend denen in Kapitel 3.2.2.1 hinzugefügten Schlüsseln. Nicht entfernt wurde jedoch:

- HKLM\SOFTWARE\Wow6432Node\ownCloud

Die entfernten Werte entsprechend ebenfalls weitestgehend den in Kapitel 3.2.2.1 hinzugefügten Werten. Nicht entfernt wurden jedoch:

- HKU\S-1-5-21-1479223796-608957520-1148870536-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage\NewShortcuts\C:\Users\m117\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\ownCloud.lnk: 0x00000001
- HKU\S-1-5-21-1479223796-608957520-1148870536-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage\NewShortcuts\C:\ProgramData\Microsoft\Windows\Start Menu\Programs\ownCloud.lnk: 0x00000001

- HKU\S-1-5-21-1479223796-608957520-1148870536-1000\Software\Microsoft\Windows\CurrentVersion\Run\ownCloud: "C:\Program Files (x86)\ownCloud\owncloud.exe"
- HKU\S-1-5-21-1479223796-608957520-1148870536-1000\Software\Microsoft\Direct3D\MostRecentApplication\Name: "owncloud.exe"

Hinzugefügt wurde ein Wert:

- HKU\S-1-5-21-1479223796-608957520-1148870536-1000\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Persisted\C:\tools\ownCloud-2.2.4.6408-setup.exe: 0x00000001

Durch die nicht entfernten Werte und Schlüssel sowie den hinzugefügten Wert lässt sich auch nach der Deinstallation des ownCloud Clients nachweisen, dass die Software auf diesem Gerät installiert war, falls die Einträge nicht manuell vom Benutzer entfernt wurden.

3.2.3 Prefetch

Prefetch Dateien gehören zu den robusten Spuren unter Windows, da die meisten Endanwender keine Kenntnis über ihre Existenz haben. Im Verzeichnis C:/Windows/Prefetch legt das Betriebssystem Prefetch Dateien von Programmen an, um den Bootvorgang und die Ladegeschwindigkeit von Programmen zu erhöhen.[15] Im Fall des ownCloud Client sind folgende Prefetch Dateien von Interesse:

FILE NAME	DATE CREATED	DATE MODIFIED	DATE LAST RUN	NUM TIMES RUN	PHYSICAL PATH
AU_.EXE-28D4D6B8.pf	21 Jan 2017 (Sa) 13:38:21	15 Jan 2017 (So) 12:55:49	15 Jan 2017 (So) 12:55:43	1	\DEVICE\HARDDISKVOLUME2\USERS\M117\APPDATA\LOCAL\TEMP\~NSUA.TMP\AU_.EXE
OWNCLOUD.EXE-6ABE8EBF.pf	21 Jan 2017 (Sa) 13:38:22	15 Jan 2017 (So) 11:29:50	15 Jan 2017 (So) 11:29:40	1	\DEVICE\HARDDISKVOLUME2\PROGRAM FILES (X86)\OWNCLOUD\OWNCLOUD.EXE
OWNCLOUD-2.2.4.6408-SETUP.EXE-6D9FF2D2.pf	21 Jan 2017 (Sa) 13:38:22	15 Jan 2017 (So) 11:28:57	15 Jan 2017 (So) 11:28:47	2	

Tabelle 6 - Prefetch Dateien nach der Deinstallation[16]

[15] Vgl. http://www.winfaq.de/faq_html/Content/tip1500/onlinefaq.php?h=tip1502.htm
[16] Analysiert mit http://www.woanware.co.uk/downloads/PrefetchForensics.v.1.0.4.zip

Der erste Eintrag ist die Deinstallationsroutine, mit der der ownCloud Client deinstalliert wurde.

Der zweite Eintrag ist der Client selbst. Auch nach der Deinstallation des Programmes ist die Prefetch Datei noch vorhanden.

Der letzte Eintrag gehört zum Installationspaket des Clients.

Die Auswertung der Prefetch Dateien wurde nach der Deinstallation durchgeführt, da hier die meisten Spuren zu finden waren.

4 Fazit

Die im vorliegenden technischen Bericht beschriebenen Ergebnisse der Anwendungsanalyse zeigen, dass die in verschiedenen Bereichen gefundenen persistenten Spuren der Anwendung ownCloud Client Rückschlüsse darauf zulassen, ob die Anwendung installiert war, mit welchem Server kommuniziert wurde und, falls die Spuren nicht entfernt wurden, auch welche Dateien synchronisiert wurden.

Aus Sicht des Dateisystems liegen auch nach der Deinstallation der Anwendung im Verzeichnis C:/Users/m117/AppData/Local/ownCloud/ die owncloud.cfg Datei vor, welche die Konfiguration des verwendeten Benutzerkontos samt Serveradresse enthält. Weiterhin bleibt auch der Synchronisationsordner unter C:/Users/m117/ownCloud bestehen, wenn er nicht manuell entfernt wird. Hier befinden sich die vom Benutzer synchronisierten Dateien sowie das Logfile und die lokale Datenbank, in der alle Dateien verzeichnet sind.

In der Registry verbleiben nach der Deinstallation des Clients ebenso Schlüssel und Werte zurück, die auf eine vorherige Installation schließen lassen. Hier lassen sich zwar keine Konfigurationsparameter entnehmen, jedoch kann eine Aussage über die (vorherige) Installation der Software getroffen werden.

Die Prefetch Dateien lassen letztlich ebenfalls den Schluss auf eine ehemalige Installation des Clients zu. Durch eine Auswertung dieser Dateien lässt sich der Zeitpunkt der Deinstallation ermitteln.