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)

Analyse der Spurenmenge der Anwendung OpenVPN Client Version 2.3.9 in Microsoft Windows

Simon Jansen

15. Februar 2016

Technischer Bericht Nr. 1

Zusammenfassung

Virtuelle private Netzwerke (VPNs) werden im Bereich der Computerkriminalität vermehrt von Tätern zu Anonymisierungszwecken eingesetzt und erlangen somit Relevanz für digitalforensische Untersuchungen. Die unter Laborbedingungen durchgeführte Analyse der Spurenmenge soll in realen forensischen Untersuchungen helfen, die persistenten Spuren des OpenVPN Clients sowohl identifizieren als auch interpretieren zu können.

Entstanden 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 Einführung

Die vorliegende Dokumentation beschreibt die Ergebnisse einer im Rahmen des Studiums Digitale Forensik im Modul Browser- und Anwedungsforensik durchgeführten Analyse der Spurenmenge einer unbekannten Anwendung. Die im Rahmen dieses Berichts beschriebenen Analyseergebnisse stammen von der Analyse der frei verfügbaren Anwendung OpenVPN Client für das Betriebssystem Microsoft Windows.

In den nachfolgenden Abschnitten werden zunächst die Aufgabenstellung sowie der Aufbau der Dokumentation beschrieben. Anschließen folgt eine eingehende Aufstellung der für die Analyse verwendeten Arbeitsumgebung.

1.1 Aufgabenstellung

Das Ziel der im Rahmen der praktischen Arbeit für das Modul Browser- und Anwendungsforensik durchgeführten Anwendungsanalyse ist die Feststellung der persistenten Spurenmenge der analysierten Anwendung.[1] Die Analyseergebnisse, die unter Laborbedingungen ermittelt wurden, sollen in späteren realen forensischen Analysen helfen, die Spuren einer bestimmten Anwendung zu erkennen und deuten zu können. Im Rahmen der Analyse kommt es daher vor allem auf die Beantwortung der folgenden Fragestellungen an.

1. Welche Sachverhalte der Anwendung kann man wo (im Dateisystem) finden?
2. Wie kann man die Spuren »auslesen«?

Ziel der Analyse ist dabei nicht die Entwicklung eines Werkzeugs, dass die Spuren der Anwendung automatisiert aus einem Datenträger(-abbild) extrahiert und analysiert. Optional ist eine derartige Entwicklung jedoch möglich.

Die Auswahl der zu analysierenden Anwendung ist grundsätzlich den Studierenden überlassen, unterliegt jedoch festgelegten Kriterien. So darf es sich bei der ausgewählten Anwendung um keine zu einfache Applikation handeln und es muss ein forensisches Interesse an der Anwendung bestehen. Durch diese beiden Kriterien wird die Relevanz für reale forensische Untersuchungen gewahrt. Des Weiteren muss die gewählte Anwendung ausreichend persistente Spuren erzeugen, die analysiert werden können. So ist beispielsweise die Anwendung `ping` auszuschließen, da diese ausschließlich volatile Spuren im Hauptspeicher erzeugt, jedoch keine persistente Spuren bei der Ausführung existieren. Abschließend ist als Auswahlkriterium definiert, dass es sich bei der gewählten Anwendung nicht um eine bereits ausreichend analysierte Anwendung handelt. Zu diesen aus forensischer Sicht ausreichend analysierten Anwendungen zählen u.a. gängige Webbrowser wie Mozilla Firefox und Google Chrome sowie der E-Mail Client Mozilla Thunderbird.

Die Dokumentation der analysierten Spurenmenge soll in einer für einen Forensiker nützlichen Form erfolgen. Die Dokumentation sollte gemäß Aufgabenstellung mindestens eine Liste der von der

[1] Die Aufgabenstellung wurde im Rahmen des ersten Online-Meetings vom Modul 16 Browser- und Anwendungsforensik am 02. November 2015 vorgestellt und kann in den Vortragsfolien des Online-Meetings nachgelesen werden.

Anwendung verwendeten Dateien[2] sowie eine kurze Zusammenfassung, wie die Anwendungsspuren analysiert werden können,[3] enthalten.

1.2 Aufbau

Die vorliegende Dokumentation der Analyseergebnisse gliedert sich in vier Kapitel. Im ersten Kapitel erfolgt eine Einführung in die Thematik sowie die Beschreibung der Aufgabenstellung. Darüber hinaus wird die für die Analyse verwendete Arbeitsumgebung detailliert dargestellt. Dies soll die Nachvollziehbarkeit der Analyse sicherstellen. Im zweiten Kapitel folgt ein kurzer Überblick über die analysierte Anwendung OpenVPN Client. Es werden kurz die Leistungsmerkmale der Anwendung und der Einsatzzweck dargestellt. Dies dient der Einordnung der Anwendung und gibt eine erste Einführung in den Funktionsumfang. Weiterhin wird in diesem Kapitel die forensische Relevanz der Anwendung motiviert. Das dritte Kapitel bildet den Hauptteil der Dokumentation, in dem die eigentlichen Analyseergebnisse beschrieben werden. Dazu wird zunächst zum Zweck der Nachvollziehbarkeit und zur Bewertung der Analysemethodik das Vorgehen bei der Analyse dargestellt. Anschließend wird die analysierte Spurenmenge beschrieben. Die Beschreibung richtet sich an einen typischen Arbeitsablauf beim Umgang mit der Anwendung. In dem darauf folgenden vierten Kapitel werden die Analyseergebnisse abschließend zusammengefasst.

1.3 Arbeitsumgebung

Als Arbeitsumgebung wird eine virtuelle Umgebung basierend auf dem Typ-2-Hypervisor[4] VMware Fusion Version 8.0.2 (3164312) eingesetzt. Als Hostsystem wird ein MacBook Pro Modell A1398 EMC 2881, das mit dem Betriebssystem Mac OS X 10.11 (15A284) betrieben wird, eingesetzt. Innerhalb der virtuellen Umgebung existieren mehrere virtuelle Maschinen, die für die Analyse genutzt werden. Tabelle 1.1 gibt einen Überblick über die genutzten virtuellen Maschinen.

In den virtuellen Maschinen werden verschiedene Softwarekomponenten für die Akquise der Spuren sowie für die Analyse derselbigen eingesetzt. Die Software für die Akquise der Spuren wird ausschließlich in der Windows 10 Client VM ausgeführt. Dabei wird die Software Process Monitor (ProcMon) von Sysinternals[5] in der Version 3.20 zur Aufzeichnung von Events des Betriebssystems eingesetzt. ProcMon nutzt zur Ermittlung der Events die Hookingtechnik und wird im Rahmen der Analyse zur Umsetzung der Ereignismethode eingesetzt. RegShot Version 1.9.0 dient zur Ermittlung von Änderungen in der Windows Registry mittels Verwendung der Zustandsmethode. Dazu wird vor Durchführung der zu analysierenden Aktion (z.B. Installation vom OpenVPN Client) ein Abbild

[2] Siehe beispielhaft `http://kb.mozillazine.org/Profile_folder_-_Firefox#Files_and_folders_in_the_profile`, abgerufen am 09.01.2016.

[3] Vgl. `http://www.forensicswiki.org/wiki/Mozilla_Firefox`, abgerufen am 09.01.2016.

[4] Bei einem Typ-2-Hypervisor handelt es sich um einen so genannten hosted Hypervisor, der ein vollwertiges Betriebssystem voraussetzt. Der Hypervisor wird in diesem Betriebssystem als eigenständige Anwendung ausgeführt. Siehe `https://de.wikipedia.org/wiki/Hypervisor#Klassifizierung` (abgerufen am 09.01.2016) für weitere Informationen.

[5] Siehe `https://technet.microsoft.com/de-de/sysinternals/bb545021.aspx`, abgerufen am 09.01.2016.

der Registry erzeugt. Nach Durchführung der Aktion wird ein weiteres Abbild der Registry erzeugt und die beiden Abbilder miteinander verglichen. Das Ergebnis des Vergleichs ist eine textbasierte Ergebnisdatei, die die hinzugefügten, veränderten und gelöschten Registry Keys enthält. Zur Erstellung von Datenträgerabbildern, die im weiteren Verlauf zur Erstellung von Zeitleisten herangezogen werden, wird die Software AccessData FTK Imager Lite Version 3.1.1.8 genutzt. Weiterhin wird der bereits erwähnte VMware Fusion Hypervisor zur Erstellung von Snapshots der Windows 10 Client VM nach jedem Analyseschritt verwendet. Neben den Analysewerkzeugen ist die zu untersuchende Anwendung OpenVPN Client[6] Version 2.3.9-I601, der die GUI-Komponente OpenVPN GUI Version 8 enthält, in der VM installiert.

Lfd. Nr.	VM-Name	Betriebssystem	Kurzbeschreibung
1	Windows 10 Client	Windows 10 Pro Version 10.0.10586 Build 10586	Innerhalb der Windows 10 VM wird die Anwendung OpenVPN Client installiert und die Spuren akquiriert.
2	OpenVPN Server	Ubuntu Server 14.04.3 LTS x64 Kernelversion 3.13.0-32-generic	Die Ubuntu Server VM stellt den OpenVPN Server bereit, mit dem sich der in der Windows 10 VM installierten OpenVPN Client verbinden kann.
3	SIFT Workstation	SIFT Workstation Version 3 basierend auf Ubuntu 14.04.1 LTS x64 Kernelversion 3.13.0-44-generic	Die SIFT Workstation von SANS wird zur Erstellung von Zeitleisten auf Basis von Datenträgerabbildern verwendet.

Tabelle 1.1: Übersicht der im Rahmen der Anwendungsanalyse verwendeten virtuellen Maschinen

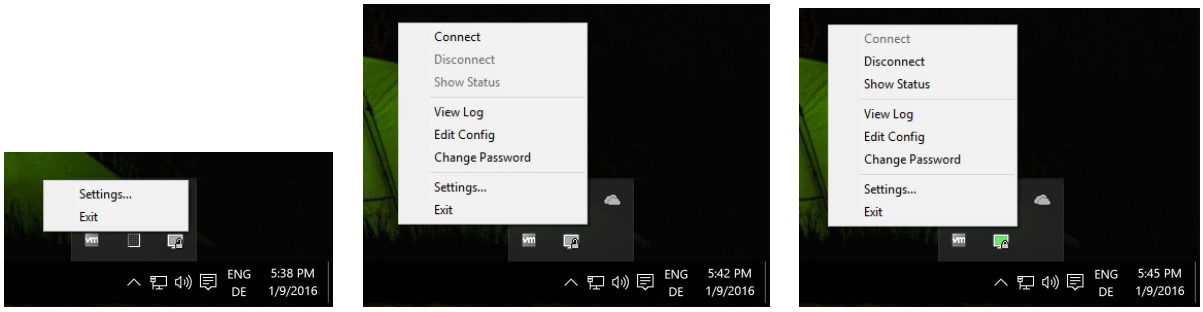
In der OpenVPN Server VM wurde die OpenVPN Serverkomponente mithilfe des Ubuntu Pakets `openvpn` sowie das Paket `easy-rsa` zur Erstellung von asymmetrischen Schlüsselpaaren zusätzlich zur Software, die mit dem Betriebssystem ausgeliefert wird, installiert. Die Konfiguration wurde nach den Vorgaben des Eintrags zu OpenVPN im Ubuntuusers Wiki durchgeführt.[7] Aus Sicht der Netzwerkkonfiguration ist darauf zu achten, dass die OpenVPN Server VM von der Windows 10 Client VM aus erreichbar ist. Dazu eignet sich am besten die Konfiguration einer Bridged-Netzwerkschnittstelle für die Ubuntu Server VM.

Von der SIFT Workstation wird die Software log2timeline Version 0.66 für die Erstellung von Zeitleisten auf Basis der mit FTK Imager erstellten Datenträgerabbilder verwendet. Zur Filterung und Sortierung der umfassenden Zeitleisten wird die Software l2t_process Version 0.2, die ebenfalls Bestandteil der SIFT Workstation ist, genutzt.

[6] Siehe `https://openvpn.net/index.php/open-source/downloads.html`, abgerufen am 09.01.2016
[7] Siehe `http://wiki.ubuntuusers.de/OpenVPN/`, abgerufen am 09.01.2016.

2 Überblick OpenVPN Client

Bei der Anwendung OpenVPN Client handelt es sich um eine clientseitig eingesetzte Software, die zum Aufbau von Virtual Private Network (VPN) Verbindungen zu einem entfernten OpenVPN Server dient. VPN-Verbindungen dienen grundsätzlich zum Aufbau einer zumeist kryptografisch gesicherten und damit privaten Kommunikationsverbindung über ein nicht vertrauenswürdiges Transportmedium (i.d.R. dem Internet). Die kryptografische Absicherung wird im Fall von OpenVPN mit der Transportverschlüsselung Secure Sockets Layer (SSL) bzw. Transport Layer Security (TLS) erreicht. Die Transportverschlüsselung nutzt einen hybriden Verschlüsselungsansatz, wobei der asymmetrische RSA-Algorithmus zum Schlüsselaustausch dient. Die Kommunikationspartner müssen sich gegenseitig authentisieren. Die Integrität der Verbindung wird durch den Einsatz von hashbasierten Message Authentication Codes (HMAC) erreicht. VPN-Lösungen werden heutzutage vielfach in Unternehmensnetzwerken eingesetzt. So können Mitarbeiter mithilfe derartiger VPN-Verbindungen unabhängig vom physischen Standort über das Internet auf das Unternehmensnetzwerk zugreifen. Besteht die VPN-Verbindung, kann der Mitarbeiter wie gewohnt Netzwerkverbindungen zu Diensten im Unternehmensnetzwerk aufbauen. Die OpenVPN-Lösung ist quelloffen und kann damit auch von Unternehmen kostenfrei eingesetzt werden.



(a) (b) (c)

Abbildung 2.1: Status des Kontextmenüs des OpenVPN Symbols

Der OpenVPN Client ist für verschiedene Plattformen (Linux, Max OS X, Windows) verfügbar. In der vorliegenden Dokumentation werden die Ergebnisse der Analyse des windowsbasierten OpenVPN Clients beschrieben. Nach dem Start des OpenVPN Clients erscheint ein Symbol im Benachrichtigungsbereich der Windows Taskleiste. Dies ist nach dem Start die einzige Komponente, die in der grafischen Oberfläche von Windows angezeigt wird. Mit einem Rechtsklick das Symbol im Benachrichtigungsbereich öffnet sich ein Kontextmenü, dem die Funktionalitäten des VPN Clients entnommen werden können. Vor der Konfiguration des Clients können mithilfe des Kontextmenüs lediglich die allgemeinen Einstellungen verändert sowie der Client beendet werden (siehe Abbildung 2.1 (a)). Innerhalb der allgemeinen Einstellungen kann, wie in Abbildung 2.2 dargestellt, die Verbindung über einen Proxyserver und die Sprache konfiguriert werden. Nach der Konfiguration des Clients kann über dieses Kontextmenü die VPN-Verbindung mithilfe der Option `Connect` zum konfigurierten OpenVPN Server aufgebaut werden (siehe Abbildung 2.1 (b)). Die Konfiguration des Clients erfolgt rein dateibasiert mithilfe eines so genannten »Client Package«. Dieses Paket enthält

neben Zertifikaten und Schlüsseln für die asymmetrische Verschlüsselung und Integritätssicherung die Konfigurationsdatei des Clients. Das Client Package wird auf dem OpenVPN Server pro berechtigtem Client erzeugt und muss anschließend manuell über einen nachweislich sicheren Weg auf den entsprechenden Clientrechner übertragen werden. Während des Verbindungsaufbaus wird ein Fenster mit Statusinformationen zum Verbindungsaufbau, wie in Abbildung 2.3 dargestellt, angezeigt. Nach erfolgreichem Aufbau der VPN-Verbindung wird das Fenster mit den Statusinformationen geschlossen und das vormals graue Symbol im Benachrichtigungsbereich der Taskleiste erscheint gelb (siehe Abbildung 2.1 (c)). Eine bestehende VPN-Verbindung kann ebenfalls über das Kontextmenü des Symbols abgebaut werden.

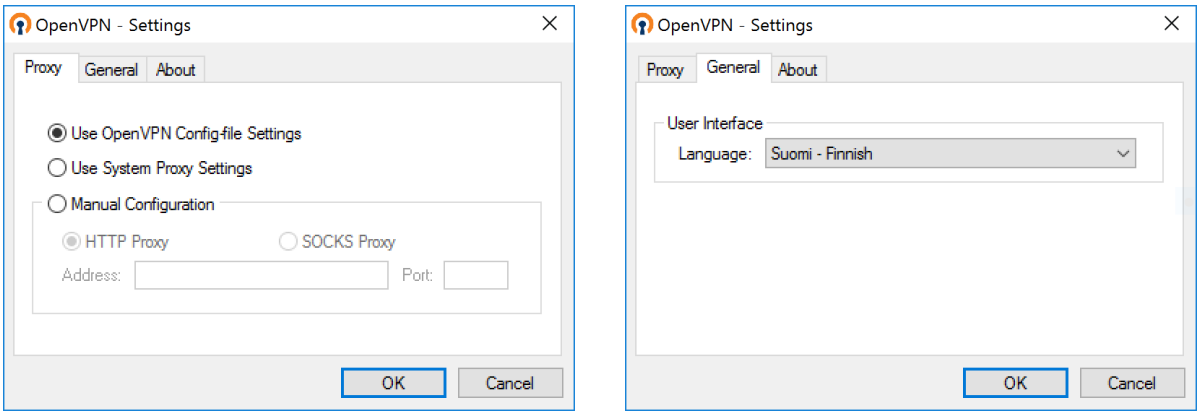


Abbildung 2.2: Konfigurationsdialog des OpenVPN Client

Der OpenVPN Client wurde für die Analyse gewählt, da es sich bei OpenVPN um eine quelloffene und weit verbreitete VPN-Lösung handelt. OpenVPN wird hauptsächlich an Netzgrenzen zwischen internen Netzen und dem Internet eingesetzt und ist in vielen quelloffenen Firewalllösungen wie IPCop[8], pfSense[9] und IPfire[10] integriert. Derartige Firewalllösungen werden nicht nur privat sondern ebenfalls vermehrt in Unternehmensnetzwerken eingesetzt, da sie kostengünstig, quelloffen und zuverlässig sind. Im Zuge dessen wird häufig die Möglichkeit zum Fernzugriff auf das interne (Unternehmens-)Netzwerk durch Aktivierung des OpenVPN Servers ermöglicht. Darüber hinaus bietet OpenVPN mit dem Access Server direkt vorgefertigte kommerzielle VPN-Lösungen, die sich an Unternehmenskunden richten.[11] Aus forensischer Sicht erlangt der OpenVPN Client Relevanz, da im Bereich der Computerkriminalität häufig VPNs von Tätern zu Anonymisierungszwecken eingesetzt werden. So kann ein Täter vor Durchführung der geplanten Tat eine VPN-Verbindung zu einem bestimmten oder auch mehreren hintereinander geschachtelten VPN-Servern aufbauen. Nach dem Aufbau der Verbindung ist der Täter Teil des entfernten Netzwerks und kann beispielsweise einen Proxy-Server innerhalb dieses Netzwerks nutzen, um Zugang zum Internet zu erhalten. Die Zurückverfolgbarkeit des Täters hängt bei der Verwendung von VPN-Verbindungen beinahe ausschließlich von der Kooperationsbereitschaft der Betreiber der VPN-Server oder des Proxy-Servers, den der An-

[8] `http://www.ipcop.org/`, abgerufen am 09.01.2016.
[9] `https://www.pfsense.org/`, abgerufen am 09.01.2016.
[10] `http://www.ipfire.org`, abgerufen am 09.01.2016.
[11] `https://openvpn.net/index.php/access-server/overview.html`, abgerufen am 09.01.2016.

greifer als letzten Hop nutzt, ab. Da der gesamte VPN-Verkehr verschlüsselt übertragen wird, kann der Datenstrom auf dem Transportweg i.d.R. nicht inhaltlich ausgewertet werden. Somit kann es durchaus von Interesse sein, ob ein VPN Client auf einem sichergestellten Rechner zu finden ist und wie dieser Client konfiguriert ist, um ggfs. weitere Sicherstellungen oder Auskunftsersuchen beim Betreiber des OpenVPN-Servers anzustellen.

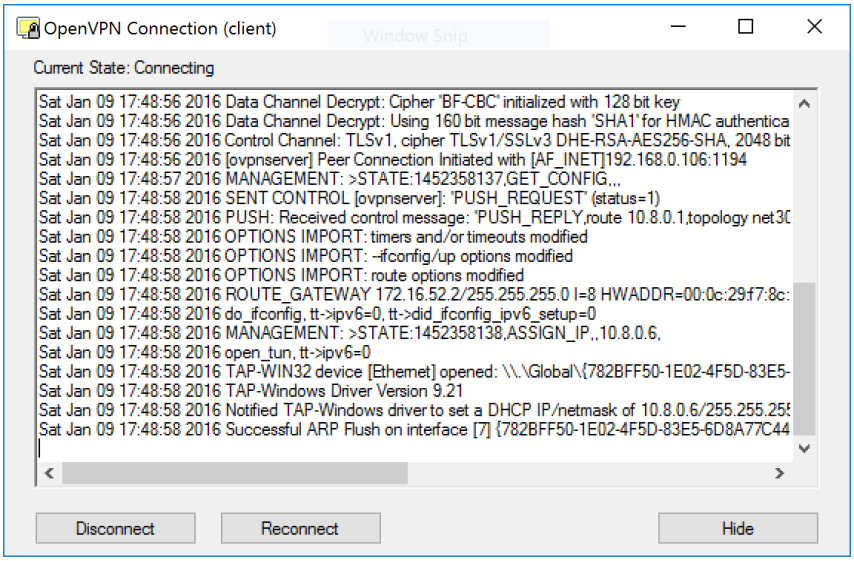


Abbildung 2.3: Stausanzeige während des VPN-Verbindungsaufbaus

3 Technische Anwendungsanalyse

Nachdem die zu untersuchende Anwendung im vorangegangenen Kapitel vorgestellt wurde, folgt in diesem Kapitel die Ergebnisdokumentation der Anwendungsanalyse. Dazu wird zunächst das Vorgehen bei der Analyse eingehend dargestellt, um die Nachvollziehbarkeit der Analyse und damit Einordnung der Ergebnisse sicherzustellen. Im Anschluss folgt ein Abschnitt, in dem die persistenten Spuren im Dateisystem identifiziert und inhaltlich dargestellt werden. Die in der Registry vorhandene Spurenmenge wird anschließend beschrieben. Den Abschluss des Kapitels bildet die Identifikation von Systemdateien, die zur Spurenmenge des OpenVPN Clients gehören. Dabei stehen ausschließlich Prefetch Dateien im Fokus.

3.1 Vorgehen

Zur Analyse der OpenVPN Clientanwendung wird die in Abschnitt 1.3 beschriebene Arbeitsumgebung eingesetzt. Das Vorgehen bei der Analyse ist an die Chronologie eines typischen Arbeitsablaufs (Workflow) mit dem OpenVPN Client angelehnt. Dazu wird der Workflow zunächst in die fünf Phasen

1. Installation des OpenVPN Clients,
2. Start des OpenVPN Clients,
3. Verbindungsaufbau zu einem entfernten OpenVPN Server,
4. Verbindungsabbau und
5. Deinstallation

unterteilt. Diese Phasen werden im Laufe der Anwendungsanalyse chronologisch durchlaufen. Zum Durchlaufen der definierten Phasen müssen Aktionen in der Windows 10 Client VM durchgeführt werden. Zur Nachvollziehbarkeit der Analyse sind diese Aktionen in Tabelle 3.1 den einzelnen Phasen zugeordnet.

Vor jeder der genannten Phase des Workflows wird ein Snapshot zur Sicherung des Status der virtuellen Windows 10 Client Maschine erzeugt. Dadurch wird sichergestellt, dass im weiteren Verlauf der Analyse jede Phase durch Zurückkehren zum Snapshot der VM bei Bedarf nochmals durchlaufen werden kann. Während bzw. nach der Durchführung der in der entsprechenden Phase geforderten Aktionen gem. Tabelle 3.1 werden die erzeugten Spuren akquiriert. So wird zur Akquise mithilfe der Ereignismethode die Software ProcMon vor der Durchführung der entsprechenden Aktionen gestartet. Die Aufzeichnung von Ereignissen startet automatisch mit dem Start der Anwendung. Nach Ausführung der Aktionen wird die Ereignisaufzeichnung mithilfe der Tastenkombination `<Strg>+E` gestoppt und die aufgezeichneten Ereignisse als native PML- und CSV-Datei gespeichert. Zur Akquise der persistenten Spuren in der Windows Registry wird die Zustandsmethode angewendet. Folglich wird vor Durchführung der geforderten Aktionen in der entsprechenden Phase der Zustand der Registry mittels RegShot gesichert (Schaltfläche `1st Shot`). Das derart erstellte Registry-Abbild bildet das

Vorher-Abbild. Nach Durchführung der Aktionen wird ein zweites Abbild (Schaltfläche `2nd Shot`) der Registry – das Nachher-Abbild – angefertigt. Abschließend werden die Abbild durch Betätigen der Schaltfläche `Compare` miteinander verglichen und die Differenz als textbasierte Datei gespeichert.

Lfd. Nr.	Workflow Phase	Aktion(en)
1	Installation des OpenVPN Clients	Installation des OpenVPN Clients mithilfe des Installationsprogramms `openvpn-install-2.3.9-I601-x86_64.exe` unter Verwendung der Standardeinstellungen des Installationsprogramms.
2	Start des OpenVPN Client	Start des OpenVPN Clients durch Aufruf der Verknüpfung `OpenVPN GUI` im Windows Startmenü.
3	Verbindungsaufbau zu einem entfernten OpenVPN Server	Konfiguration des OpenVPN Clients mit einem auf dem OpenVPN Server erzeugten Client Package. Das Client Package wird im Unterverzeichnis `config` im Installationsverzeichnis des OpenVPN Clients abgelegt. Anschließend erfolgt der Verbindungsaufbau zum entfernten OpenVPN Server durch Auswahl der Option `Connect` im Kontextmenü des OpenVPN Symbols in dem Benachrichtigungsbereich der Taskleiste.
4	Verbindungsabbau	Abbau der zuvor aufgebauten Verbindung durch Auswahl der Option `Disconnect` im Kontextmenü des OpenVPN Symbols in dem Benachrichtigungsbereich der Taskleiste.
5	Deinstallation	Deinstallation des OpenVPN Clients mithilfe der Windows Systemsteuerung (Apps & Features).

Tabelle 3.1: Aufstellung der in den Workflow Phasen durchgeführten Aktionen

Zur Verifikation der Spurenmenge wird nach Durchführung der geforderten Aktionen zusätzlich ein Datenträgerabbild mit dem AccessData FTK Imager Lite im Format `E01` erstellt. Da die Gerichtsfestigkeit und forensische Korrektheit des Abbilds bei der Anwendungsanalyse eine untergeordnete Rolle zukommt, wird das Datenträgerabbild der Einfachheit halber im laufenden Betrieb aus der VM heraus erstellt. Das derart erstellte Datenträgerabbild wird im Anschluss in der SIFT Workstation VM mithilfe des log2timeline Werkzeugs prozessiert und durch Anwendung des l2t_process Werkezeugs auf die resultierende Zeitleiste sortiert und gefiltert. Der Befehl zur Erstellung der Zeitleiste als CSV-Datei am Beispiel des Datenträgerabbilds nach der Installation ist der folgenden Auflistung zu entnehmen. Das Abbild ist zuvor in das Dateisystem der SIFT Workstation einzubinden.[12]

```
~# log2timeline -r -z Europe/Berlin /mnt/mount_win/ -w /mnt/hgfs/SIFT/
    Installation_OpenVPN_Client_Diskimage_timeline.csv
```

[12] Siehe `https://digital-forensics.sans.org/blog/2011/12/07/digital-forensic-sifting-super-timeline-analysis-and-creation`, abgerufen am 09.01.2016.

Die Filterung beschränkt die Zeitleiste auf den für die Analyse relevanten Zeitraum und wird mit dem folgenden Befehl umgesetzt.

```
~# l2t_process -b Installation_OpenVPN_Client_Diskimage_timeline.csv
    01-01-2016..01-08-2016 >
    Installation_OpenVPN_Client_Diskimage_timeline_filtered.csv
```

Damit sich die Akquisemethoden nicht gegenseitig beeinflussen und nicht relevante Spuren erzeugen, wird jede Akquise unabhängig voneinander angewendet. Dazu wird vor jeder Akquisemethode der zuvor erstellte Snapshot der Windows 10 Client VM wiederhergestellt. Es ergibt sich der in Abbildung 3.1 dargestellte Ablauf am Beispiel der Installation.

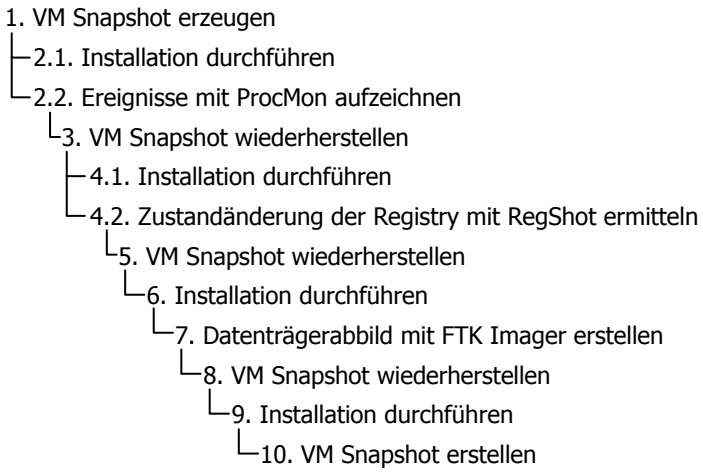


Abbildung 3.1: Schritte der Spurenakquise am Beispiel der Installationsphase

Im Rahmen der Analyse werden die relevanten Spuren anhand geeigneter Ereignisse aus der Menge der mit ProcMon aufgezeichneten Ereignissen extrahiert. Dabei wird der Fokus auf Dateisystemoperation wie das Erstellen, Schreiben und Lesen von Dateien gelegt, um die persistente Spurenmenge zu erhalten. Weiterhin wird die Filterfunktion von ProcMon genutzt, um die Anzeige auf die Ereignisse der relevanten OpenVPN-Prozesse sowie deren Kindprozesse zu beschränken. Zur Einschränkung wird die Prozessbaumansicht von ProcMon (Tastenkürzel `<Strg>+T`) genutzt. Die Validierung der mittels Ereignismethode identifizierten Spuren erfolgt anhand der erzeugten Zeitleiste. Sind die relevanten Dateien ermittelt, folgt eine inhaltliche Analyse der entsprechenden Dateien, sofern diese sinnvoll und aus forensischen Gesichtspunkten relevant ist. Zur Analyse der Registry Veränderungen wird die textbasierte Differenzdatei von RegShot analysiert und die relevanten Veränderungen identifiziert. Zur Validierung der Registry Veränderungen werden die im Rahmen der Ereignismethode mittels ProcMon aufgezeichneten Ereignisse herangezogen.

3.2 Persistente Spurenmenge

Auf Basis des zuvor beschriebenen Vorgehens werden die Spuren der OpenVPN Client Anwendung akquiriert und eingehend analysiert. Die nachfolgenden Abschnitte beschreiben die Analyseergebnisse und stellen folglich die gesamte Spurenmenge eingehend dar. Die Spurenmenge wird in die Bereiche Dateisystem, Registry und Prefetch unterteilt. Die Aufteilung wird vorgenommen, da es sich bei der Registry um eine zentrale Substruktur im Dateisystems des Windows Betriebssystems handelt. Auf Ebene des Dateisystems sind demnach nur Zugriffe auf die Registry Hive Dateien ersichtlich. Die eigentlichen inhaltlichen Änderungen der Registry bleiben auf Dateisystemebene verborgen. Die Ergebnisse der Analyse sind in Form eines zweiseitigen Cheatsheet im Anhang D zusammengefasst und können so praxisorientiert im Rahmen von realen forensischen Analysen genutzt werden.

3.2.1 Dateisystem

Die durch den OpenVPN Client verursachten Dateisystemoperationen und damit die persistente Spurenmenge bezogen auf das Dateisystem werden im Folgenden kategorisiert nach den einzelnen in Abschnitt 3.1 festgelegten Phasen beschrieben. Eine vollständige und kommentierte Gesamtübersicht der Dateisystemoperationen ist in Anhang A zu finden.

Installation

Im Laufe der Installation des OpenVPN Clients wird zunächst das Installationsverzeichnis `C:\Program Files\OpenVPN` erzeugt. Das Verzeichnis enthält alle für den OpenVPN Client relevanten Daten, die in einer festgelegten Subverzeichnisstruktur abgelegt werden. Die erzeugten Dateien und Verzeichnisse sind in Tabelle 3.2 aufgelistet sowie deren Zweck erläutert.

Lfd. Nr.	Datei	Beschreibung
1	`C:\Program Files\OpenVPN`	Installationsverzeichnis des OpenVPN Client
2	`C:\Program Files\OpenVPN\bin`	Subverzeichnis enthält die binären Dateien, die zur Ausführung des OpenVPN Clients benötigt werden
3	`C:\Program Files\OpenVPN\bin\openvpn.exe`	Ausführbare Datei des OpenVPN Clients (Kommandozeilenprogramm)
4	`C:\Program Files\OpenVPN\bin\openvpn-gui.exe`	GUI-Komponente des OpenVPN Clients
5	`C:\Program Files\OpenVPN\bin\openvpnserv.exe`	Ausführbare Datei des OpenVPN Dienstes

Lfd. Nr.	Datei	Beschreibung
6	`C:\Program Files\OpenVPN\bin\libeay32.dll`	Dynamische Bibliothek, die Verschlüsselungsfunktionen wie bspw. RSA bereitstellt (Teil der OpenSSL Bibliothek)[13]
7	`C:\Program Files\OpenVPN\bin\ssleay32.dll`	Dynamische Bibliothek, die SSL-Funktionalitäten bereitstellt (Teil der OpenSSL Bibliothek)[14]
8	`C:\Program Files\OpenVPN\bin\liblzo2-2.dll`	Dynamische Bibliothek, die verlustfreie Kompressionsfunktionalitäten bereitstellt (Lempel-Ziv-Oberhumer Kompressionsalgorithmus[15])
9	`C:\Program Files\OpenVPN\bin\libpkcs11-helper-1.dll`	Dynamische Bibliothek, die Funktionalitäten zur Erstellung und Manipulation von kryptografischen Token bereitstellt (Public-Key Cryptography Standard 11[16])
10	`C:\Program Files\OpenVPN\doc`	Subverzeichnis enthält die Dokumentation des OpenVPN Clients
11	`C:\Program Files\OpenVPN\doc\INSTALL-win32.txt`	Textdatei, die Installationshinweise für das Upgrade von früheren OpenVPN Versionen enthält
12	`C:\Program Files\OpenVPN\doc\openvpn.8.html`	HTML-Datei, die die Manpage der ausführbaren Datei openvpn.exe enthält
13	`C:\Program Files\OpenVPN\doc\license.txt`	Textdatei, die Lizenzinformationen des OpenVPN Client enthält
14	`C:\Program Files\OpenVPN\config`	Subverzeichnis enthält die Konfiguration des OpenVPN Client (Ablageort für das Client Package)
15	`C:\Program Files\OpenVPN\config\README.txt`	Textdatei, die Hinweise zur Ablage des Client Package enthält
16	`C:\Program Files\OpenVPN\sample-config`	Subverzeichnis enthält Beispielkonfigurationsdateien

[13] Informationen stammen aus der Analyse der von der Bibliothek exportierten Funktionen, die mittels DLL Export Viewer (`http://www.nirsoft.net/utils/dll_export_viewer.html`, abgerufen am 10.01.2016) ermittelt wurden.

[14] Informationen stammen aus der Analyse der von der Bibliothek exportierten Funktionen, die mittels DLL Export Viewer ermittelt wurden.

[15] Siehe `https://de.wikipedia.org/wiki/Lempel-Ziv-Oberhumer` für weitere Informationen, abgerufen am 10.01.2016.

[16] Vgl. `http://www.cryptsoft.com/pkcs11doc/STANDARD/pkcs-11v2-11r1.pdf` für detaillierte Informationen, abgerufen am 10.01.2016.

Lfd. Nr.	Datei	Beschreibung
17	`C:\Program Files\OpenVPN\sample-config\sample.ovpn`	Textdatei, die eine beispielhafte OpenVPN Konfiguration enthält (client- oder serverseitig ist nicht bestimmbar)
18	`C:\Program Files\OpenVPN\sample-config\client.ovpn`	Textdatei, die eine beispielhafte Konfiguration der Clientseite enthält
19	`C:\Program Files\OpenVPN\sample-config\server.ovpn`	Textdatei, die eine beispielhafte Konfiguration der Serverseite enthält
20	`C:\Program Files\OpenVPN\log`	Subverzeichnis enthält Log-Dateien, die beim Verbindungsaufbau und -abbau vom OpenVPN Client erzeugt werden
21	`C:\Program Files\OpenVPN\log\README.txt`	Textdatei, die eines Hinweis zum Zweck des `log` Subverzeichnisses enthält
22	`C:\Program Files\OpenVPN\icon.ico`	Grafisches Symbol des OpenVPN Clients
23	`C:\Program Files\OpenVPN\Uninstall.exe`	Ausführbare Datei zur Deinstallation des OpenVPN Clients

Tabelle 3.2: Auflistung der im OpenVPN Installationsverzeichnis abgelegten Dateien

Demnach enthält das Subverzeichnis `bin` jegliche Binärdateien, die zur Ausführung des OpenVPN Clients benötigt werden. Dazu zählen dynamische Bibliotheken und die ausführbaren Dateien für den OpenVPN Client und dessen GUI-Komponente. Das Subverzeichnis `doc` enthält eine kurze Dokumentation des OpenVPN Clients sowie Installationshinweise und Lizenzinformationen. Das `config` Subverzeichnis enthält nach der Installation lediglich einen Hinweis zur Konfiguration des OpenVPN Clients. Im Rahmen der Konfiguration des OpenVPN Clients wird in diesem Verzeichnis das Client Package abgelegt. Textbasierte Beispielkonfigurationsdateien sind im Subverzeichnis `sample-config` abgelegt. Zuletzt wird noch das Subverzeichnis `log` erzeugt, das Log-Dateien, die üblicherweise beim Verbindungsaufbau und -abbau geschrieben werden, enthält. Analog zum `config` Subverzeichnis enthält das `log` Verzeichnis nach der Installation lediglich eine Textdatei, die den Hinweis enthält, dass in diesem Verzeichnis die Logdateien des OpenVPN Clients abgelegt werden. Da die in Tabelle 3.2 aufgelisteten Dateien aus inhaltlicher Sicht keine Relevanz für forensische Untersuchungen haben, werden diese nicht weiter analysiert.

Neben dem Installationsverzeichnis werden Einträge im Startmenü durch den OpenVPN Client Installer erzeugt. Diese Dateien werden in der Tabelle 3.3 aufgelistet und inhaltlich nicht weiter analysiert, da diesen Dateien keine forensische Relevanz zukommt.

Lfd. Nr.	Datei	Beschreibung
1	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN`	OpenVPN Verzeichnis im Windows Start Menü
2	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation`	Subverzeichnis enthält die Start Menü Einträge für die Dokumenation
3	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Manual Page.lnk`	Verweis auf die OpenVPN Handbuch Seite
4	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Windows Notes.lnk`	Verweis auf die Datei `INSTALL-win32.txt`
5	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN HOWTO.url`	Verweis auf die OpenVPN HOWTO Webseite
6	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Web Site.url`	Verweis auf die OpenVPN Webseite
7	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Wiki.url`	Verweis auf das OpenVPN Wiki
8	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Support.url`	Verweis auf die OpenVPN Support Webseite
9	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Utilities`	Subverzeichnis enthält die Start Menü Einträge für Werkzeuge

Lfd. Nr.	Datei	Beschreibung
10	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Utilities\Generate a static OpenVPN key.lnk`	Verweis auf die ausführbare Datei `openvpn.exe`, die mit Parametern zur Erzeugung eines statischen Schlüssels aufgerufen wird
11	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Shortcuts`	Subverzeichnis enthält Verweise auf verschiedene Verzeichnisse im Installationsverzeichnis
12	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Shortcuts\OpenVPN Sample Configuration Files.lnk`	Veweis auf das Verzeichnis `sample-config` im Installationsverzeichnis
13	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Shortcuts\OpenVPN log file directory.lnk`	Verweis auf das Verzeichnis `log` im Installationsverzeichnis
14	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Shortcuts\OpenVPN configuration file directory.lnk`	Verweis auf das Verzeichnis `config` im Installationsverzeichnis
15	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\OpenVPN GUI.lnk`	Veweis auf die ausführbare Datei `openvpn-gui.exe`
16	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Uninstall OpenVPN.lnk`	Verweis auf die Datei `Uninstall.exe`

Tabelle 3.3: Auflistung der im Windows Start Menü während der Installation abgelegten Dateien

Weitere Dateizugriffe abgesehen von Zugriffen auf Standardbibliotheken des Windowsbetriebssystems und temporären Dateien erfolgen seitens des OpenVPN Client Installers nicht.

Im Rahmen der Installation des OpenVPN Clients werden die Kindprozesse `tap-windows.exe` und `tapinstall.exe` zur Installation eines virtuellen Netzwerkadapters – dem so genannten TAP-Adapter[17] – erzeugt (siehe Abbildung 3.2). Die Installation dieses TAP-Adapter erzeugt ebenfalls persistente Spuren auf dem Datenträger. Analog zum OpenVPN Client Installer wird ein Installa-

[17] Siehe `https://de.wikipedia.org/wiki/TUN/TAP` für weitere Informationen, abgerufen am 10.01.2016.

tionsverzeichnis und ein Verzeichnis im Start Menü durch den Prozess `tap-windows.exe` erzeugt. Das Installationsverzeichnis wird unter `C:\Program Files\TAP-Windows` angelegt. Die nachfolgende Tabelle 3.4 enthält die persistenten Daten im Installationsverzeichnis.

Lfd. Nr.	Datei	Beschreibung
1	`C:\Program Files\TAP-Windows`	Installationsverzeichnis von TAP-Windows
2	`C:\Program Files\TAP-Windows\bin`	Subverzeichnis enthält binäre Dateien von TAP-Windows
3	`C:\Program Files\TAP-Windows\bin\tapinstall.exe`	Ausführbare Datei zur Installation und Deinstallation von TAP-Netzwerkschnittstellen
4	`C:\Program Files\TAP-Windows\bin\addtap.bat`	Batch-Skript zur Erzeugung der TAP-Netzwerkschnittstelle `tap0901` mithilfe von `tapinstall.exe`
5	`C:\Program Files\TAP-Windows\bin\deltapall.bat`	Batch-Skript zum Entfernen aller TAP-Netzwerkschnittstellen `tap0901` mithilfe von `tapinstall.exe`
6	`C:\Program Files\TAP-Windows\driver`	Subverzeichnis enthält Treiber für die TAP-Netzwerkschnittstelle
7	`C:\Program Files\TAP-Windows\driver\OemVista.inf`	Textbasierte Konfigurationsdatei für die Installation der TAP-Netzwerkschnittstelle
8	`C:\Program Files\TAP-Windows\driver\tap0901.cat`	Treiberkatalogdatei für die Installation der TAP-Netzwerkschnittstelle
9	`C:\Program Files\TAP-Windows\driver\tap0901.sys`	Treiberdatei für die Installation der TAP-Schnittstelle
10	`C:\Program Files\TAP-Windows\license.txt`	Textbasierte Datei, die Lizenzinformationen enthält
11	`C:\Program Files\TAP-Windows\icon.ico`	Grafisches Symbol von OpenVPN
12	`C:\Program Files\TAP-Windows\Uninstall.exe`	Ausführbare Datei zur Deinstallation von TAP-Windows

Tabelle 3.4: Auflistung der im TAP-Windows Installationsverzeichnis abgelegten Dateien

Zusätzlich zu den Dateien im Installationsverzeichnis werden Start Menü Einträge erzeugt. Diese können der nachfolgenden Tabelle entnommen werden.

Lfd. Nr.	Datei	Beschreibung
1	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows`	Verzeichnis der Start Menü Einträge von TAP-Windows
2	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows\Utilities`	Subverzeichnis enthält Werkzeuge von TAP-Windows
3	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows\Utilities\Add a new TAP virtual ethernet adapter.lnk`	Verweis auf das Batch-Skript `addtap.bat` zum Hinzufügen einer TAP-Schnittstelle
4	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows\Utilities\Delete ALL TAP virtual ethernet adapters.lnk`	Verweis auf das Batch-Skript `deltapall.bat` zum Entfernen aller TAP-Netzwerkschnittstellen

Tabelle 3.5: Auflistung der durch die Installation von TAP-Windows im Start Menü abgelegten Dateien

Vom Prozess `tapinstall.exe` werden im Installationsablauf lediglich temporäre Dateien erzeugt, die nach der Installation entfernt werden und daher nicht Teil der relevanten persistenten Spurenmenge sind. Zur Einrichtung der TAP-Schnittstelle greift der Prozess auf die Treiberdateien (lfd. Nr. 7 bis 9 in Tabelle 3.4) zu und schreibt Informationen nach der erfolgreichen Einrichtung der Schnittstelle in die textbasierte Datei `C:\Windows\INF\setupapi.dev.log`. Die nachfolgende Auflistung zeigt einen Ausschnitt der dort hinterlegten Informationen. Der gesamte relevante Inhalt der Datei ist in Anhang B hinterlegt.

```
>>>  [Device Install (UpdateDriverForPlugAndPlayDevices) - tap0901]
>>>  Section start 2016/01/08 02:57:15.581
      cmd: "C:\Program Files\TAP-Windows\bin\tapinstall.exe" install "C:\Program
           Files\TAP-Windows\driver\OemVista.inf" tap0901
     ndv: INF path: C:\Program Files\TAP-Windows\driver\OemVista.inf
     ndv: Install flags: 0x00000001
     ndv: {Update Device Driver - ROOT\NET\0000}
     ndv:      Search options: 0x00000080
     ndv:      Searching single INF 'C:\Program Files\TAP-Windows\driver\
         OemVista.inf'
     dvi:      {Build Driver List} 02:57:15.597
     dvi:           Searching for hardware ID(s):
```

```
     dvi:                tap0901
     sig:           {_VERIFY_FILE_SIGNATURE} 02:57:15.613
     sig:                Key      = oemvista.inf
     sig:                FilePath = c:\program files\tap-windows\driver\oemvista
         .inf
     sig:                Catalog  = c:\program files\tap-windows\driver\tap0901.
         cat
!    sig:                Verifying file against specific (valid) catalog failed!
     (0x800b0109)
!    sig:                Error 0x800b0109: A certificate chain processed, but
    terminated in a root certificate which is not trusted by the trust provider.
     sig:           {_VERIFY_FILE_SIGNATURE exit(0x800b0109)} 02:57:15.659
[...]
```

Aus den Informationen in der Datei `setupapi.dev.log` ist ersichtlich, dass das Gerät `tap0901` mit dem bekannten Treiber aus dem Installationsverzeichnis von TAP-Windows installiert wurde. Dies ist ein hinreichender Hinweis auf die Installation des OpenVPN Clients, da dieser zwingend eine TAP-Schnittstelle benötigt und diese Schnittstelle immer vom Typ `tap0901` ist. Der Installationszeitpunkt der TAP-Schnittstelle ist der letzten Zeile des Abschnitts zur Geräteinstallation zu entnehmen. Der Zeitstempel ist in der lokal am Rechner konfigurierten Zeitzone hinterlegt.

```
<<<  Section end 2016/01/08 02:57:19.408
<<<  [Exit status: SUCCESS]
```

Da die Installation der TAP-Schnittstelle mit der Installation des OpenVPN Clients einhergeht, entspricht der angegebene Zeitstempel hinreichend genau dem Installationszeitpunkt des OpenVPN Clients. Aus forensischer Sicht interessant ist, dass diese Informationen auch nach der Deinstallation des OpenVPN Clients erhalten bleiben. Die Datei `setupapi.dev.log` kann somit als guter Indikator herangezogen werden, ob jemals ein OpenVPN Client auf einem zu analysierenden Rechner installiert war. Als Vorgriff zur Beschreibung der Deinstallationsphase sei an dieser Stelle angemerkt, dass auch der Entfernungszeitpunkt der TAP-Schnittstelle und damit ein guter Indikator für die Deinstallation des OpenVPN Clients in der Datei `setupapi.dev.log` hinterlegt wird.

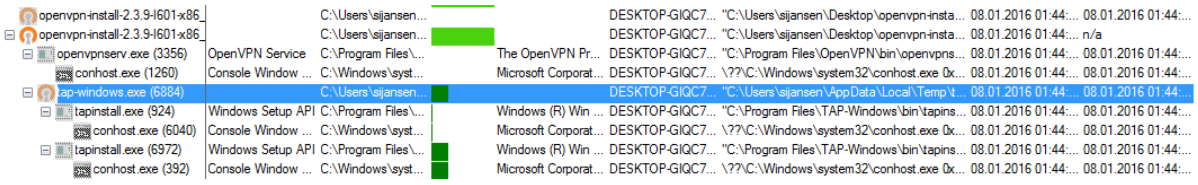


Abbildung 3.2: Prozessbaum der OpenVPN Client Installation

Start des OpenVPN Clients

Durch den Start des OpenVPN Clients entstehen lesende Zugriffe auf die im Subverzeichnis `bin` im Installationsverzeichnis des Clients abgelegten Dateien (siehe lfd. Nr. 3 bis 9 in Tabelle 3.2). Weiterhin prüft der Client die Existenz des Subverzeichnisses `log` und fragt den Inhalt des Subverzeichnisses `config` beim Start ab. Weitere Zugriffe erfolgen auf Standardbibliotheken, die keine Relevanz für die Spurenmenge aufweisen.

Verbindungsaufbau zu einem entfernten OpenVPN Server

Bevor der Verbindungsaufbau zu einem OpenVPN Server stattfinden kann, muss das Client Package manuell im Subverzeichnis `config` persistiert werden. Damit gehört das Client Package zu der Spurenmenge des OpenVPN Clients. Typischerweise besteht das Client Package aus den in Tabelle 3.6 aufgelisteten Dateien, wobei die Dateinamen in realen Szenarien natürlich abweichen können. Die Dateiendungen bleiben jedoch i.d.R. konstant.

Lfd. Nr.	Datei	Beschreibung
1	`C:\Program Files\OpenVPN\config\ca.crt`	Certificate Authority Zertifikat des OpenVPN Servers. Sowohl das Zertifikat des Servers als auch Clients sind zu Authentisierungszwecken mit dem zu diesem Stammzertifikat passenden privaten Schlüssel signiert.
2	`C:\Program Files\OpenVPN\config\client.ovpn`	Textbasierte Konfigurationsdatei für den OpenVPN Client
3	`C:\Program Files\OpenVPN\config\client1.crt`	Zertifikat des Clients
4	`C:\Program Files\OpenVPN\config\client1.key`	Zu dem Clientzertifikat passender privater Schlüssel

Tabelle 3.6: Auflistung der im Client Package enthaltenen Dateien

Es können mehrere Client Packages im `config` Subverzeichnis abgelegt sein. Aus forensischer Sicht interessant sind die Clientkonfigurationsdatei `client.ovpn` und das Clientzertifikat `client1.crt`. Die Clientkonfigurationsdatei enthält den Hostnamen des entfernen OpenVPN Servers sowie den Kommunikationsport und Verweise auf das verwendete Clientzertifikat. Die entsprechenden Konfigurationsdirektiven `remote` und `cert` können der folgenden Auflistung entnommen werden. In diesem Fall lautet der Hostname des entfernten OpenVPN Servers `ovpnserver` und die VPN-Verbindung wird über den UDP-Port 1194 aufgebaut.

```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote ovpnserver 1194
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random
[...]
# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert client1.crt
key client1.key
```

Aus dem Clientzertifikat kann der Distinguished Name (kurz DN) des Clients extrahiert werden. Der DN ist im Subject Feld des Zertifikats hinterlegt (siehe Abbildung 3.3) und kann bei ggfs. nachgelagerten Analysen auf dem OpenVPN Server hilfreich sein. Denn der DN des Clients wird bei jedem Verbindungsaufbau an den Server gesendet und in dessen Logdatei[18] aufgenommen, sodass die Verbindungen von dem entsprechenden Client zum Server eindeutig identifiziert werden können, sofern das Client Package ausschließlich von dem entsprechenden Client verwendet wird.

Während des Verbindungsaufbaus wird die clientseitige Logdatei `client.log` im Subverzeichnis `log` im Installationsverzeichnis des OpenVPN Clients erzeugt. Die Logdatei enthält die während des Verbindungsaufbaus angezeigten Informationen (vgl. Abbildung 2.3) in textueller Form. Anhang C enthält eine beispielhafte Logdatei. Der Logdatei kann unter anderem entnommen werden, wann eine VPN-Verbindung zu welchem OpenVPN Server hergestellt wurde. Die Logdatei wird bei jedem Verbindungsaufbau überschrieben. Folglich kann der clientseitigen Logdatei lediglich Informationen zu der letzten bzw. der aktuell aktiven VPN-Verbindung entnommen werden. Eine historische Zusammenstellung der aufgebauten VPN-Verbindungen existiert clientseitig nicht.

Im Laufe des Verbindungsaufbaus entstehen neben der Erzeugung des Clientlogs lesende Zugriff auf die Dateien im Subverzeichnis `bin` im Installationsverzeichnis des OpenVPN Clients. Zudem wird lesend auf die Dateien des Client Package zugegriffen.

[18] In der Regel werden die Logeinträge vom OpenVPN Server von Syslog verarbeitet und landen in der Datei `/var/log/-syslog`. Zur Filterung der Datei auf relevante Meldungen kann die Zeichenkette `ovpn-server` genutzt werden.

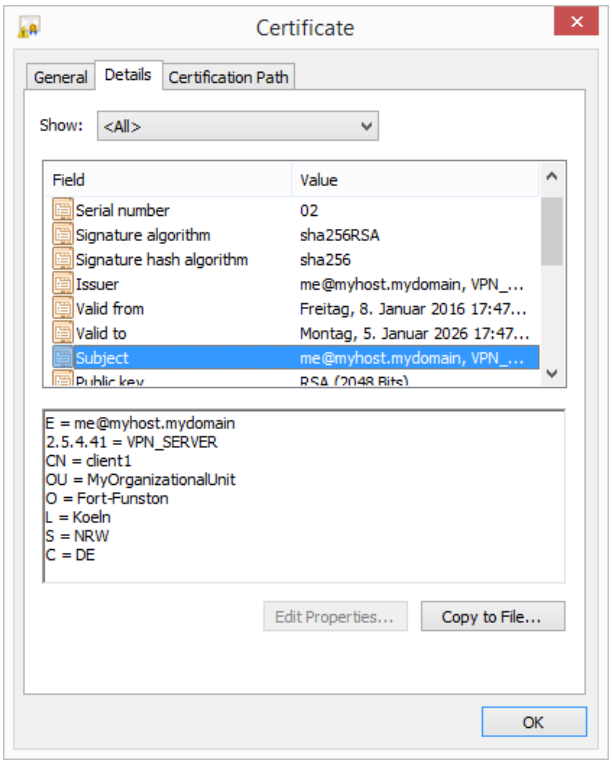


Abbildung 3.3: DN im Clientzertifikat

Verbindungsabbau

Dein einzig forensisch relevante Dateisystemoperation erfolgt im Rahmen des Verbindungsabbaus durch den schreibenden Zugriff auf das Clientlog im Subverzeichnis `log` des Installationsverzeichnisses vom OpenVPN Client. Durch den schreibenden Zugriff wird der Logdatei Informationen zum Verbindungsabbau angehängt. Weitere, aus forensischer Sicht relevante Dateisystemzugriffe, erfolgen im Rahmen des Verbindungsabbaus nicht.

Deinstallation

Während der Deinstallation des OpenVPN Clients werden lediglich temporäre Dateien im Dateisystem erzeugt, die keine weitere Relevanz für die Spurenmenge der Anwendung aufweisen. Da der Client bei der Deinstallation vom System entfernt wird, stehen in dieser Phase jedoch vor allem Löschoperationen auf dem Dateisystem im Vordergrund. Aus forensischer Sicht von besonderem Interesse sind dabei persistente Spuren, die im Rahmen der Installation erzeugt, jedoch bei der Deinstallation nicht entfernt werden. Derartige Spuren helfen bei der Beantwortung der Frage, ob der OpenVPN Client jemals in dem aktuellen Betriebssystem des zu untersuchenden Rechners installiert war.

Bei der Deinstallation des OpenVPN Clients werden die in den Tabellen 3.3, 3.4 und 3.5 genannten Datei ausnahmslos entfernt. Eine Ausnahme bildet das OpenVPN Installationsverzeichnis (vgl. Tabelle 3.2). Hier bleibt nach der Deinstallation das `config` Subverzeichnis inkl. Client Package er-

halten. Vermutlich aus Gründen der Benutzerfreundlichkeit, sodass bei einer erneuten Installation des Clients nicht nochmalig das Client Package persistiert werden muss. Folglich ist das Client Package im Installationsverzeichnis des OpenVPN Clients ein guter Indikator zur Beantwortung der Frage, ob der OpenVPN Client auf einem zu untersuchenden Rechner installiert war. An dieser Stelle sei jedoch angemerkt, dass diese Spur aus Sicht des Anwenders mit geringen Aufwand durch Löschen des Installationsverzeichnisses beseitigt werden kann. Eine robustere Spur bietet da die bereits angedeutete Datei `C:\Windows\INF\setupapi.dev.log`. Im Laufe der Deinstallation wird schreibend auf die Datei zugegriffen und die Deinstallation der TAP-Netzwerkschnittstelle vermerkt. Die nachfolgende Auflistung stellt den entsprechenden Eintrag dar.

```
>>>  [Delete Device - ROOT\NET\0000]
>>>  Section start 2016/01/08 20:19:42.015
       cmd: "C:\Program Files\TAP-Windows\bin\tapinstall.exe" remove tap0901
      dvi: Query-and-Remove succeeded
<<<  Section end 2016/01/08 20:19:42.109
<<<  [Exit status: SUCCESS]
```

Es ist ersichtlich, dass anhand der Datei `setupapi.dev.log` sowohl der Installationszeitpunkt als auch der Deinstallationszeitpunkt des OpenVPN Clients entnommen werden kann. Da aus Anwendersicht während der Installation kein Indiz aufkommt, dass Informationen zur Installation des OpenVPN Clients in der Datei `setupapi.dev.log` hinterlegt werden, ist es unwahrscheinlich, dass ein Anwender diese Spur entfernt. Weiterhin ist der Aufwand für den Anwender recht hoch, da dieser zunächst den OpenVPN Client eingehend analysieren muss, bevor er erkennt, dass in der genannten Datei eine persistente Spur existiert.

3.2.2 Windows Registry

Analog zum Aufbau des vorangegangen Abschnitts zu Spuren im Dateisystem werden in diesem Abschnitt die persistenten Spuren in der Windows Registry, kategorisiert nach den bekannten Phasen, beschrieben. Da die Spuren in der Registry weitaus umfangreicher ausfallen, werden diese nicht oder nur auszugsweise im Dokument aufgelistet. Eine detaillierte Aufstellung der relevanten Registryänderungen sind nach den Phasen geordnet im zur Abgabe gehörenden Verzeichnis `RegShot_Results` hinterlegt. Erfolgt in den folgenden Abschnitten ein Veweis auf eine Datei, so ist diese im genannten Verzeichnis zu finden.

Installation

Während der Installation werden umfangreiche Änderungen an der Registry vorgenommen. Aufgrund des Umfangs werden die Registryänderungen im folgenden lediglich grob und stark abstrahiert in einzelne Schritte eingeteilt. Details zu den Änderungen in Form einer Aufstellung der veränderten Registry Keys können der Datei `Reg-Changes_Installation_OpenVPN_GUI_Client_Relevant.txt` entnommen werden. Diese Änderungen umfassen grob eingeteilt die Schritte

- Registrierung der Dateiendung `ovpn`,
- Ablage von Informationen für die Deinstallation,
- Einrichtung der TAP-Netzwerkschnittstelle `tap0901`,
- Registrierung des OpenVPN Dienstes in der Windows Dienstverwaltung und
- Modifikation der Umgebungsvariable `PATH` durch Hinzufügen des Pfads `C:\Program Files\OpenVPN\bin`.

Start des OpenVPN Clients

Beim Start des OpenVPN Clients werden Konfigurationsparameter der GUI-Komponten im Registry Key `HKLM\SOFTWARE\OpenVPN-GUI` als Registry Werte hinzugefügt. Die nachfolgende Tabelle 3.7 stellt die hinzugefügten Registry Werte inkl. Beschreibung zusammen. Die Werte können zusätzlich der Datei `Reg-Changes_First_Start_OpenVPN_Client_Relevant.txt` entnommen werden.

Lfd. Nr.	Registry Wert	Beschreibung
1	`HKLM\SOFTWARE\OpenVPN-GUI\ config_dir: "C:\Program Files\ OpenVPN\config"`	Verzeichnis für die Client Packages
2	`HKLM\SOFTWARE\OpenVPN-GUI\ config_ext: övpn"`	Dateierweiterung der clientseitigen Konfigurationsdatei
3	`HKLM\SOFTWARE\OpenVPN-GUI\ exe_path: "C:\Program Files\ OpenVPN\bin\openvpn.exe"`	Pfad zur ausführbaren Datei `openvpn.exe`
4	`HKLM\SOFTWARE\OpenVPN-GUI\ log_dir: "C:\Program Files\ OpenVPN\log"`	Verzeichnis der Log-Dateien
5	`HKLM\SOFTWARE\OpenVPN-GUI\ log_append: "0"`	Fortschreibung der Logdatei
6	`HKLM\SOFTWARE\OpenVPN-GUI\priority: "NOR-MAL_PRIORITY_CLASS"`	Prozesspriorität
7	`HKLM\SOFTWARE\OpenVPN-GUI\ log_viewer: "C:\Windows\ notepad.exe"`	Programm zum Öffnen der Logdatei
8	`HKLM\SOFTWARE\OpenVPN-GUI\ editor: "C:\Windows\notepad.exe"`	Editoranwendung zur Konfiguration

Lfd. Nr.	Registry Wert	Beschreibung
9	HKLM\SOFTWARE\OpenVPN-GUI\ allow_edit: "1"	Anzeige der Option zur Editierung der Konfigurationsdatei im Kontextmenü des OpenVPN Symbols im Benachrichtungsbereich
10	HKLM\SOFTWARE\OpenVPN-GUI\ allow_service: "0"	Anzeige des Dienstkontrollmenüs im Kontextmenü des OpenVPN Symbols im Benachrichtungsbereich
11	HKLM\SOFTWARE\OpenVPN-GUI\ allow_password: "1"	Anzeige der Option zur Passwortänderung im Kontextmenü des OpenVPN Symbols im Benachrichtungsbereich
12	HKLM\SOFTWARE\OpenVPN-GUI\ allow_proxy: "1"	Anzeige der Option zur Proxykonfiguration im Kontextmenü des OpenVPN Symbols im Benachrichtungsbereich
13	HKLM\SOFTWARE\OpenVPN-GUI\ service_only: "0"	Aktivierung des Service Only Modus
14	HKLM\SOFTWARE\OpenVPN-GUI\ show_balloon: "1"	Anzeige des Balloon Tips im Benachrichtigungsbereich beim erfolgreichen Aufbau der VPN-Verbindung
15	HKLM\SOFTWARE\OpenVPN-GUI\ silent_connection: "0"	Unterdrückung der Statusanzeige beim Verbindungsaufbau
16	HKLM\SOFTWARE\OpenVPN-GUI\ show_script_window: "1"	Unterdrückung der Anzeige des Scriptfensters
17	HKLM\SOFTWARE\OpenVPN-GUI\ passphrase_attempts: "3"	Max. Anzahl an Versuchen zur Eingabe der Passphrase beim Verbindungsaufbau
18	HKLM\SOFTWARE\OpenVPN-GUI\ connectscript_timeout: "15"	Timeout für die Ausführung eines Connect-Skripts
19	HKLM\SOFTWARE\OpenVPN-GUI\ disconnectscript_timeout: "10"	Timeout für die Ausführung eines Disconnect-Skripts
20	HKLM\SOFTWARE\OpenVPN-GUI\ preconnectscript_timeout: "10"	Timeout für die Ausführung eines Preconnect-Skripts

Tabelle 3.7: Beim Start des OpenVPN Clients erzeugte Registry Werte

Die in der Tabelle dargestellten Registry Werte bleiben auch nach der Deinstallation des OpenVPN Clients erhalten und können somit gut als Spuren herangezogen werden.

Verbindungsaufbau zu einem entfernten OpenVPN Server

Im Laufe des Verindungsaufbaus werden hauptsächlich TCP/IP Informationen, die durch das DHCP-Protokoll vom OpenVPN Server bezogen wurden, in der Registry hinterlegt. Dazu werden Regsitry Werte erzeugt. Die nachfolgende Auflistung enthält einen Ausschnitt der relevanten Änderungen in der Registry. Die gesamten Änderungen sind der Datei `Reg-Changes_First_VPN_Connection_Relevant.txt` zu entnehmen.

```
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\DhcpIPAddress: "10.8.0.6"
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\DhcpSubnetMask: "255.255.255.252"
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\DhcpServer: "10.8.0.5"
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\Lease: 0x01E13380
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\LeaseObtainedTime: 0x568FF390
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\T1: 0x57808D50
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\T2: 0x583500A0
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\LeaseTerminatesTime: 0x58712710
HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\{782bff50-1e02-4
    f5d-83e5-6d8a77c44d09}\AddressType: 0x00000000
```

Die in dieser Phase erzeugten Registry Werte bleiben sowohl nach dem Verbindungsabbau als auch nach der Deinstallation des OpenVPN Clients erhalten. In der Auflistung ist jedoch ersichtlich, dass es sich bei dem Identifikationsmerkmal der Schnittstelle um eine UUID gem. RFC 4122[19] handelt, die durchaus variieren kann.

Verbindungsabbau

Beim Verbindungsabbau werden Änderungen an Registry Werten vorgenommen, die im Rahmen der Anwendungsanalyse nicht eindeutig identifiziert werden konnten. Die Änderungen lassen auf gewisse Konfigurationen der TAP-Netzwerkschnittstelle schließen, können abschließend jedoch nicht verifiziert werden. Die Änderungen sind in der Datei `Reg-Changes_Disconnect_VPN_Relevant.txt` zusammengefasst.

[19] Siehe `http://www.ietf.org/rfc/rfc4122.txt`, abgerufen am 11.01.2016

Deinstallation

Ähnlich zur Installation werden in der Deinstallationsphase ebenfalls eine Vielzahl von Änderungen an der Registry vorgenommen. Die detaillierte Auflistung von Änderungen ist der Datei `RegChanges_Deinstallation_OpenVPN_Client_Relevant.txt` zu entnehmen. Die Änderungen sind grundsätzlich komplementär zu den Änderungen in der Installationsphase zu sehen und gliedern sich grob in die Schritte

1. Deregistrierung der Dateiendung `ovpn`,
2. Entfernung der Deinstallationsinformationen,
3. Entfernung der TAP-Netzwerkschnittstelle `tap0901`,
4. Deregistrierung des OpenVPN Dienstes in der Windows Dienstverwaltung und
5. Entfernung des Pfads `C:\Program Files\OpenVPN\bin` von der Umgebungsvariable `PATH`.

Spuren, die nach der Deinstallation in der Registry verbleiben wurden bereits in den vorangegangen Abschnitten beschrieben.

3.2.3 Prefetch

Beim Durchlaufen der verschiedenen Workflow Phasen werden aufgrund des Aufrufs von ausführbaren Dateien eine Vielzahl von Prefetch Dateien erzeugt und im Verzeichnis `C:\Windows\Prefetch` abgelegt. Bei Prefetch Dateien handelt es sich grundsätzlich um Systemdateien, die von Windows zur Startoptimierung von Anwendungen genutzt werden. Prefetch Dateien sind daher ein guter Indikator für die (ehemalige) Existenz von ausführbaren Dateien auf einem System. Prefetch Dateien können inhaltlich mithilfe von bereits existierenden Werkzeugen[20] analysiert und unter anderem Zeitstempel zur letzten Ausführung der entsprechenden ausführbaren Dateien extrahiert werden. Grundsätzlich sind Prefetch Dateien als robuste Spuren einzuordnen, da Endanwender meist nicht über die nötige Kenntnis der Existenz von Prefetch Dateien verfügen. Weitergehend muss der Anwender die Anwendung eingehend analysiert haben, um alle erzeugten Prefetch Spuren zu beseitigen. Die nachfolgende Tabelle 3.8 enthält die phasenübergreifend erzeugten Prefetch Dateien.

Lfd. Nr.	Datei	Beschreibung
1	`C:\Windows\Prefetch\OPENVPN-INSTALL-2.3.9-I601-X8-14A92838.pf`	Wird durch den Aufruf des OpenVPN Client Installers erzeugt
2	`C:\Windows\Prefetch\OPENVPNSERV.EXE-F84A4D0B.pf`	Wird im Rahmen der Installation des OpenVPN Clients erzeugt

[20] Siehe u.a. `https://bitbucket.org/cybertools/prefetch-parser/src` und `https://github.com/PoorBillionaire/Windows-Prefetch-Parser`, beide abgerufen am 11.01.2016.

Lfd. Nr.	Datei	Beschreibung
3	C:\Windows\Prefetch\TAP-WINDOWS.EXE-14AB1A49.pf	Wird im Rahmen der Installation des TAP-Netzwerkadapters erzeugt
4	C:\Windows\Prefetch\TAPINSTALL.EXE-A37B18C4.pf	Wird im Rahmen der Installation des TAP-Netzwerkadapters erzeugt
5	C:\Windows\Prefetch\DRVINST.EXE-4CB4314A.pf	Wird im Rahmen der Installation des TAP-Netzwerkadapters erzeugt
6	C:\Windows\Prefetch\OPENVPN-GUI.EXE-52CF0A3D.pf	Wird durch den Start des OpenVPN Clients erzeugt
7	C:\Windows\Prefetch\OPENVPN.EXE-1E8C6F93.pf	Wird durch den Start des OpenVPN Clients erzeugt
8	C:\Windows\Prefetch\UNINSTALL.EXE-F3C4FC36.pf	Wird bei der Deinstallation des OpenVPN Clients erzeugt
9	C:\Windows\Prefetch\AU_.EXE-0723E3F4.pf	Wird bei der Deinstallation des OpenVPN Clients erzeugt
10	C:\Windows\Prefetch\BU_.EXE-88237329.pf	Wird bei der Deinstallation des OpenVPN Clients erzeugt

Tabelle 3.8: Phasenübergreifend erzeugte Prefetch Dateien

Bei den Prefetch Dateien ist zu beachten, dass der Teil nach dem Bindestrich im Dateinamen variabel ist.

4 Fazit

Die im vorliegenden Dokument beschriebenen Ergebnisse der Anwendungsanalyse haben gezeigt, dass sich die persistente Spurenmenge des OpenVPN Clients aus Dateisystemspuren, Registryspuren und Prefetch Spuren zusammensetzt. Ein aus forensischer Sicht interessanter historischer Verlauf von aufgebauten VPN-Verbindungen kann bei Verwendung des OpenVPN Clients mit der Standardkonfiguration nicht ermittelt werden, da die clientseitige Logdatei bei jedem Verbindungsaufbau überschrieben wird. Dennoch können mit dem Clientlog Aussagen über die zuletzt aufgebaute bzw. zum Zeitpunkt der Sicherstellung aufgebaute VPN-Verbindung getroffen werden. Darüber hinaus konnten im Rahmen der Analyse robuste Spuren, die die (ehemalige) Existenz des OpenVPN Clients auf einem System beweisen, identifiziert werden.

Aus Dateisystemsicht ist das Installationsverzeichnis des OpenVPN Clients `C:\Program Files\OpenVPN` von Interesse, da hier sowohl die clientseitige Logdatei als auch das Client Package, welches zur Konfiguration des OpenVPN Clients dient, gefunden werden kann. Mithilfe dieser Dateien können Aussagen zum entfernten OpenVPN Server und zum Aufbau von VPN-Verbindungen getroffen werden. Muss die (ehemalige) Existenz eines OpenVPN Clients auf einem System bewiesen werden, so hat sich die Datei `C:\Windows\INF\setupapi.dev.log` als robuste Spur erwiesen, die Zeitstempel der Installation und Deinstallation des OpenVPN Clients enthält.

Im Laufe der Analyse der Veränderungen der Windows Registry hat sich der Registry Key `HKLM\SOFTWARE\OpenVPN-GUI` als robuste persistente Spur herausgestellt, da dieser Key auch nach der Deinstallation in der Registry erhalten bleibt. In dem Key sind Konfigurationsparameter der OpenVPN GUI-Komponente enthalten. Darüber hinaus bleibt die Konfiguration des TAP-Netzwerkadapters nach der Deinstallation erhalten. Zu beachten ist bei dieser Spur jedoch, dass der Key aus einer UUID besteht, die nicht konstant ist.

Im Bereich der Spuren in Systemdateien können Prefetch Dateien einen guten Beitrag zur Beantwortung der Frage, ob jemals ein OpenVPN Client auf einem System installiert war, leisten. Durch die inhaltliche Analyse der Prefetch Dateien kann zudem auf Basis der enthaltenen Zeitstempel eine zeitliche Einordnung der (De-)Installation geschehen.

Anhang A

Persistente Spurenmenge des OpenVPN Clients im Dateisystem

Lfd. Nr.	Datei	Beschreibung
1	`C:\Program Files\OpenVPN`	Installationsverzeichnis des OpenVPN Client
2	`C:\Program Files\OpenVPN\bin`	Subverzeichnis enthält die binären Dateien, die zur Ausführung des OpenVPN Clients benötigt werden
3	`C:\Program Files\OpenVPN\bin\openvpn.exe`	Ausführbare Datei des OpenVPN Clients (Kommandozeilenprogramm)
4	`C:\Program Files\OpenVPN\bin\openvpn-gui.exe`	GUI-Komponente des OpenVPN Clients
5	`C:\Program Files\OpenVPN\bin\openvpnserv.exe`	Ausführbare Datei des OpenVPN Servers
6	`C:\Program Files\OpenVPN\bin\libeay32.dll`	Dynamische Bibliothek, die Verschlüsselungsfunktionen wie bspw. RSA bereitstellt (Teil der OpenSSL Bibliothek)
7	`C:\Program Files\OpenVPN\bin\ssleay32.dll`	Dynamische Bibliothek, die SSL-Funktionalitäten bereitstellt (Teil der OpenSSL Bibliothek)
8	`C:\Program Files\OpenVPN\bin\liblzo2-2.dll`	Dynamische Bibliothek, die verlustfreie Kompressionsfunktionalitäten bereitstellt (Lempel-Ziv-Oberhumer Kompressionsalgorithmus)
9	`C:\Program Files\OpenVPN\bin\libpkcs11-helper-1.dll`	Dynamische Bibliothek, die Funktionalitäten zur Erstellung und Manipulation von kryptografischen Token bereitstellt (Public-Key Cryptography Standard 11)
10	`C:\Program Files\OpenVPN\doc`	Subverzeichnis enthält die Dokumentation des OpenVPN Clients
11	`C:\Program Files\OpenVPN\doc\INSTALL-win32.txt`	Textdatei, die Installationshinweise für das Upgrade von früheren OpenVPN Versionen enthält
12	`C:\Program Files\OpenVPN\doc\openvpn.8.html`	HTML-Datei, die die Manpage der ausführbaren Datei openvpn.exe enthält
13	`C:\Program Files\OpenVPN\doc\license.txt`	Textdatei, die Lizenzinformationen des OpenVPN Client enthält

Lfd. Nr.	Datei	Beschreibung
14	`C:\Program Files\OpenVPN\config`	Subverzeichnis enthält die Konfiguration des OpenVPN Client (Ablageort für das Client Package). Das Verzeichnis bleibt nach der Deinstallation erhalten.
15	`C:\Program Files\OpenVPN\config\README.txt`	Textdatei, die Hinweise zur Ablage des Client Package enthält
16	`C:\Program Files\OpenVPN\config\ca.crt`	Certificate Authority Zertifikat des OpenVPN Servers. Sowohl das Zertifikat des Servers als auch Clients sind zu Authentisierungszwecken mit dem zu diesem Stammzertifikat passenden privaten Schlüssel signiert. Der Dateiname kann in realen Szenarien variieren. Die Datei bleibt nach der Deinstallation erhalten.
17	`C:\Program Files\OpenVPN\config\client.ovpn`	Textbasierte Konfigurationsdatei für den OpenVPN Client. Der Dateiname kann in realen Szenarien variieren. Die Datei bleibt nach der Deinstallation erhalten.
18	`C:\Program Files\OpenVPN\config\client1.crt`	Zertifikat des Clients. Der Dateiname kann in realen Szenarien variieren. Die Datei bleibt nach der Deinstallation erhalten.
19	`C:\Program Files\OpenVPN\config\client1.key`	Zu dem Clientzertifikat passender privater Schlüssel. Der Dateiname kann in realen Szenarien variieren. Die Datei bleibt nach der Deinstallation erhalten.
20	`C:\Program Files\OpenVPN\sample-config`	Subverzeichnis enthält Beispielkonfigurationsdateien
21	`C:\Program Files\OpenVPN\sample-config\sample.ovpn`	Textdatei, die eine beispielhafte OpenVPN Konfiguration enthält (Client- oder serverseitig ist nicht bestimmbar)
22	`C:\Program Files\OpenVPN\sample-config\client.ovpn`	Textdatei, die eine beispielhafte Konfiguration der Clientseite enthält
23	`C:\Program Files\OpenVPN\sample-config\server.ovpn`	Textdatei, die eine beispielhafte Konfiguration der Serverseite enthält

Lfd. Nr.	Datei	Beschreibung
24	`C:\Program Files\OpenVPN\log`	Subverzeichnis enthält Log-Dateien, die beim Verbindungsaufbau und -abbau vom OpenVPN Client erzeugt werden
25	`C:\Program Files\OpenVPN\log\README.txt`	Textdatei, die eines Hinweis zum Zweck des `log` Subverzeichnisses enthält
26	`C:\Program Files\OpenVPN\log\client.log`	Textdatei, die das clientseitige Log des OpenVPN Clients enthält
27	`C:\Program Files\OpenVPN\icon.ico`	Grafisches Symbol des OpenVPN Clients
28	`C:\Program Files\OpenVPN\Uninstall.exe`	Ausführbare Datei zur Deinstallation des OpenVPN Clients
29	`C:\Program Files\TAP-Windows`	Installationsverzeichnis von TAP-Windows
30	`C:\Program Files\TAP-Windows\bin`	Subverzeichnis enthält binäre Dateien von TAP-Windows
31	`C:\Program Files\TAP-Windows\bin\tapinstall.exe`	Ausführbare Datei zur Installation und Deinstallation von TAP-Netzwerkschnittstellen
32	`C:\Program Files\TAP-Windows\bin\addtap.bat`	Batch-Skript zur Erzeugung der TAP-Netzwerkschnittstelle `tap0901` mithilfe von `tapinstall.exe`
33	`C:\Program Files\TAP-Windows\bin\deltapall.bat`	Batch-Skript zum Entfernen aller TAP-Netzwerkschnittstellen `tap0901` mithilfe von `tapinstall.exe`
34	`C:\Program Files\TAP-Windows\driver`	Subverzeichnis enthält Treiber für die TAP-Netzwerkschnittstelle
35	`C:\Program Files\TAP-Windows\driver\OemVista.inf`	Textbasierte Konfigurationsdatei für die Installation der TAP-Netzwerkschnittstelle
36	`C:\Program Files\TAP-Windows\driver\tap0901.cat`	Treiberkatalogdatei für die Installation der TAP-Netzwerkschnittstelle
37	`C:\Program Files\TAP-Windows\driver\tap0901.sys`	Treiberdatei für die Installation der TAP-Schnittstelle
38	`C:\Program Files\TAP-Windows\license.txt`	Textbasierte Datei, die Lizenzinformationen enthält
39	`C:\Program Files\TAP-Windows\icon.ico`	Grafisches Symbol von OpenVPN

Lfd. Nr.	Datei	Beschreibung
40	`C:\Program Files\TAP-Windows\Uninstall.exe`	Ausführbare Datei zur Deinstallation von TAP-Windows
41	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN`	OpenVPN Verzeichnis im Windows Start Menü
42	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation`	Subverzeichnis enthält die Start Menü Einträge für die Dokumenation
43	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Manual Page.lnk`	Verweis auf die OpenVPN Handbuch Seite
44	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Windows Notes.lnk`	Verweis auf die Datei `INSTALL-win32.txt`
45	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN HOWTO.url`	Verweis auf die OpenVPN HOWTO Webseite
46	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Web Site.url`	Verweis auf die OpenVPN Webseite
47	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Wiki.url`	Verweis auf das OpenVPN Wiki
48	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Documentation\OpenVPN Support.url`	Verweis auf die OpenVPN Support Webseite
49	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Utilities`	Subverzeichnis enthält die Start Menü Einträge für Werkzeuge

Lfd. Nr.	Datei	Beschreibung
50	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Utilities\Generate a static OpenVPN key.lnk	Verweis auf die ausführbare Datei openvpn.exe, die mit Parametern zur Erzeugung eines statischen Schlüssels aufgerufen wird
51	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Shortcuts	Subverzeichnis enthält Verweise auf verschiedene Verzeichnisse im Installationsverzeichnis
52	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Shortcuts\OpenVPN Sample Configuration Files.lnk	Veweis auf das Verzeichnis sample-config im Installationsverzeichnis
53	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Shortcuts\OpenVPN log file directory.lnk	Verweis auf das Verzeichnis log im Installationsverzeichnis
54	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Shortcuts\OpenVPN configuration file directory.lnk	Verweis auf das Verzeichnis config im Installationsverzeichnis
55	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\OpenVPN GUI.lnk	Veweis auf die ausführbare Datei openvpn-gui.exe
56	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN\Uninstall OpenVPN.lnk	Verweis auf die Datei Uninstall.exe
57	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows	Verzeichnis der Start Menü Einträge von TAP-Windows
58	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows\Utilities	Subverzeichnis enthält Werkzeuge von TAP-Windows
59	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows\Utilities\Add a new TAP virtual ethernet adapter.lnk	Verweis auf das Batch-Skript addtap.bat zum Hinzufügen einer TAP-Schnittstelle

Lfd. Nr.	Datei	Beschreibung
60	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows\Utilities\Delete ALL TAP virtual ethernet adapters.lnk`	Verweis auf das Batch-Skript `deltapall.bat` zum Entfernen aller TAP-Netzwerkschnittstellen
61	`C:\Windows\INF\setupapi.dev.log`	Enthält Informationen zu Geräteinstallationen. Für den OpenVPN Client relevant ist die Installation der TAP-Schnittstelle `tap0901`. Die Spur bleibt nach der Deinstallation des Clients erhalten.
62	`C:\Windows\Prefetch\OPENVPN-INSTALL-2.3.9-I601-X8-14A92838.pf`	Wird durch den Aufruf des OpenVPN Client Installers erzeugt
63	`C:\Windows\Prefetch\OPENVPNSERV.EXE-F84A4D0B.pf`	Wird im Rahmen der Installation des OpenVPN Clients erzeugt
64	`C:\Windows\Prefetch\TAP-WINDOWS.EXE-14AB1A49.pf`	Wird im Rahmen der Installation des TAP-Netzwerkadapters erzeugt
65	`C:\Windows\Prefetch\TAPINSTALL.EXE-A37B18C4.pf`	Wird im Rahmen der Installation des TAP-Netzwerkadapters erzeugt
66	`C:\Windows\Prefetch\DRVINST.EXE-4CB4314A.pf`	Wird im Rahmen der Installation des TAP-Netzwerkadapters erzeugt
67	`C:\Windows\Prefetch\OPENVPN-GUI.EXE-52CF0A3D.pf`	Wird durch den Start des OpenVPN Clients erzeugt
68	`C:\Windows\Prefetch\OPENVPN.EXE-1E8C6F93.pf`	Wird durch den Start des OpenVPN Clients erzeugt
69	`C:\Windows\Prefetch\UNINSTALL.EXE-F3C4FC36.pf`	Wird bei der Deinstallation des OpenVPN Clients erzeugt
70	`C:\Windows\Prefetch\AU_.EXE-0723E3F4.pf`	Wird bei der Deinstallation des OpenVPN Clients erzeugt
71	`C:\Windows\Prefetch\BU_.EXE-88237329.pf`	Wird bei der Deinstallation des OpenVPN Clients erzeugt

Tabelle A.1: Auflistung der durch den OpenVPN Client erzeugten persisteten Dateisystemspuren

Anhang B

Relevanter Inhalt der Datei `C:\Windows\INF\setupapi.dev.log`

```
[...]
>>>  [Device Install (UpdateDriverForPlugAndPlayDevices) - tap0901]
>>>  Section start 2016/01/08 02:57:15.581
      cmd: "C:\Program Files\TAP-Windows\bin\tapinstall.exe" install "C:\Program
           Files\TAP-Windows\driver\OemVista.inf" tap0901
     ndv: INF path: C:\Program Files\TAP-Windows\driver\OemVista.inf
     ndv: Install flags: 0x00000001
     ndv: {Update Device Driver - ROOT\NET\0000}
     ndv:      Search options: 0x00000080
     ndv:      Searching single INF 'C:\Program Files\TAP-Windows\driver\
         OemVista.inf'
     dvi:      {Build Driver List} 02:57:15.597
     dvi:           Searching for hardware ID(s):
     dvi:                tap0901
     sig:           {_VERIFY_FILE_SIGNATURE} 02:57:15.613
     sig:                Key      = oemvista.inf
     sig:                FilePath = c:\program files\tap-windows\driver\oemvista
         .inf
     sig:                Catalog  = c:\program files\tap-windows\driver\tap0901.
         cat
!    sig:                Verifying file against specific (valid) catalog failed!
     (0x800b0109)
!    sig:                Error 0x800b0109: A certificate chain processed, but
    terminated in a root certificate which is not trusted by the trust provider.
     sig:           {_VERIFY_FILE_SIGNATURE exit(0x800b0109)} 02:57:15.659
     sig:           {_VERIFY_FILE_SIGNATURE} 02:57:15.675
     sig:                Key      = oemvista.inf
     sig:                FilePath = c:\program files\tap-windows\driver\oemvista
         .inf
     sig:                Catalog  = c:\program files\tap-windows\driver\tap0901.
         cat
     sig:                Success: File is signed in Authenticode(tm) catalog.
     sig:                Error 0xe0000242: The publisher of an Authenticode(tm)
         signed catalog has not yet been established as trusted.
     sig:           {_VERIFY_FILE_SIGNATURE exit(0xe0000242)} 02:57:15.691
     dvi:           Created Driver Node:
     dvi:                HardwareID   - tap0901
     dvi:                InfName      - c:\program files\tap-windows\driver\
         oemvista.inf
     dvi:                DevDesc      - TAP-Windows Adapter V9
```

```
dvi:                Section      - tap0901.ndi
dvi:                Rank         - 0x00ff0000
dvi:                Signer Score - Authenticode
dvi:                DrvDate      - 11/05/2014
dvi:                Version      - 9.0.0.21
dvi:      {Build Driver List - exit(0x00000000)} 02:57:15.722
dvi:      {DIF_SELECTBESTCOMPATDRV} 02:57:15.722
dvi:           Default installer: Enter 02:57:15.722
dvi:                {Select Best Driver}
dvi:                     Class GUID of device changed to: {4d36e972-e325-11
    ce-bfc1-08002be10318}.
dvi:                     Selected:
dvi:                          Description - [TAP-Windows Adapter V9]
dvi:                          InfFile     - [c:\program files\tap-windows\
    driver\oemvista.inf]
dvi:                          Section     - [tap0901.ndi]
dvi:                {Select Best Driver - exit(0x00000000)}
dvi:           Default installer: Exit
dvi:      {DIF_SELECTBESTCOMPATDRV - exit(0x00000000)} 02:57:15.753
ndv:      Forcing driver install:
ndv:           Inf Name       - oemvista.inf
ndv:           Driver Date    - 11/05/2014
ndv:           Driver Version - 9.0.0.21
sto:      {Setup Import Driver Package: c:\program files\tap-windows\driver
    \oemvista.inf} 02:57:15.769
inf:           Provider: TAP-Windows Provider V9
inf:           Class GUID: {4d36e972-e325-11ce-bfc1-08002be10318}
inf:           Driver Version: 11/05/2014,9.00.00.21
inf:           Catalog File: tap0901.cat
sto:           {Copy Driver Package: c:\program files\tap-windows\driver\
    oemvista.inf} 02:57:15.784
sto:                Driver Package = c:\program files\tap-windows\driver\
    oemvista.inf
sto:                Flags          = 0x00000007
sto:                Destination    = C:\Users\sijansen\AppData\Local\Temp\{
    b44b285d-2729-f94f-bd1c-c93a111af2a6}
sto:                Copying driver package files to 'C:\Users\sijansen\
    AppData\Local\Temp\{b44b285d-2729-f94f-bd1c-c93a111af2a6}'.
flq:                Copying 'c:\program files\tap-windows\driver\oemvista.
    inf' to 'C:\Users\sijansen\AppData\Local\Temp\{b44b285d-2729-f94f-bd1c-
    c93a111af2a6}\oemvista.inf'.
flq:                Copying 'c:\program files\tap-windows\driver\tap0901.
    cat' to 'C:\Users\sijansen\AppData\Local\Temp\{b44b285d-2729-f94f-bd1c-
    c93a111af2a6}\tap0901.cat'.
flq:                Copying 'c:\program files\tap-windows\driver\tap0901.
    sys' to 'C:\Users\sijansen\AppData\Local\Temp\{b44b285d-2729-f94f-bd1c-
```

```
        c93a111af2a6}\tap0901.sys'.
     sto:           {Copy Driver Package: exit(0x00000000)} 02:57:15.831
     pol:           {Driver package policy check} 02:57:15.878
     pol:           {Driver package policy check - exit(0x00000000)}
        02:57:15.878
     sto:           {Stage Driver Package: C:\Users\sijansen\AppData\Local\Temp
        \{b44b285d-2729-f94f-bd1c-c93a111af2a6}\oemvista.inf} 02:57:15.878
     inf:                {Query Configurability: C:\Users\sijansen\AppData\Local
        \Temp\{b44b285d-2729-f94f-bd1c-c93a111af2a6}\oemvista.inf} 02:57:15.878
     inf:                     Driver package 'oemvista.inf' is configurable.
     inf:                {Query Configurability: exit(0x00000000)} 02:57:15.878
     flq:                Copying 'C:\Users\sijansen\AppData\Local\Temp\{b44b285d
        -2729-f94f-bd1c-c93a111af2a6}\oemvista.inf' to 'C:\Windows\System32\
        DriverStore\Temp\{d3bb3c40-e840-4341-ac13-63b534eafa3a}\oemvista.inf'.
     flq:                Copying 'C:\Users\sijansen\AppData\Local\Temp\{b44b285d
        -2729-f94f-bd1c-c93a111af2a6}\tap0901.cat' to 'C:\Windows\System32\
        DriverStore\Temp\{d3bb3c40-e840-4341-ac13-63b534eafa3a}\tap0901.cat'.
     flq:                Copying 'C:\Users\sijansen\AppData\Local\Temp\{b44b285d
        -2729-f94f-bd1c-c93a111af2a6}\tap0901.sys' to 'C:\Windows\System32\
        DriverStore\Temp\{d3bb3c40-e840-4341-ac13-63b534eafa3a}\tap0901.sys'.
     sto:                {DRIVERSTORE IMPORT VALIDATE} 02:57:15.909
     sig:                     {_VERIFY_FILE_SIGNATURE} 02:57:15.925
     sig:                          Key      = oemvista.inf
     sig:                          FilePath = C:\Windows\System32\DriverStore\
        Temp\{d3bb3c40-e840-4341-ac13-63b534eafa3a}\oemvista.inf
     sig:                          Catalog  = C:\Windows\System32\DriverStore\
        Temp\{d3bb3c40-e840-4341-ac13-63b534eafa3a}\tap0901.cat
!    sig:                          Verifying file against specific (valid)
    catalog failed! (0x800b0109)
!    sig:                          Error 0x800b0109: A certificate chain
    processed, but terminated in a root certificate which is not trusted by the
    trust provider.
     sig:                     {_VERIFY_FILE_SIGNATURE exit(0x800b0109)}
        02:57:15.925
     sig:                     {_VERIFY_FILE_SIGNATURE} 02:57:15.925
     sig:                          Key      = oemvista.inf
     sig:                          FilePath = C:\Windows\System32\DriverStore\
        Temp\{d3bb3c40-e840-4341-ac13-63b534eafa3a}\oemvista.inf
     sig:                          Catalog  = C:\Windows\System32\DriverStore\
        Temp\{d3bb3c40-e840-4341-ac13-63b534eafa3a}\tap0901.cat
     sig:                          Success: File is signed in Authenticode(tm)
        catalog.
     sig:                          Error 0xe0000242: The publisher of an
        Authenticode(tm) signed catalog has not yet been established as trusted
        .
```

```
     sig:                    {_VERIFY_FILE_SIGNATURE exit(0xe0000242)}
         02:57:15.956
!    sig:                    Driver package signer is unknown, but user trusts
    signer.
     sig:                    Driver package certificate was successfully
         installed.
     sto:               {DRIVERSTORE IMPORT VALIDATE: exit(0x00000000)}
         02:57:17.784
     sig:               Signer Score = 0x0F000000
     sig:               Signer Name  = OpenVPN Technologies, Inc.
     sto:               {DRIVERSTORE IMPORT BEGIN} 02:57:17.784
     sto:               {DRIVERSTORE IMPORT BEGIN: exit(0x00000000)}
         02:57:17.784
     cpy:               {Copy Directory: C:\Windows\System32\DriverStore\Temp\{
         d3bb3c40-e840-4341-ac13-63b534eafa3a}} 02:57:17.784
     cpy:                    Target Path = C:\Windows\System32\DriverStore\
         FileRepository\oemvista.inf_amd64_690431ea2d4f48b2
     cpy:               {Copy Directory: exit(0x00000000)} 02:57:17.799
     idb:               {Register Driver Package: C:\Windows\System32\
         DriverStore\FileRepository\oemvista.inf_amd64_690431ea2d4f48b2\oemvista
         .inf} 02:57:17.799
     idb:                    Created driver package object 'oemvista.
         inf_amd64_690431ea2d4f48b2' in DRIVERS database node.
     idb:                    Created driver INF file object 'oem11.inf' in
         DRIVERS database node.
     idb:                    Registered driver package 'oemvista.
         inf_amd64_690431ea2d4f48b2' with 'oem11.inf'.
     idb:               {Register Driver Package: exit(0x00000000)}
         02:57:17.799
     idb:               {Publish Driver Package: C:\Windows\System32\
         DriverStore\FileRepository\oemvista.inf_amd64_690431ea2d4f48b2\oemvista
         .inf} 02:57:17.799
     idb:                    Activating driver package 'oemvista.
         inf_amd64_690431ea2d4f48b2'.
     cpy:                    Published 'oemvista.inf_amd64_690431ea2d4f48b2\
         oemvista.inf' to 'oem11.inf'.
     idb:                    Indexed 3 device IDs for 'oemvista.
         inf_amd64_690431ea2d4f48b2'.
     sto:                    Flushed driver database node 'DRIVERS'. Time = 16
         ms
     sto:                    Flushed driver database node 'SYSTEM'. Time = 0 ms
     idb:               {Publish Driver Package: exit(0x00000000)} 02:57:17.815
     sto:               {DRIVERSTORE IMPORT END} 02:57:17.831
     sig:                    Installed catalog 'tap0901.cat' as 'oem11.cat'.
     sto:               {DRIVERSTORE IMPORT END: exit(0x00000000)} 02:57:17.846
     sto:          {Stage Driver Package: exit(0x00000000)} 02:57:17.846
```

```
sto:      {Setup Import Driver Package - exit (0x00000000)} 02:57:17.862
dvi:      Searching for hardware ID(s):
dvi:           tap0901
dvi:      Class GUID of device changed to: {4d36e972-e325-11ce-bfc1-08002
    be10318}.
dvi:      {Plug and Play Service: Device Install for ROOT\NET\0000}
ndv:           Driver INF Path: C:\Windows\INF\oem11.inf
ndv:           Driver Node Name: oemvista.inf:3beb73aff103cc24:tap0901.ndi
    :9.0.0.21:tap0901
ndv:           Driver Store Path: C:\Windows\System32\DriverStore\
    FileRepository\oemvista.inf_amd64_690431ea2d4f48b2\oemvista.inf
dvi:           Searching for hardware ID(s):
dvi:                tap0901
dvi:           Class GUID of device changed to: {4d36e972-e325-11ce-bfc1
    -08002be10318}.
ndv:           {Core Device Install} 02:57:17.893
ndv:                {Install Device - ROOT\NET\0000} 02:57:17.893
ndv:                     Parent device: HTREE\ROOT\0
sto:                     {Configure Driver Package: C:\Windows\System32\
    DriverStore\FileRepository\oemvista.inf_amd64_690431ea2d4f48b2\oemvista
    .inf}
sto:                          Source Filter  = tap0901
inf:                          Class GUID     = {4d36e972-e325-11ce-bfc1
    -08002be10318}
inf:                          Class Options  = Configurable
inf:                          {Configure Driver: TAP-Windows Adapter V9}
inf:                               Section Name = tap0901.ndi
inf:                               {Add Service: tap0901}
inf:                                    Start Type    = 3
inf:                                    Service Type  = 1
inf:                                    Error Control = 1
inf:                                    Image Path    = \SystemRoot\
    System32\drivers\tap0901.sys
inf:                                    Display Name  = TAP-Windows Adapter
     V9
inf:                                    Group         = NDIS
inf:                                    Created new service 'tap0901'.
inf:                               {Add Service: exit(0x00000000)}
inf:                               Hardware Id  = tap0901
inf:                               {Configure Driver Configuration: tap0901
    .ndi}
inf:                                    Service Name  = tap0901
inf:                                    Config Flags  = 0x00000000
inf:                               {Configure Driver Configuration: exit(0
    x00000000)}
inf:                          {Configure Driver: exit(0x00000000)}
```

```
     flq:                         Copying 'C:\Windows\System32\DriverStore\
         FileRepository\oemvista.inf_amd64_690431ea2d4f48b2\tap0901.sys' to 'C:\
         Windows\System32\drivers\tap0901.sys'.
     dvi:                         Existing files modified, may need to restart
         related services.
     sto:                     {Configure Driver Package: exit(0x00000bc3)}
     ndv:                     Restart required for any devices using this driver
         .
     dvi:                     Install Device: Configuring device (oem11.inf:
         tap0901,tap0901.ndi). 02:57:17.940
     dvi:                     Install Device: Configuring device completed.
         02:57:17.940
     dvi:                     {Restarting Devices} 02:57:17.940
     dvi:                          Restart: ROOT\NET\0000
     dvi:                     {Restarting Devices exit} 02:57:19.252
     ndv:                {Install Device - exit(0x00000000)} 02:57:19.252
     ndv:           {Core Device Install - exit(0x00000000)} 02:57:19.252
     ndv:           Waiting for device post-install to complete. 02:57:19.252
     ndv:           Device post-install completed. 02:57:19.377
     ump:      {Plug and Play Service: Device Install exit(00000000)}
     ndv: {Update Device Driver - exit(00000000)}
<<<  Section end 2016/01/08 02:57:19.408
<<<  [Exit status: SUCCESS]


[Boot Session: 2016/01/08 20:10:02.507]

>>>  [Delete Device - ROOT\NET\0000]
>>>  Section start 2016/01/08 20:19:42.015
      cmd: "C:\Program Files\TAP-Windows\bin\tapinstall.exe" remove tap0901
     dvi: Query-and-Remove succeeded
<<<  Section end 2016/01/08 20:19:42.109
<<<  [Exit status: SUCCESS]
[...]
```

Anhang C

Clientseitige OpenVPN Logdatei `C:\Program Files\OpenVPN\doc\client.log`

```
Fri Jan 08 19:19:52 2016 OpenVPN 2.3.9 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO]
    [PKCS11] [IPv6] built on Dec 16 2015
Fri Jan 08 19:19:52 2016 library versions: OpenSSL 1.0.1q 3 Dec 2015, LZO 2.09
Enter Management Password:
Fri Jan 08 19:19:52 2016 MANAGEMENT: TCP Socket listening on [AF_INET
    ]127.0.0.1:25340
Fri Jan 08 19:19:52 2016 Need hold release from management interface, waiting...
Fri Jan 08 19:19:53 2016 MANAGEMENT: Client connected from [AF_INET
    ]127.0.0.1:25340
Fri Jan 08 19:19:53 2016 MANAGEMENT: CMD 'state on'
Fri Jan 08 19:19:53 2016 MANAGEMENT: CMD 'log all on'
Fri Jan 08 19:19:53 2016 MANAGEMENT: CMD 'hold off'
Fri Jan 08 19:19:53 2016 MANAGEMENT: CMD 'hold release'
Fri Jan 08 19:19:53 2016 Socket Buffers: R=[65536->65536] S=[65536->65536]
Fri Jan 08 19:19:53 2016 MANAGEMENT: >STATE:1452277193,RESOLVE,,,
Fri Jan 08 19:19:53 2016 UDPv4 link local: [undef]
Fri Jan 08 19:19:53 2016 UDPv4 link remote: [AF_INET]192.168.0.106:1194
Fri Jan 08 19:19:53 2016 MANAGEMENT: >STATE:1452277193,WAIT,,,
Fri Jan 08 19:19:53 2016 MANAGEMENT: >STATE:1452277193,AUTH,,,
Fri Jan 08 19:19:53 2016 TLS: Initial packet from [AF_INET]192.168.0.106:1194,
    sid=228e601d 0ad69cdd
Fri Jan 08 19:19:53 2016 VERIFY OK: depth=1, C=DE, ST=NRW, L=Koeln, O=Fort-
    Funston, OU=MyOrganizationalUnit, CN=Fort-Funston CA, name=VPN_SERVER,
    emailAddress=me@myhost.mydomain
Fri Jan 08 19:19:53 2016 VERIFY OK: nsCertType=SERVER
Fri Jan 08 19:19:53 2016 VERIFY OK: depth=0, C=DE, ST=NRW, L=Koeln, O=Fort-
    Funston, OU=MyOrganizationalUnit, CN=ovpnserver, name=VPN_SERVER,
    emailAddress=me@myhost.mydomain
Fri Jan 08 19:19:53 2016 Data Channel Encrypt: Cipher 'BF-CBC' initialized with
    128 bit key
Fri Jan 08 19:19:53 2016 Data Channel Encrypt: Using 160 bit message hash 'SHA1'
     for HMAC authentication
Fri Jan 08 19:19:53 2016 Data Channel Decrypt: Cipher 'BF-CBC' initialized with
    128 bit key
Fri Jan 08 19:19:53 2016 Data Channel Decrypt: Using 160 bit message hash 'SHA1'
     for HMAC authentication
Fri Jan 08 19:19:53 2016 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-
    AES256-SHA, 2048 bit RSA
Fri Jan 08 19:19:53 2016 [ovpnserver] Peer Connection Initiated with [AF_INET
    ]192.168.0.106:1194
```

```
Fri Jan 08 19:19:54 2016 MANAGEMENT: >STATE:1452277194,GET_CONFIG,,,
Fri Jan 08 19:19:56 2016 SENT CONTROL [ovpnserver]: 'PUSH_REQUEST' (status=1)
Fri Jan 08 19:19:56 2016 PUSH: Received control message: 'PUSH_REPLY,route
    10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5'
Fri Jan 08 19:19:56 2016 OPTIONS IMPORT: timers and/or timeouts modified
Fri Jan 08 19:19:56 2016 OPTIONS IMPORT: --ifconfig/up options modified
Fri Jan 08 19:19:56 2016 OPTIONS IMPORT: route options modified
Fri Jan 08 19:19:56 2016 ROUTE_GATEWAY 172.16.52.2/255.255.255.0 I=8 HWADDR=00:0
    c:29:f7:8c:cc
Fri Jan 08 19:19:56 2016 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Fri Jan 08 19:19:56 2016 MANAGEMENT: >STATE:1452277196,ASSIGN_IP,,10.8.0.6,
Fri Jan 08 19:19:56 2016 open_tun, tt->ipv6=0
Fri Jan 08 19:19:56 2016 TAP-WIN32 device [Ethernet] opened: \\.\Global\{782
    BFF50-1E02-4F5D-83E5-6D8A77C44D09}.tap
Fri Jan 08 19:19:56 2016 TAP-Windows Driver Version 9.21
Fri Jan 08 19:19:56 2016 Notified TAP-Windows driver to set a DHCP IP/netmask of
     10.8.0.6/255.255.255.252 on interface {782BFF50-1E02-4F5D-83E5-6D8A77C44D09
    } [DHCP-serv: 10.8.0.5, lease-time: 31536000]
Fri Jan 08 19:19:56 2016 Successful ARP Flush on interface [7] {782BFF50-1E02-4
    F5D-83E5-6D8A77C44D09}
Fri Jan 08 19:20:01 2016 TEST ROUTES: 1/1 succeeded len=1 ret=1 a=0 u/d=up
Fri Jan 08 19:20:01 2016 MANAGEMENT: >STATE:1452277201,ADD_ROUTES,,,
Fri Jan 08 19:20:01 2016 C:\Windows\system32\route.exe ADD 10.8.0.1 MASK
    255.255.255.255 10.8.0.5
Fri Jan 08 19:20:01 2016 ROUTE: CreateIpForwardEntry succeeded with
    dwForwardMetric1=20 and dwForwardType=4
Fri Jan 08 19:20:01 2016 Route addition via IPAPI succeeded [adaptive]
Fri Jan 08 19:20:01 2016 Initialization Sequence Completed
Fri Jan 08 19:20:01 2016 MANAGEMENT: >STATE:1452277201,CONNECTED,SUCCESS
    ,10.8.0.6,192.168.0.106
Fri Jan 08 19:39:13 2016 [ovpnserver] Inactivity timeout (--ping-restart),
    restarting
Fri Jan 08 19:39:13 2016 SIGUSR1[soft,ping-restart] received, process restarting
Fri Jan 08 19:39:13 2016 MANAGEMENT: >STATE:1452278353,RECONNECTING,ping-restart
    ,,
Fri Jan 08 19:39:13 2016 Restart pause, 2 second(s)
Fri Jan 08 19:39:15 2016 Socket Buffers: R=[65536->65536] S=[65536->65536]
Fri Jan 08 19:39:15 2016 MANAGEMENT: >STATE:1452278355,RESOLVE,,,
Fri Jan 08 19:39:15 2016 UDPv4 link local: [undef]
Fri Jan 08 19:39:15 2016 UDPv4 link remote: [AF_INET]192.168.0.106:1194
Fri Jan 08 19:39:15 2016 MANAGEMENT: >STATE:1452278355,WAIT,,,
Fri Jan 08 19:39:15 2016 MANAGEMENT: >STATE:1452278355,AUTH,,,
Fri Jan 08 19:39:15 2016 TLS: Initial packet from [AF_INET]192.168.0.106:1194,
    sid=94929cd1 94bdeb5e
Fri Jan 08 19:39:15 2016 VERIFY OK: depth=1, C=DE, ST=NRW, L=Koeln, O=Fort-
    Funston, OU=MyOrganizationalUnit, CN=Fort-Funston CA, name=VPN_SERVER,
```

```
    emailAddress=me@myhost.mydomain
Fri Jan 08 19:39:15 2016 VERIFY OK: nsCertType=SERVER
Fri Jan 08 19:39:15 2016 VERIFY OK: depth=0, C=DE, ST=NRW, L=Koeln, O=Fort-
    Funston, OU=MyOrganizationalUnit, CN=ovpnserver, name=VPN_SERVER,
    emailAddress=me@myhost.mydomain
Fri Jan 08 19:39:15 2016 Data Channel Encrypt: Cipher 'BF-CBC' initialized with
    128 bit key
Fri Jan 08 19:39:15 2016 Data Channel Encrypt: Using 160 bit message hash 'SHA1'
     for HMAC authentication
Fri Jan 08 19:39:15 2016 Data Channel Decrypt: Cipher 'BF-CBC' initialized with
    128 bit key
Fri Jan 08 19:39:15 2016 Data Channel Decrypt: Using 160 bit message hash 'SHA1'
     for HMAC authentication
Fri Jan 08 19:39:15 2016 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-
    AES256-SHA, 2048 bit RSA
Fri Jan 08 19:39:15 2016 [ovpnserver] Peer Connection Initiated with [AF_INET
    ]192.168.0.106:1194
Fri Jan 08 19:39:16 2016 MANAGEMENT: >STATE:1452278356,GET_CONFIG,,,
Fri Jan 08 19:39:17 2016 SENT CONTROL [ovpnserver]: 'PUSH_REQUEST' (status=1)
Fri Jan 08 19:39:17 2016 PUSH: Received control message: 'PUSH_REPLY,route
    10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5'
Fri Jan 08 19:39:17 2016 OPTIONS IMPORT: timers and/or timeouts modified
Fri Jan 08 19:39:17 2016 OPTIONS IMPORT: --ifconfig/up options modified
Fri Jan 08 19:39:17 2016 OPTIONS IMPORT: route options modified
Fri Jan 08 19:39:17 2016 Preserving previous TUN/TAP instance: Ethernet
Fri Jan 08 19:39:17 2016 Initialization Sequence Completed
Fri Jan 08 19:39:17 2016 MANAGEMENT: >STATE:1452278357,CONNECTED,SUCCESS
    ,10.8.0.6,192.168.0.106
Fri Jan 08 19:39:27 2016 C:\Windows\system32\route.exe DELETE 10.8.0.1 MASK
    255.255.255.255 10.8.0.5
Fri Jan 08 19:39:27 2016 Route deletion via IPAPI succeeded [adaptive]
Fri Jan 08 19:39:27 2016 Closing TUN/TAP interface
Fri Jan 08 19:39:27 2016 SIGTERM[hard,] received, process exiting
Fri Jan 08 19:39:27 2016 MANAGEMENT: >STATE:1452278367,EXITING,SIGTERM,,
```

Anhang D

Cheatsheet zur Anwendungsanalyse des OpenVPN Clients

Relevante persistente Spuren des OpenVPN Clients sind in den folgenden Verzeichnissen zu finden.

Lfd. Nr.	Datei	Beschreibung
1	`C:\Program Files\OpenVPN`	Installationsverzeichnis des OpenVPN Client
2	`C:\Program Files\OpenVPN\bin`	Subverzeichnis enthält die binären Dateien, die zur Ausführung des OpenVPN Clients benötigt werden
3	`C:\Program Files\OpenVPN\config`	Subverzeichnis enthält die Konfiguration des OpenVPN Client (Ablageort für das Client Package). Das Verzeichnis bleibt nach der Deinstallation erhalten.
4	`C:\Program Files\OpenVPN\log`	Subverzeichnis enthält Log-Dateien, die beim Verbindungsaufbau und -abbau vom OpenVPN Client erzeugt werden
5	`C:\Program Files\TAP-Windows`	Installationsverzeichnis von TAP-Windows
6	`C:\Program Files\TAP-Windows\driver`	Subverzeichnis enthält Treiber für die TAP-Netzwerkschnittstelle
7	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OpenVPN`	OpenVPN Verzeichnis im Windows Start Menü
8	`C:\ProgramData\Microsoft\Windows\Start Menu\Programs\TAP-Windows`	Verzeichnis der Start Menü Einträge von TAP-Windows

Tabelle D.1: Auflistung der Verzeichnisse relevanter persistenter Spuren des OpenVPN Clients (Auszug)

Mit welchen Parametern ist der OpenVPN Client konfiguriert?

Wichtige Informationen zur Konfiguration des OpenVPN Clients sind in der Clientkonfigurationsdatei mit der Dateiendung `ovpn` im Subverzeichnis `config` enthalten. Aus der Datei kann unter anderem der Hostname des entfernten OpenVPN Server entnommen werden. Der Distinguished Name des Clients kann dem Subject Feld im Clientzertifikat, das sich ebenfalls im Subverzeichnis `config` befindet,

entnommen werden.

Wann wurde die letzte VPN-Verbindung zu welchem OpenVPN Server aufgebaut?

Zur Beantwortung der Frage, wann die letzte VPN-Verbindung aufgebaut wurde, wird die clientseitige Logdatei `client.log` im Subverzeichnis `log` des OpenVPN Installationsverzeichnis analysiert. Es handelt sich um eine textbasierte Logdatei, die Informationen zum Verbindungsaufbau und -abbau enthält.

War der OpenVPN Client auf einem System installiert?

Die Frage, ob ein OpenVPN Client auf einem System installiert war, wird mithilfe von Spuren des Clients beantwortet, die nach der Deinstallation erhalten bleiben. Die robusteste Spur bildet dabei die Datei `C:\Windows\INF\setupapi.dev.log`. Die textbasierte Datei enthält mit Bezug zum OpenVPN Client Informationen zur Installation der von OpenVPN benötigten TAP-Schnittstelle `tap0901`. Die enthaltenen Zeitstempel spiegeln den Installations- und Deinstallationszeitpunkt wieder.

```
>>>  [Device Install (UpdateDriverForPlugAndPlayDevices) - tap0901]
>>>  Section start 2016/01/08 02:57:15.581
[...]
>>>  [Delete Device - ROOT\NET\0000]
>>>  Section start 2016/01/08 20:19:42.015
      cmd: "C:\Program Files\TAP-Windows\bin\tapinstall.exe" remove tap0901
```

Eine weitere wichtige Spur bildet das Subverzeichnis `config` im Installationsverzeichnis des OpenVPN Clients, das nach der Deinstallation erhalten bleibt und die Clientkonfigurationsdatei enthält. So kann die ehemalige Konfiguration des Clients rekonstruiert werden. Weitere Spuren, die die (ehemalige) Existenz des OpenVPN Clients auf einem System belegen sind in dem Registry Key `HKLM\SOFTWARE\OpenVPN-GUI` und dem Windows Prefetch Verzeichnis `C:\Windows\Prefetch` zu finden. Die vom OpenVPN Client erzeugten Dateien sind der folgenden Auflistung zu entnehmen. Es ist zu beachten, dass der Teil nach dem Bindestrich im Dateinamen variabel ist.

```
C:\Windows\Prefetch\OPENVPN-INSTALL-2.3.9-I601-X8-14A92838.pf
C:\Windows\Prefetch\OPENVPNSERV.EXE-F84A4D0B.pf
C:\Windows\Prefetch\TAP-WINDOWS.EXE-14AB1A49.pf
C:\Windows\Prefetch\TAPINSTALL.EXE-A37B18C4.pf
C:\Windows\Prefetch\DRVINST.EXE-4CB4314A.pf
C:\Windows\Prefetch\OPENVPN-GUI.EXE-52CF0A3D.pf
C:\Windows\Prefetch\OPENVPN.EXE-1E8C6F93.pf
C:\Windows\Prefetch\UNINSTALL.EXE-F3C4FC36.pf
C:\Windows\Prefetch\AU_.EXE-0723E3F4.pf
C:\Windows\Prefetch\BU_.EXE-88237329.pf
```