TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Systemarchitektur, Professur Datenschutz und Datensicherheit

Komplexpraktikum
Datenschutzfreundliche Technologien im Internet

DATENSCHUTZ-ANALYSE VON WINDOWS 10 ENTERPRISE LTSB

Kilian Becher

Christoph Hofmann

Paul Völker

PRAKTIKUMSBERICHT

Betreuer
Dr. Stefan Köpsell

Eingereicht am: 23. Mai 2016

INHALTSVERZEICHNIS

1 EINLEITUNG

Bereits vor der offiziellen Markteinführung von Microsofts neuem Betriebssystem Windows 10 häuften sich Berichte zu dessen Datenschutzunfreundlichkeit. Im Fokus steht seitdem Microsofts Datensammelleidenschaft, die nicht nur Namen, Geschlecht und Kalendereinträge einbezieht, sondern sich auch über Standortinformationen bis hin zu Web-Browser-Verläufen erstreckt [Bag15]. Derartige Erkenntnisse führen zunehmend zu Vergleichen mit einer privaten Abhöranlage. [RP15]
Anfangs bezogen sich die Untersuchungen zwar noch auf die Technical Preview des Systems [Mer14], doch auch nach Veröffentlichung der finalen Home-Edition von Windows 10 für Privatanwender änderte sich wenig an den Resultaten derartiger Analysen [Ant15].

Da Datenschutz nicht nur für Privatpersonen von Bedeutung ist, sondern auch für Organisationen und Unternehmen hohe Priorität hat, rücken letztere in den Fokus der folgenden Betrachtungen.
Ziel dieser Arbeit soll es daher sein, aufbauend auf den Erkenntnissen früherer Untersuchungen, die Datenschutz(un)freundlichkeit von Windows 10 im Kontext der Technischen Universität Dresden zu bewerten.
Neben genannter Home-Edition, welche bisher die meiste Aufmerksamkeit erhielt, bietet Microsoft jedoch eine Vielzahl weiterer Windows-10-Editionen an [Pro15]. In der zu betrachtenden Umgebung wird vorrangig Windows 10 Enterprise mit dem Zusatzdienst LTSB (vgl. Abschnitt 2.1.3) Verwendung finden. Daher soll diese Version auch im Mittelpunkt der Untersuchungen stehen.

Dazu sollen über einen längeren Zeitraum von mehreren Wochen verschiedene PCs mit Windows 10 überwacht werden. Dabei gilt es, die gesamte Kommunikation zwischen den Test-Rechnern und der Außenwelt, insbesondere Microsoft-Servern, zu registrieren, gegebenenfalls zu entschlüsseln und anschließend bezüglich ihrer Datenschutzfreundlichkeit zu bewerten.
Am Ende dieser Analysen soll eine fundierte Aussage zur Eignung von Windows 10 Enterprise mit LTSB für den Einsatz auf Rechnern der Technischen Universität Dresden getroffen werden.

2 GRUNDLAGEN UND VERWANDTE ARBEITEN

Dieses Kapitel ist in zwei voneinander unabhängige Teile getrennt. Dabei widmet sich der erste dieser beiden Teile der Vermittlung von technischen Grundlagen, welche für die Durchführung der Untersuchungen vonnöten waren. Um den in Abschnitt 3.1 beschriebenen Versuchsaufbau und die damit erzielten Ergebnisse nachvollziehen zu können, sind diese unerlässlich.
Der zweite Teil hingegen gibt einen Überblick über verwandte Arbeiten, welche als Einstieg in die durchgeführte Datenschutz-Analyse dienen. Dabei wird auf verschiedene Schwerpunkte und Betrachtungswinkel eingegangen.

2.1 TECHNISCHE GRUNDLAGEN

Wie einführend erläutert, sollen im Folgenden nötige technische Grundlagen vermittelt werden. Aus ersten Voruntersuchungen wurde deutlich, dass ein großer Teil der Kommunikation von Windows 10 verschlüsselt erfolgt. Dies musste im Versuchsaufbau berücksichtigt werden.
Bei der Verschlüsselung kommt das Netzwerkprotokoll Hypertext Transfer Protocol Secure (HTTPS) zum Einsatz, dessen Verschlüsselung auf dem Protokoll Transport Layer Security (TLS) basiert. Abschnitt 2.1.1 gibt einen Überblick über die Funktionsweise dieses Protokolls. Um die übertragenen Inhalte lesen zu können, ist eine Analyse-Technik nötig, mit der die TLS-basierte Verschlüsselung umgangen werden kann. Für diesen Anwendungsfall bietet sich ein Verfahren an, welches als Man-in-the-Middle-Angriff (MitM-Angriff) bezeichnet wird. Dieser wird in Abschnitt 2.1.2 vorgestellt.
Darüber hinaus erfolgt in Abschnitt 2.1.3 eine kurze Vorstellung der Edition Windows 10 Enterprise und der hierfür verfügbaren Zusatzfunktion Long Term Servicing Branch (LTSB). Wie in Kapitel 1 beschrieben, steht diese kurz als Windows 10 Enterprise LTSB bezeichnete Version im Mittelpunkt der Betrachtungen.

2.1.1 TRANSPORT LAYER SECURITY

Bei Transport Layer Security, vormals Secure Socket Layer (SSL), handelt es sich um ein Verschlüsselungsprotokoll, welches die Ende-zu-Ende-Verschlüsselung von Inhalten auf Anwendungsebene ermöglicht [Opp09]. Dabei greift es sowohl auf symmetrische als auch auf asymmetrische Kryptosysteme zurück [DR08]. Es kann daher als hybrides Verschlüsselungsprotokoll betrachtet werden.

Einer der häufigsten Anwendungsfälle von TLS ist die Kombination mit dem Hypertext Transfer Protocol (HTTP) zu HTTPS, wobei das angehängte „S" für „Secure" steht. Damit lassen sich gewöhnliche Web-Kommunikationen verschlüsseln. Anders als bei HTTP wird HTTPS-Kommunikation standardmäßig nicht über Port 80, sondern über Port 443 geleitet [TW12].

Fundamentaler Bestandteil und Ausgangspunkt einer jeden TLS-Verbindung ist der TLS-Handshake. Dieser dient der Etablierung eines gemeinsamen Schlüssels. Dazu wird gemäß [TW12] wie folgt vorgegangen:

- Schritt 1: Der Client sendet eine Verbindungsanfrage an den Server. Diese Anfrage beinhaltet neben der TLS-Version und bevorzugten Algorithmen auch eine Nonce N_C („Number used only once").
- Schritt 2: Nun wählt der Server aus den bevorzugten Algorithmen des Clients jene aus, die im Folgenden zum Verschlüsseln und Komprimieren verwendet werden sollen. Diese Auswahl wird dem Client zusammen mit einer Nonce N_S des Servers mitgeteilt.
- Schritt 3: Im dritten Schritt sendet der Server ein Zertifikat $Cert_S$ mit seinem öffentlichen Schlüssel K_S. Zudem können weitere Informationen und Anfragen an den Client erfolgen, beispielsweise eine Anforderung seines öffentlichen Schlüssels inklusive Zertifikat.
- Schritt 4: Der Client validiert das Zertifikat des Servers und stellt so sicher, dass er mit dem gewünschten Gegenüber kommuniziert. Im Anschluss sendet der Client einen Premaster-Schlüssel P an den Server, verschlüsselt mit dessen öffentlichem Schlüssel K_S. Aus diesem Premaster-Schlüssel wird nun, gemeinsam mit den vorab ausgetauschten Nonces, der eigentliche Sitzungsschlüssel K_{CS} ermittelt.

Diese vier Hauptschritte des TLS-Handshakes werden in Abbildung 2.1 grafisch veranschaulicht.

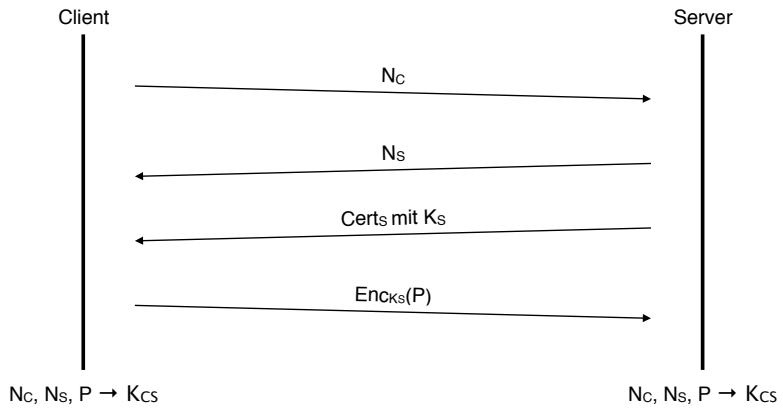

Abbildung 2.1: TLS-Handshake zwischen Client und Server

Nach diesem Verbindungsaufbau wird nun jeder Datenaustausch symmetrisch mit dem Sitzungsschlüssel K_{CS} verschlüsselt [TW12].

Grundlegend für die Sicherheit von TLS ist der Austausch der Zertifikate, welche den öffentlichen Schlüssel und die Domain des Kommunikationspartners beinhalten (vgl. Schritt 3). Ein vertraulicher Datenaustausch kann nur dann gewährleistet werden, wenn beide Parteien ausschließlich mit ihrem gewünschten Gegenüber kommunizieren. Diesen Umstand nutzen Man-in-the-Middle-Angriffe aus.

2.1.2 MAN-IN-THE-MIDDLE-ANGRIFF

Ziel eines Man-in-the-Middle-Angriffs ist es, den gesamten Datenverkehr zwischen zwei Kommunikationspartnern zu kontrollieren. Um dies zu erreichen, gibt sich der Angreifer beiden Partnern gegenüber als der jeweils andere aus. Im allgemeinen Fall unterbricht er dazu jeden einzelnen Datenaustausch, modifiziert oder liest das Gesendete und leitet es an den eigentlichen Empfänger weiter. So verläuft jeder Sendevorgang indirekt vom Sender, über den Man-in-the-Middle zum Empfänger.

Im Fall von TLS-gesicherter Kommunikation ist das Modifizieren und Lesen aufgrund der verwendeten Mechanismen zum Schutz der Integrität und Vertraulichkeit jedoch deutlich aufwendiger. Da der Sitzungsschlüssel von beiden Partnern berechnet und nicht direkt ausgetauscht wird, benötigt der Man-in-the-Middle die zur Berechnung verwendeten Nonces und den Premaster-Schlüssel.
Ein praktikabler Ansatz hierfür ist der Aufbau zweier separater TLS-Verbindungen, jeweils eine mit dem Client und eine mit dem Server. Dabei unterbricht der Angreifer jeden Schritt des TLS-Handshakes ohne das Wissen der beiden Kommunikationspartner.
Beim Verbindungsaufbau, initiiert durch Schritt 1 vom Client, agiert der Man-in-the-Middle als Server und führt in der Folge die Schritte 2 und 3 durch. Um sicherzustellen, dass der Client zur Verschlüsselung des Premaster-Schlüssels den öffentlichen Schlüssel des Angreifers verwendet, muss er diesem gegenüber die Identität des Servers vortäuschen. Das wiederum erfordert ein valides Zertifikat, welches den öffentlichen Schlüssel des Angreifers glaubhaft an die Identität des Servers bindet.
Gelingt dies dem Man-in-the-Middle, so ist nach Schritt 4 die TLS-Verbindung zum Client erfolgreich aufgebaut, ohne dass für diesen ersichtlich ist, dass er in Wirklichkeit nicht mit dem gewünschten Server kommuniziert.
Parallel zu diesem Verbindungsaufbau leitet der Angreifer einen TLS-Handshake mit dem Server ein. Dabei agiert er diesem gegenüber als Client und führt die Schritte 1 und 4 aus. Da hier die Schritte 2 und 3 nicht von ihm durchzuführen sind, entfällt die Notwendigkeit eines gefälschten Zertifikates. Abbildung 2.2 veranschaulicht den Ablauf eines vollen Angriffs.

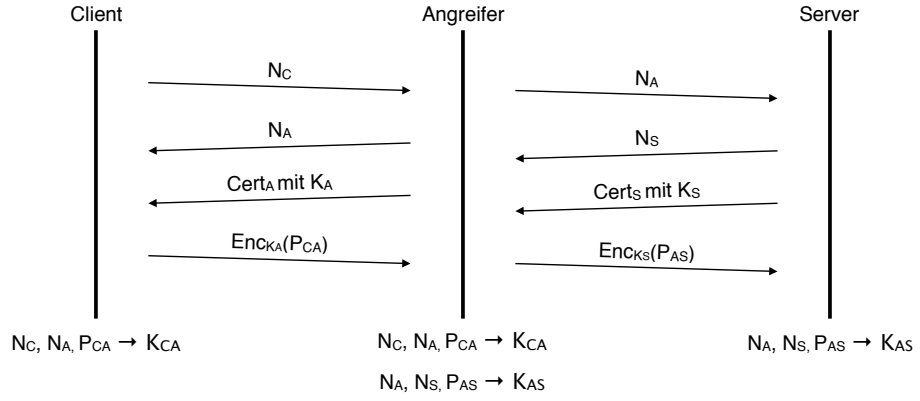

Abbildung 2.2: Man-in-the-Middle-Angriff auf TLS mit zwei separaten TLS-Handshakes

Fortan wird jede Nachricht, die der Client an den Server senden möchte, symmetrisch mit dem Sitzungsschlüssel K_{CA} verschlüsselt. Der Angreifer ermittelt daraus den Klartext, modifiziert bzw. liest diesen und sendet die Nachricht verschlüsselt mit K_{AS} an den Server.
In die andere Richtung werden die Sitzungsschlüssel entsprechend in der umgekehrten Reihenfolge angewandt.

Wie bereits in Abschnitt 2.1.1 angedeutet, beruhen Man-in-the-Middle-Angriffe auf TLS auf dem Vortäuschen einer Identität mithilfe gefälschter Zertifikate. Dem Client ein Zertifikat mit dem öffentlichen Schlüssel des Angreifers und der Domain des Servers zuzuspielen, welches dieser akzeptiert, ist hierbei die schwierigste Aufgabe des Angreifers.
Dazu ist es nötig, dass er entweder eine Certificate Authority täuscht um so an ein gültiges Zertifikat zu gelangen, oder es selbst mit einem eigenen Root-Zertifikat erstellt. Letzteres erfordert jedoch wiederum, dass der Angreifer dieses Root-Zertifikat vorab im System des Clients verankert hat.

Wurde das Zertifikat des Man-in-the-Middle vom Client akzeptiert und anschließend die beiden TLS-Verbindungen erfolgreich aufgebaut, so kann der Angreifer fortan die gesamte Kommunikation zwischen Client und Server lesen und auch verändern.

2.1.3 WINDOWS 10 ENTERPRISE LTSB

Neben Windows 10 Home, Pro, Mobile und Education bietet Microsoft auch eine Windows 10 Enterprise genannte Edition des aktuellen Betriebssystems an. Diese baut auf Pro auf und soll besonders in mittelständischen und größeren Organisationen zum Einsatz kommen. Dazu bietet sie umfangreichere Sicherheitsfunktionen und gewährt Zugang zum Long Term Servicing Branch von Windows 10. [Pro15]

Die Verwendung dieser, kurz Windows 10 Enterprise LTSB genannten Version, ist für viele Unternehmen von großem Vorteil. Einerseits stehen dieser lediglich Updates zur Verfügung, welche sicherheitsrelevant sind oder bestehende Fehler beseitigen. Dabei wird ein Service-Zeitraum von zehn Jahren garantiert. So lässt sich ein dauerhaft nahezu gleichbleibendes System erreichen. [Seb15]
Andererseits beinhaltet Windows 10 Enterprise LTSB viele datenschutzkritische Funktionen der übrigen Editionen nicht. Beispielsweise werden Microsofts neuer Browser Edge und die Assistentin Cortana nicht mitgeliefert [Seb15]. Besonders diese Produkte wurden in früheren Untersuchungen (vgl. Abschnitt 2.2) mit einem hohen Telemetrieaufkommen und der daraus resultierenden Datenschutzunfreundlichkeit von Windows 10 in Verbindung gebracht. [Adm15]

2.2 VERWANDTE ARBEITEN

Im Folgenden sollen Arbeiten vorgestellt und zusammengefasst werden, welche als Grundlagen für das durchgeführte Praktikum verwendet wurden bzw. sich mit ähnlichen Themen beschäftigen.

Der in [Jä15] veröffentlichte Artikel beschreibt, welche Rechte sich Microsoft in seinen aktuellen Nutzungsbedingungen einräumt. Besonders im Fokus liegen dabei Bestimmungen, die das Sammeln, Übertragen und Weitergeben personenbezogener Daten regeln. Aus diesen geht hervor, dass jene Daten sowohl zu Werbezwecken und zum Verkauf als auch für den Dienst Cortana erhoben werden. Zudem wird deutlich, dass die Standard-Datenschutzeinstellungen Microsoft jeweils die weitreichendsten Rechte einräumen.
Gesammelte Daten umfassen unter anderem den Browserverlauf, Informationen zu installierten Apps und deren Nutzungsverhalten, WLAN-Namen und dazugehörige Passwörter, Ort, Kalendereinträge, Kontaktdaten, E-Mails, Spracheingaben und eine globale Advertising-ID des Nutzers.
Ähnliches geht auch aus einem Bericht der Verbraucherzentrale Rheinland-Pfalz hervor. Dabei wird besonders die monetäre Verwertbarkeit der erhobenen Daten betont. [RP15]

Besonders kritische Beobachtungen werden in [Adm15] vorgestellt. Hierbei wird unter anderem von Keyloggern berichtet, welche in regelmäßigen Abständen aufgezeichnete Tastatureingaben an Microsoft-Server senden. Für diesen und weitere Dienste wurde zudem eine Liste von Domains zusammengetragen, mit denen Windows 10 im Leerlauf kommuniziert.

In [Ant15] wird darüber hinaus vor allem über Möglichkeiten der Deaktivierung oder Minimierung des Sammelns und Übertragens von Daten berichtet. Dabei werden die wichtigsten Einstellungen vorgestellt, welche je nach Datenschutzbedürfnis zu deaktivieren sind. Jedoch wird dabei auch angemerkt, dass viele dieser Anpassungen zu Einschränkungen einzelner Dienste wie Cortana führen können.
Allerdings lassen sich einige Einstellungen, wie beispielsweise die Feedback-Häufigkeit des Betriebssystems, nur in Windows 10 Enterprise vollständig deaktivieren. Dies legt nahe, dass die übrigen Editionen mitunter stärker mit Microsoft-Servern kommunizieren.

Darüber hinaus wird in [Bri15] darauf hingewiesen, dass Windows 10 selbst nach der Deaktivierung diverser Telemetriedienste teils noch viele Daten an Microsoft-Server sendet. So werden bei Eingabe von Suchbegriffen in die Suchfunktion selbst dann noch Cortana-bezogene Informationen übertragen, wenn Cortana vollständig deaktiviert wurde. Dies umfasst auch persistente Machine-IDs. Andere Funktionen hingegen, wie beispielsweise Network Connection Status Indicator (NCSI) haben durchaus ihre Daseinsberechtigung und beeinflussen die Privatsphäre des Nutzers zugleich nur minimal.

Dem Artikel in [Bot16] können Informationen zu Microsofts Universal Telemetry Client entnommen werden. Hierbei werden verschiedene Stufen von Telemetrie sowie deren Zweck, Inhalt und Aufbewahrungsform vorgestellt.

Die betrachteten Arbeiten beziehen sich größtenteils auf Windows 10 Home bzw. die Windows Insider Preview. Da im Folgenden die Version Enterprise LTSB im Fokus stehen soll, dienen die genannten Artikel als Grundlage für ähnliche Untersuchungen und Analysen sowie als Anhaltspunkte für spätere Vergleiche.

3 UMSETZUNG

Im Folgenden soll die praktische Umsetzung der Untersuchung vorgestellt werden. Hierzu wird zunächst der Versuchsaufbau beschrieben. Anschließend werden die verschiedenen Szenarien, welche im Rahmen der Untersuchung zum Einsatz kamen, dargestellt. Abschließend soll auf einige Einschränkungen und Probleme der Untersuchung und insbesondere des Versuchsaufbaus hingewiesen werden.

3.1 VERSUCHSAUFBAU UND TECHNISCHE REALISIERUNG

Um die Kommunikation von Windows 10 überwachen und anschließend auswerten zu können, muss der gesamte Datenverkehr von und zu Windows 10 mitgeschnitten werden. Hierzu bietet sich zunächst das freie Netzwerkanalysetool Wireshark [Wir] an. Dieses Analysetool bietet die Möglichkeit, alle Datenpakete und somit den gesamten Datenverkehr mitzuschneiden und anschließend oder währenddessen umfassend zu analysieren. Während unverschlüsselter Netzwerkverkehr über Port 80 direkt gelesen werden kann, ist dies bei TLS-verschlüsseltem Datenverkehr über Port 443 nicht ohne weiteres möglich. Um diese Daten ebenfalls lesen und somit überprüfen zu können, muss die bereits in Abschnitt 2.1.2 vorgestellte Technik eines Man-in-the-Middle-Angriffes zur Anwendung kommen.
Hierzu wird in den durchgeführten Untersuchungen das Softwaretool Burp Suite [Bur] in der kostenlosen Free Edition eingesetzt, welche einige Einschränkungen gegenüber der kostenpflichtigen Professional Edition aufweist. Diese umfassende Security- und Penetration-Testing-Suite bietet viele Funktionen, darunter auch einen Man-in-the-Middle-Proxy, mit dem es möglich wird, auch verschlüsselte Kommunikation zu lesen. Hierzu stellt die Burp Suite ein Zertifikat bereit, welches auf dem zu untersuchenden System hinzugefügt werden muss. Auch die Verwendung eines selbst erstellten Zertifikats ist möglich.
Um die Instanz von Windows 10 bereits von Anfang an überwachen zu können und um mögliche Interferenzen durch die Installation von zusätzlicher Software auszuschließen, sollte die Überwachung nicht direkt auf der Windows-Instanz erfolgen. Bei den durchgeführten Untersuchungen kommt deshalb ein Linux-System zum Einsatz, welches für Windows 10 als Router und Verbindung ins Internet agiert und gleichzeitig den Mitschnitt des Datenverkehrs mittels Wireshark und Burp Suite übernimmt. Als Linux-System wird hierbei die auf Debian Linux basierende und besonders auf Sicherheitstests ausgelegte Distribution KALI Linux [Kal] verwendet. Doch auch die Verwendung jeder anderen Linux-Distribution ist möglich. Die im Folgenden beschriebenen Schritte zur Einrichtung können jedoch je nach verwendeter Distribution leicht abweichen. Der beschriebene Aufbau ist in Abbildung 3.1 schematisch dargestellt.

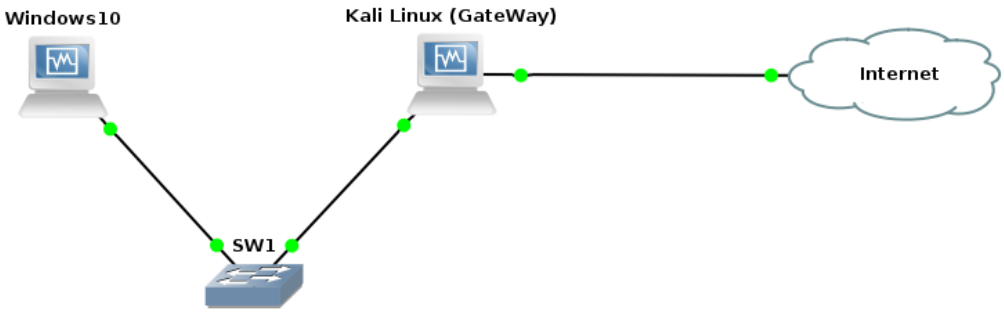

Abbildung 3.1: Schema des Untersuchungsaufbaus

Um überhaupt Netzwerkpakete weiterleiten zu können und somit als Router zu agieren, muss zunächst generelles IP-Forwarding auf dem Linux-System eingeschaltet werden. Dies kann mit folgendem Kommando erreicht werden:

```
1 echo 1 > /proc/sys/net/ipv4/ip_forward
```

Damit das Windows-System über das Dynamic Host Configuration Protocol (DHCP) eine automatische IP-Adresse beziehen kann und Hostnamen mittels des Domain Name System (DNS) abgefragt werden können, muss auf dem Linux-Router des Weiteren der dnsmasq Service gestartet werden, der diese Aufgaben übernimmt. Dies erfolgt durch folgendes Kommando:

```
1 service dnsmasq start
```

Anschließend werden einige Regeln mittels IP-Tables definiert. Zunächst wird eine Network Address Translation (NAT) für die Schnittstelle eth0 eingerichtet, über welche die Verbindung zum Internet aufgebaut wird. Die NAT bewirkt hierbei, dass jedes Paket, welches die Schnittstelle eth0 verlässt, die gleiche Quell-Adresse hat. Das Linux-System fungiert somit wie ein typischer NAT-Router. Zusätzlich werden Forwarding-Regeln eingerichtet, welche den vom Windows-System kommenden TCP-Datenverkehr (Schnittstelle eth0) von Port 80 und 443 auf Port 10000 umleiten, auf dem wiederum der Burp Suite MitM-Proxy lauscht. Vorangegangene Untersuchungen mit Wireshark ergaben, dass Windows 10 ausschließlich diese Standard HTTP- bzw. HTTPS-Ports zur Übertragung von Daten verwendet.

```
1 iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
  iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-ports
    ↪ 10000
3 iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 443 -j REDIRECT --to-ports
    ↪ 10000
```

Damit ist die grundsätzliche Einrichtung des Linux-Systems als Router abgeschlossen. Da nun alle TCP-Datenpakete, welche über Port 80 oder 443 auf dem System ankommen, auf Port 10000 weitergeleitet werden, muss der Burp Suite Proxy jetzt so konfiguriert werden, dass dort die Pakete entgegengenommen werden. Hierzu wird in der Burp Suite im Reiter „Proxy“ unter „Options“ ein sogenannter „Proxy Listener“ auf Port 10000 eingerichtet (vgl. Abbildung 3.2).

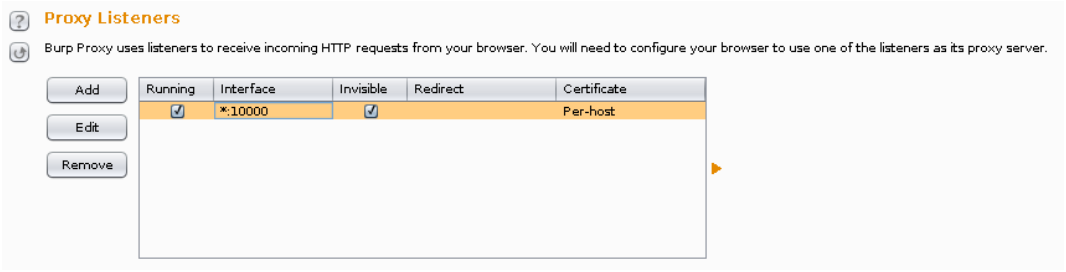

Abbildung 3.2: Einrichtung des Ports auf dem der Burp Suite Proxy lauscht

Die Burp Suite bietet die Funktion, jede Verbindung einzeln zu betrachten und entweder zu verwerfen, weiterzuleiten oder gar manipulierend einzugreifen. Da für die durchgeführte Untersuchung jedoch alle Pakete lediglich mitgeschnitten und möglichst ohne Verzögerung weitergeleitet werden sollen, bietet es sich an, unter „Interception" diese Option, welche standardmäßig aktiviert ist, zu deaktivieren. Somit muss nicht jede einzelne Verbindung explizit vom Untersuchenden bestätigt werden. Alle anderen Einstellungen können auf den voreingestellten Standardwerten belassen werden.
Zusätzlich zum Mitschnitt mit der Burp Suite erfolgt auch ein Mitschnitt mittels Wireshark, welches jeglichen Datenverkehr auf der Schnittstelle eth1, also zwischen dem untersuchten Windows-System und dem Router, aufzeichnet.

Um die Untersuchung möglichst flexibel und effizient zu gestalten und eine einfache Lösung zum Erstellen von Snapshots und Wiederherstellungspunkten zu gewährleisten, werden beide Systeme, sowohl Windows 10 als auch KALI Linux, in jeweils einer eigenen virtuellen Maschine installiert. Dies wird mithilfe der Virtualisierungslösung VirtualBox [Vbo] realisiert und so konfiguriert, dass sich beide virtuellen Maschinen im gleichen Netzwerk befinden.
Darüber hinaus wird im weiteren Verlauf der Untersuchung auch noch ein Test auf realer Hardware durchgeführt, bei dem sowohl für die Windows-10-Instanz als auch für KALI Linux physische Geräte zum Einsatz kommen. Nähere Informationen hierzu finden sich in Abschnitt 3.3.
Die Installation der untersuchten Version Windows 10 Enterprise LTSB erfolgt mit der von der Stabsstelle für Informationssicherheit der Technischen Universität Dresden zur Verfügung gestellten, bereits mit einigen Voreinstellung versehenen, Version von einem virtuellen CD-Laufwerk aus. Die Installation dieser bereits vorkonfigurierten Version läuft automatisch ohne dass weitere Nutzereingaben oder Einstellungen vorgenommen werden müssen. Nach einigen Neustarts muss lediglich ein Benutzername festgelegt und ein entsprechendes Kennwort sowie ein Kennworthinweis angegeben werden. Für die Untersuchungen werden hierfür stets die folgenden Nutzerdaten verwendet:

Verwendete Nutzerdaten	
Benutzername	AABBCC112233KCP
Kennwort	aabbcc
Kennworthinweis	Ersten 6 Zeichen klein

Tabelle 3.1: Übersicht der verwendeten Nutzerdaten

Es bestand die Annahme, derartig auffällige Zeichenketten würden bei einer späteren Untersuchung leicht auffallen. Aus dem gleichen Grund wird dem Benutzerkonto zu einem späteren Zeitpunkt das unter Abbildung 3.3 dargestellte Bild hinzugefügt.
Für den MitM-Angriff mittels Burp Suite und die daraus resultierende Möglichkeit, TLS-verschlüsselte Verbindungen überprüfen zu können, muss dem Windows-System das vom Burp

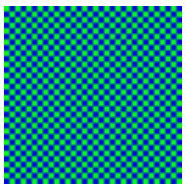

Abbildung 3.3: Verwendetes Benutzerprofilbild

Suite Proxy verwendete Zertifikat als Root-Zertifikat hinzugefügt werden. Dies geschieht durch das folgende Kommando in der Windows-Eingabeaufforderung:

```
1 certutil -addstore "Root" WINCERT.CRT
```

Dabei stellt die Datei WINCERT.CRT das von der Burp Suite verwendete Zertifikat dar. Weitere Einstellungen werden auf dem Windows-System zunächst nicht vorgenommen. Während des gesamten Installations- und Einrichtungsprozesses besteht keine Netzwerkverbindung. Diese wird erst nach einem Neustart aktiviert und die Untersuchung somit gestartet.

3.2 SZENARIEN

Für die Untersuchungen wurden verschiedene Anwendungsszenarien getestet, um so die möglicherweise unterschiedlichen Reaktionen von Windows 10 zu beobachten. Zunächst wurde die Windows-Instanz nur hochgefahren und der angelegte Benutzer wurde angemeldet. Eine weitere Interaktion mit dem System erfolgte vorerst nicht. In einem erweiterten Schritt wurden verschiedene Interaktionen mit dem System vorgenommen. So wurden beispielsweise die Einstellungen geöffnet oder Begriffe in das im Startmenü befindliche Suchfeld eingegeben. Jede Interaktion mit dem System wurde in einem gewissen zeitlichen Abstand ausgeführt und nebenbei genau dokumentiert, um Rückschlüsse auf etwaige Netzwerkkommunikationen zuzulassen. Die Interaktion mit dem System wurde stetig gesteigert, um schließlich eine möglichst realistische Verwendung des Systems zu simulieren. So wurden auch Peripheriegeräte wie etwa externe Speichermedien (USB-Sticks, CDs) und Kopfhörer angeschlossen. Mit dem Aufruf verschiedener Webseiten wurde schließlich auch ein gewisses Surfverhalten simuliert und naturgemäß viel Netzwerkverkehr erzeugt.

3.3 PHYSISCHER TEST

Neben der bereits beschriebenen Variante mit virtuellen Maschinen wurde auch noch ein Test durchgeführt, bei dem physische Maschinen zum Einsatz kamen. Hierdurch sollte untersucht werden, ob sich Windows 10 in einer virtuellen Maschine anders verhält als unter echten Bedingungen wie sie im Regelfall herrschen. Hierzu wurde Windows 10 Enterprise LTSB mittels USB-Stick auf einem herkömmlichen Universitätsbürorechner installiert. Alle Einstellung erfolgten analog zu den in der virtuellen Maschine vorgenommenen und wie bereits oben beschrieben. Als Router kam in diesem Fall ein Laptop zum Einsatz, der mittels WLAN über das Eduroam-Netzwerk an das Internet angebunden wurde. Der Windows-Rechner wurde durch ein Ethernet Kabel über einen Switch mit dem Laptop verbunden.

3.4 EINSCHRÄNKUNGEN UND PROBLEME

Im Folgenden sollen einige Einschränkungen und Probleme, die im Zusammenhang mit der Untersuchung auftraten, erläutert und erklärt werden. Zunächst sei noch einmal erwähnt, dass für die Untersuchung die Free Edition der Burp Suite von Portswigger zum Einsatz kam. Diese kostenlose Variante hat gegenüber der kostenpflichtigen Professional Variante unter anderem den Nachteil, dass vorgenommene Einstellungen nach dem Beenden des Programms nicht persistent bleiben und Mitschnitte nicht gespeichert werden können um sie später erneut mit der Burp Suite zu öffnen. Diese Einschränkungen führten zu der Entwicklung eines eigenen Visualisierungstools, welches in Abschnitt 4.2.1 näher beschrieben wird.

Des Weiteren sei erwähnt, dass die Performance der Burp Suite und damit das Ergebnis der Untersuchung teilweise von der Rechenleistung des Linux-Systems abhängt. So zeigte sich vor allem bei der Verwendung von wenig Ressourcen (Speicher und CPU) für die virtuelle Maschine, dass Verbindungen von der Burp Suite teilweise verworfen oder stark verzögert wurden. Eine Erhöhung der Ressourcen für die virtuelle Maschine brachte hier Besserung. Dennoch besteht bei Verwendung der Burp Suite das Problem, dass insbesondere bei großen Downloads, wie sie etwa bei Updates vorkommen, die Weiterleitung zum Windowsrechner stark verzögert wird. Dies kann im besonderen Fall zu einem Timeout und somit zum Abbruch der Verbindung führen. Ursächlich hierfür ist die Eigenschaft der Burp Suite, die Verbindung zunächst komplett zwischenzuspeichern, um sie anschließend im Ganzen weiterzureichen. Dies geschieht, um die Manipulation der Verbindung zu ermöglichen. Beobachtungen haben jedoch gezeigt, dass dies für die datenschutztechnische Untersuchung keine Einschränkung darstellt, da auch im echten Betrieb Engpässe auftreten und Verbindungen abbrechen können. Abgebrochene Verbindungen werden gegebenenfalls neu aufgebaut und die Dateien zu einem späteren Zeitpunkt erfolgreich geladen.

Auch im Zusammenhang mit der im Anschluss beschriebenen Analyse der Verbindungsdaten soll auf einige Einschränkungen hingewiesen werden. So ist es trotz der Möglichkeit, auch TLS-verschlüsselten Datenverkehr öffnen und mitlesen zu können, oftmals nicht möglich, den Sinn oder die Bedeutung eines übertragenen Inhaltes zu bemessen, da hierbei oft Hashes oder IDs zum Einsatz kommen, deren genaue Bedeutung im Verborgenen bleibt. Dennoch soll im folgenden Kapitel ein Überblick und eine Einordnung der mitgeschnittenen Verbindungen erfolgen.

4 ANALYSE

4.1 TRACES

Bevor im Folgenden die Auswertung und eine Analyse der gewonnenen Daten vorgenommen wird, soll zunächst beschrieben werden, welche Informationen diese Daten enthalten und wie sie charakterisiert werden können. Die Burp Suite stellt unter „HTTP History" jede aufgebaute Verbindung dar. Diese, im Folgenden auch als Traces bezeichneten Verbindungen, werden zunächst nach der vom Sender (Windows 10) ausgehenden Anfrage (Request) und der vom angefragten Gegenüber (z.B. Update-Server) erhaltenen Antwort (Response) unterteilt. Des Weiteren werden für jede Verbindung bestimmte Eigenschaften wie beispielsweise der angefragte Host, die Anfragemethode oder die Größe der Verbindung ermittelt und angezeigt (vgl. 4.1).

#	Host	Method	URL	Length	MIME...	SSL	IP	Time	Listener port
17	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1964	XML	☐	65.55.113.12	14:51:25 ...	8080
18	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:25 ...	8080
19	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1970	XML	☐	65.55.113.12	14:51:25 ...	8080
20	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:25 ...	8080
21	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1964	XML	☐	65.55.113.12	14:51:25 ...	8080
22	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:25 ...	8080
23	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1970	XML	☐	65.55.113.12	14:51:26 ...	8080
24	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:26 ...	8080
25	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1964	XML	☐	65.55.113.12	14:51:26 ...	8080
26	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:26 ...	8080
27	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1970	XML	☐	65.55.113.12	14:51:26 ...	8080
28	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:26 ...	8080
29	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1970	XML	☐	65.55.113.12	14:51:26 ...	8080
30	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:27 ...	8080
31	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1964	XML	☐	65.55.113.12	14:51:27 ...	8080
32	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:27 ...	8080
33	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1964	XML	☐	65.55.113.12	14:51:27 ...	8080
34	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:51:27 ...	8080
35	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1964	XML	☐	65.55.113.12	14:51:27 ...	8080
36	http://officeclient.microsoft.com	HEAD	/config16?services=FontServi...	472	XML	☐	23.97.178.173	14:52:52 ...	8080
37	http://officeclient.microsoft.com	GET	/config16?services=FontServi...	733	XML	☐	23.97.178.173	14:52:52 ...	8080
38	https://fs.microsoft.com	HEAD	/fs/2.2/fontset.json	394	app	☑	23.57.22.139	14:52:53 ...	8443
39	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:52:56 ...	8080
40	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1962	XML	☐	65.55.113.12	14:52:56 ...	8080
41	https://sls.update.microsoft.com	GET	/SLS/%7B9482F4B4-E343-43...	16440	app	☑	157.56.96.54	14:52:56 ...	8443
42	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:52:56 ...	8080
43	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1968	XML	☐	65.55.113.12	14:52:56 ...	8080
44	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:52:56 ...	8080
45	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1962	XML	☐	65.55.113.12	14:52:56 ...	8080
46	http://go.microsoft.com	POST	/fwlink/?LinkID=252669&clcid...	196		☐	23.57.28.45	14:52:56 ...	8080
47	http://dmd.metaservices.microsoft.com	POST	/dms/metadata.svc	1968	XML	☐	65.55.113.12	14:52:57 ...	8080

Abbildung 4.1: Auszug eines Burp-Suite-Mitschnittes

4.2 AUSWERTUNG DER TRACES

4.2.1 VISUALISIERUNGSTOOL

Um die Internetkommunikation auswerten zu können, mussten die Verbindungsinformationen zuerst dargestellt werden. Die Burp Suite bietet die Möglichkeit, in jede einzelne Verbindung hineinzuschauen, jedoch lassen sich die Sitzungen in der kostenlosen Free Edition nicht speichern und wiederherstellen. Aus diesem Grund begann die Entwicklung eines eigenen Visualisierungstools, welches die Möglichkeit der Burp Suite nutzt, selektierte Verbindungsinformationen als XML-Datei zu exportieren.

AUFBAU

Beim Visualisierungstool handelt es sich um eine HTML5-Datei, welche mithilfe von JavaScript die Tabellen erstellt und mit CSS intern einen Style vorgibt. Die Dateien werden mit einem FileReader geladen, welcher in der HTML5 File API [W3C15] spezifiziert ist und anschließend durch JavaScript verarbeitet. Um die Requests und Responses gut lesen zu können, müssen diese grafisch aufbereitet werden. Hierfür werden zwei externe Bibliotheken verwendet. Zum einen vkBeautify [Kir12] um die Struktur von XML-Code wieder herzustellen und zum zweiten highlight.js [Sag16] um das Ergebnis einzufärben. Die Sortierbarkeit der Tabelle wird mit der Bibliothek Table Sort [Ber15] erreicht.

VERGLEICH DER VISUALISIERUNG MIT DER BURP SUITE

Im Laufe der Erstellung des Visualisierungstool wurden viele weitere Vorteile sichtbar. Die Suche nach bestimmten Namen oder IDs gestaltet sich in der Burp Suite als sehr schwierig, ist aber durch die Verwendung von HTML und JavaScript in einem Browser von vornherein gegeben. Des Weiteren kann die Vergleichbarkeit von Kommunikationen deutlich erhöht werden, indem in verschiedenen Browserfenstern verschiedene Sitzungen geöffnet werden. Ein weiterer Vorteil ergibt sich daraus, dass im entwickelten Tool die Größe, bis zu welcher Übertagungsinhalte angezeigt werden, frei einstellbar ist. Bei der Burp Suite wird ab einer bestimmten Größe der Fehler ausgegeben, dass der Inhalt für die Darstellung zu groß ist. Es sind auch Übertragungen aufgetreten, bei denen die Burp Suite den Request und den Response nicht von Base64 nach UTF8 konvertiert hat. Weitere Vorteile des eigenen Visualisierungstool ergeben sich durch die deutlich verbesserte Formatierung und die Nutzung von Highlighting, wodurch die übertragenen Daten viel schneller verständlich werden.

Das entwickelte Tool hat aber auch einen Nachteil. Bisher werden die Kommunikationsmitschnitte vollständig eingelesen, was aber von den Browsern nur bis zu einer Größe von 256 MB unterstützt wird. Eine erweiterte Version, welche diese Begrenzung nicht mehr aufweist, ist durchaus denkbar und wurde bereits ansatzweise umgesetzt.

VERWENDUNG

Zu Beginn muss mit der Burp Suite eine Datei mit Verbindungsinformationen erzeugt werden. Dies geschieht indem auf der Burp-Suite-Oberfläche unter „Proxy“ der Punkt „HTTP History“ angewählt wird. Anschließend werden alle Items markiert und mit einem Rechtsklick und „Save items“ gespeichert. Das Gleiche lässt sich auch unter „Target“ auf der „Site map“ durchführen, jedoch enthalten die so erstellten Dateien nicht den vollständigen Kommunikationsmitschnitt. Um die Größe der resultierenden Datei zu reduzieren, können die größten Daten vor dem Speichern deselektiert und einzeln gesichert werden.

Das Visualisierungstool muss nicht installiert werden. Die Datei DataVisualizer.html muss neben dem „libs“ Ordner liegen und kann mit jedem Browser geöffnet werden, wobei die unterschiedlichen Browser bei der Darstellung von sehr großen Tabellen unterschiedlich schnell sind.

Auf der Oberfläche (vgl. Abbildung 4.2) muss zuerst entschieden werden, in welchem Modus die Datei geladen werden soll. Hier stehen drei Varianten zur Verfügung:

Eingeklappt: Dies ist der Standardmodus. Hierbei werden die Requests und Responses nicht sofort mitgeladen, sondern können über einen Button einzeln geöffnet werden. Die Daten werden dabei formatiert und farblich unterlegt, allerdings werden Informationen von nicht geöffneten Requests und Responses bei einer Suche auch nicht berücksichtigt.

Ausgeklappt: Hierbei werden alle Daten sofort geladen, es findet aber keine Formatierung und kein Highlighting der Requests und Responses statt.

Eingefärbt: In diesem Modus werden alle Daten sofort geladen, eingefärbt und formatiert. Aufgrund der vielen rechenintensiven Operationen kann dies bei sehr vielen Übertragungsdaten sehr lange dauern, weshalb manche Browser die Berechnung nach einer bestimmten Zeit abbrechen.

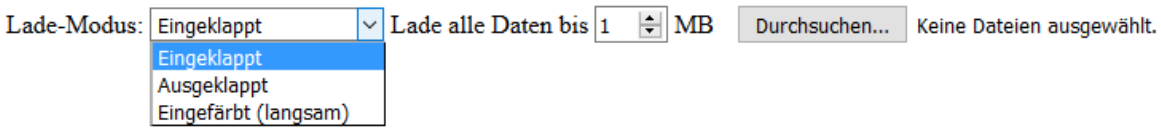

Abbildung 4.2: Oberfläche des Visualisierungstools

Um das Laden und die Arbeit auf den Daten zu beschleunigen, kann anschließend angegeben werden, bis zu welcher Größe Requests und Responses mitgeladen werden. D.h. wenn eine Verbindung größer als der angegebene Wert ist, wird sie dargestellt, nur der Request oder der Response wird weggeschnitten.

Anschließend kann mit dem Durchsuchen-Button eine Datei (bis zu 256 MB), welche von der Burp Suite erstellt wurde, auf dem Rechner ausgewählt werden.

Die Daten werden in Tabellenform dargestellt. Standardmäßig sind die Übertragungen so sortiert wie sie es in der Burp Suite waren, bevor sie exportiert wurden. Es kann aber nach jeder Spalte (außer Request und Response) neu sortiert werden. Die letzte Spalte „Comment“ stellt Kommentare dar, welche in der Burp Suite hinterlegt wurden.

Bei manchen Zeilen (vgl. Abbildung 4.3) sind die Buttons für das Öffnen von Requests und Responses grün gefärbt. Es handelt sich hierbei um den Hinweis, dass in den Übertragungen ein Attribut, welches mit „X-“ beginnt, vorkommt. Diese Attribute werden von Microsoft unter anderem dafür verwendet, Systeminformationen anzugeben.

Beim Laden einer weiteren Datei wird eine zweite Tabelle erstellt, welche unter der ersten dargestellt wird.

time	url	host	port	protocol	method	path	extension	request	status	responselength	mimetype	response	comment
6.1.2016, 11:06:55	http://www.msftncsi.com /ncsi.txt	www.msftncsi.com 92.226.0.209	80	http	GET	/ncsi.txt	txt	Öffnen	200	179	text	Öffnen	

Abbildung 4.3: Grafische Darstellung einer Kommunikation mit dem Visualisierungstool

4.2.2 GRUPPIERUNG DER TRACES

Aufgrund der hohen Anzahl auszuwertender Traces war für diese Arbeit ein strukturiertes Vorgehen bei deren Untersuchung vonnöten. Dabei wurde besonderer Wert auf das Zusammenfassen ähnlicher Kommunikation gelegt.

Im Mittelpunkt der dafür gewählte Herangehensweise steht die Gruppierung der Kommunikationsverläufe nach angesprochenem Domain-Namen. Dadurch lassen sich verschiedene Microsoft-Dienste voneinander trennen, da diese meist eigene Domain-Namen verwenden. Hinzu kommt eine Unterscheidung verschiedener Pfade einer Domain um so die diversen Funktionen und Ressourcen eines Dienstes unterscheiden zu können.

Eine Sortierung nach IP-Adressen wäre nicht zielführend, da für einige Dienste mitunter mehrere externe Server der Firma Akamai betrieben werden und so eine Vielzahl möglicher IP-Adressen für den gleichen Dienst verwendet wird.
Dennoch wurde es für diese Arbeit als zweckdienlich erachtet, zu jeder Gruppe von Traces auch die hierfür aufgerufenen IP-Adressen zu erfassen. Dadurch ließe sich unter Umständen ein Routing-Verhalten von Microsoft-Prozessen ableiten, Zusammenhänge zwischen einzelnen Ereignissen erkennen und gegebenenfalls bestimmte Kommunikationen leichter blockieren.
Nach erfolgreicher Gruppierung gleicher oder ähnlicher Traces rückt die Auswertung und Klassifizierung dieser Gruppen in den Fokus. Dazu werden jeweils die Häufigkeit und die Zeitpunkte der Kommunikation ergänzt. Daraus ergibt sich die Möglichkeit, Parallelen zu den Nutzungsszenarien aufdecken und gegebenenfalls Rückschlüsse auf das Zeitverhalten einzelner Dienste ziehen zu können.
Ein weiteres aussagekräftiges Attribut ist das verwendete Kommunikationsprotokoll mit dem jeweils verwendeten Port. Da die Burp Suite auf HTTP und HTTPS beschränkt ist, kommen hier lediglich die Ports 80 und 443 in Frage. Dennoch reicht diese Information aus, um erkennen zu können, ob die Datenübertragung verschlüsselt wurde (vgl. Abschnitt 2.1.1). Dies ist je nach gesendetem Inhalt bei der späteren Einstufung von großer Bedeutung.

Da jedoch nicht nur das äußere Erscheinungsbild der Übertragungen relevant ist, sind die einzelnen Pakete der Gruppen auch inhaltlich zu untersuchen. Dazu bietet es sich an, zwischen Requests und Responses zu differenzieren.
Sowohl für Requests als auch für Responses ist zuerst der HTTP(S)-Header auszuwerten und nach auffälligen Feldern zu durchsuchen. Besonders relevant sind hierbei Cookies und andere Identifikationsmöglichkeiten.
Anschließend wird der jeweilige Nachrichtenrumpf analysiert, wobei vor allem gesendete bzw. empfangene Daten von Interesse sind, welche Rückschlüsse auf den Nutzer oder dessen Verhalten zulassen. Dazu zählen nicht nur personenbezogene Daten und Eingaben des Nutzers sondern auch wiederkehrende IDs. Über letztere lassen sich mitunter auch Parallelen zu anderen Traces und Diensten erkennen. Diese sind ebenfalls zu vermerken.
Darüber hinaus sind die Serverantworten auf etwaige Befehle zu durchsuchen. Außerdem sollten übertragene Multimediadaten und Programmcodes verzeichnet werden.
Diese Auswertung der Paketinhalte soll anschließend kurz in Form einer Einschätzung mit besonderen Auffälligkeiten zusammengefasst werden.

Schließlich ist die gesamte Gruppe in wenigen Worten zu bewerten. Dabei wird nicht nur auf den Zweck und eine mögliche Initiierung durch den Nutzer sondern auch auf die damit verbundenen Möglichkeiten zur Nutzerprofilierung und andere Risiken eingegangen.

Das hier beschriebene Vorgehen zur Auswertung der Traces spiegelt sich auch in der Aufteilung der Auswertungstabelle wieder, welche in Abschnitt 4.3.1 vorgestellt wird. Die Tabelle enthält zudem auch ein farbliches Schema zur Einstufung der einzelnen Gruppen, welches im Folgenden als Ampelsystem bezeichnet wird.
Dieses Ampelsystem besteht aus vier Stufen, welche durch die Farben grün, gelb, orange und rot gekennzeichnet sind. Dadurch wird eine Skala von unbedenklicher Kommunikation (grün) bis hin zu sehr bedenklicher Kommunikation (rot) beschrieben.
Die genauen Einstufungskriterien sind Abbildung 4.1 zu entnehmen.

Ampelsystem			
Rq	Rs	Bedeutung	Einstufungskriterien
		Unbedenklich	Klar erkennbarer Nutzen, keine bzw. kaum Profilierung möglich
		Wenig bedenklich	Klar erkennbarer Nutzen, Potential zur Profilierung gegeben (z.B. Cookies)
		Bedenklich	Aussage schwierig, möglicherweise nützlich, großes Potential zur Profilierung (z.B. wiederkehrende IDs)
		Sehr bedenklich	Vermutlich zum Zweck von Profilierung bzw. anderweitige ernste Gefahren

Tabelle 4.1: Ampelsystem für Request (Rq) und Response (Rs) mit Erläuterung

4.3 ERGEBNISSE

4.3.1 GRUPPIERTE TRACES

Abbildung 4.2 zeigt eine vereinfachte Liste von Traces mit Bewertung, Domain und Begründung der Bewertung. Eine genauere Auflistung mit zusätzlichen Informationen und Unterscheidungskriterien sowie einer feineren Gliederung der einzelnen Domains kann dem Anhang (Daten-DVD) entnommen werden.

Gruppierte Traces - Seite 1			
Rq	Rs	Domain	Begründung
		officeclient.microsoft.com	Nützlich, keine Profilierung möglich
		fs.microsoft.com	Nützlich, keine Profilierung möglich
		ctldl.windowsupdate.com	Nützlich, keine Profilierung möglich
		sls.update.microsoft.com	Request unauffällig, Response inhaltlich nicht verständlich, verdeckter Kanal möglich
		msftncsi.com	Nützlich, keine Profilierung möglich
		go.microsoft.com	Erkennbarer Nutzen, kann für Angriffe genutzt werden (Forwarding umlenken), kann Firewallregeln etc. umgehen durch flexibles Forwarding; unnötiges Senden von Informationen obwohl nur Forwarding-Request, ohne Verschlüsselung
		dmd.metaservices.microsoft.com	Request mit hardwarebezogenen IDs, Senden bei Änderung der angesteckten Hardware, Zweck möglicherweise in Kompatibilität und Versorgung mit passenden Treibern
		compatexchange1.trafficmanager.net	Nutzen (Stabilität) erkennbar, zwar durch Request recht genaue Profilierung möglich, jedoch Profilierung der Hardware und nicht des Nutzers bzw. Nutzerverhaltens
		microsoft.com	Nur Fowarding ohne persönliche Informationen, jedoch kann auch woandershin umgeleitet werden, Forwarding für nützlichen Dienst
		definitionupdates.microsoft.com	Nützlich, unauffällig, für Sicherheit nötig
		platform.bing.com	Nutzen (Bing) für Organisationen durchaus groß, Request und Response mit wiederkehrende IDs, teils dienstübergreifend
		bing.com	Nutzen unklar, sehr großes Potenzial zur Profilierung, recht genaue Charakterisierung des Systems, wiederkehrende IDs, erkennt Systembetrieb in Virtueller Maschine, tritt auch nach lokaler Suche auf
		setting.data.microsoft.com	Request unauffällig, kaum Profilierung möglich, Response schwer zu beurteilen
		telecommand.telemetry.microsoft.com	Nutzen nicht erkennbar, Request mit wiederkehrenden Cookies, IDs und Systemkonfiguration, Maschine eindeutig identifizierbar, Response mit XML-Datei ohne verwertbaren Inhalt
		sqm.telemetry.microsoft.com	Dient generell der Softwarequalität durch Sammeln von Nutzungsinformationen, Request und Response hier jedoch ohne relevante Informationen
		inprod.support.services.microsoft.com	Nützlich, vom Nutzer initiiert, Request ohne relevante Informationen, System nicht eindeutig identifizierbar
		login.live.com	Nützlich, teils vom Nutzer initiiert, Anforderung für Login nötiger Ressourcen
		auth.gfx.ms	Nützlich, Request und Response unauffällig, kaum Profilierung möglich
		ajax.aspnetcdn.com	Nützlich, Request und Response unauffällig, kaum Profilierung möglich
		c.s-microsoft.com	Nützlich, Request und Response unauffällig, kaum Profilierung möglich
		c.microsoft.com	Nützlich, vom Nutzer initiiert, Request und Response teils mit persistenten IDs, Profilierung möglich
		ieonline.microsoft.com	Nützlich, Request und Response unauffällig, kaum Profilierung möglich
		vortex.data.microsoft.com	Nutzen nicht erkennbar, Request in Versuchen sehr unterschiedlich, teils mit HTTP-History, teils keine relevanten Informationen, Response gibt Microsoft Zugriffsrechte auf zum Client übertragene Ressource Einstufung in einigen Versuchen "rot" da sehr bedenklich (HTTP-History), Einstufung in anderen Versuchen "grün" da unbedenklich (keine relevanten Informationen)

Gruppierte Traces - Seite 2			
Rq	Rs	Domain	Begründung
		api.bing.com	Nützlich, vom Nutzer indirekt initiiert, Request mit persistenten Cookies (über mehrere Sitzungen), Response unauffällig
		onedrive.live.com	Nützlich, vom Nutzer indirekt initiiert, Request mit wiederkehrenden Cookies (über mehrere Dienste), Response unauffällig
		sc.imp.live.com	Nützlich, vom Nutzer indirekt initiiert, Request mit wiederkehrenden Cookies (über mehrere Dienste), Response unauffällig
		p.sfx.ms	Request und Response unauffällig, kaum Profilierung möglich
		s1-officeapps-15.cdn.office.net	Request unauffällig, meist einmalige IDs, Response zweckdienlich
		s1-word-view-15.cdn.office.net	Request unauffällig, meist einmalige IDs, Response zweckdienlich
		s1-word-edit-15.cdn.office.net	Request unauffällig, meist einmalige IDs, Response zweckdienlich
		s1-excel-15.cdn.office.net	Request unauffällig, meist einmalige IDs, Response zweckdienlich
		s1-powerpoint-15.cdn.office.net	Request unauffällig, meist einmalige IDs, Response zweckdienlich
		s1-onenote-15.cdn.office.net	Request unauffällig, meist einmalige IDs, Response zweckdienlich
		iecvlist.microsoft.com	Nützlich, Request unauffällig, Response mit einmaligen IDs
		ocsp.verisign.com	Nützlich, unauffälliger Request und Response

Tabelle 4.2: Gruppierte Traces mit Bewertung, Domain und Begründung der Bewertung

4.3.2 KRITISCHE KOMMUNIKATIONEN

Die einzelnen Kommunikationen wurden aus datenschutztechnischer Sicht sehr unterschiedlich bewertet. Am kritischsten scheint die Kommunikation mit `telecommand.telemetry.microsoft.com` zu sein, da hier nicht nur persistente Cookies gespeichert werden sondern auch IDs und Systemkonfigurationen wiederkehrend auftauchen. Es ist davon auszugehen, dass über diesen Dienst die Maschine eindeutig identifiziert und charakterisiert werden kann. Von offizieller Seite gibt es zu diesem, wie zu vielen anderen Diensten, keine Informationen.

Ein weiterer Dienst, der sehr viele Daten über den Rechner preisgibt, ist `dmd.metaservices.microsoft.com`. Dieser übermittelt ungeschützt (auf Port 80) viele Hardwareinformationen wie Namen und Beschreibungen, Namen der Hersteller, Kategorien, Windows-Store-Geräte-Anwendungen, bevorzugte Anwendungen und viele weitere. Zudem werden auch Daten zum Drucker und zu verwendeten Sprachen übermittelt. Aufgefallen ist dieser Dienst auch, da er jedes Mal aktiv wird, wenn sich eine Hardwarekomponente ändert. Dies ist beispielsweise der Fall beim Anstecken eines Kopfhörers (egal ob USB oder Klinke), USB-Sticks oder anderer Hardware. Dieses Verhalten könnte zwecks Kompatibilität und Versorgung mit passenden Treibern durchaus sinnvoll sein, daher ist denkbar, dass die Verhinderung der Kommunikation in Langzeitversuchen zu Einschränkungen führt.

Der Dienst go.microsoft.com ist für Weiterleitungen zuständig. Hintergedanke ist vermutlich eine deutlich erhöhte Flexibilität von Microsoft-Diensten. Dies kann aber sehr einfach ausgenutzt werden, um Firewall-Regeln, wie das Blocken einer URL, zu umgehen. Die Kommunikation erfolgt unverschlüsselt, die Daten selbst sehen aber nicht datenschutzkritisch aus.

Bei dem Dienst bing.com handelt es sich unter anderem um die Suchmaschine von Microsoft. Die Kommunikation wird aber ebenfalls als kritisch eingeschätzt, da hier persistente Cookies und IDs über mehrere Kommunikationen auffindbar sind und somit die Maschine eindeutig identifiziert werden kann. Weiterhin gibt es einige Daten, deren Sinn nicht erkannt werden konnte. So ist z.B. unbekannt, was das Cortana-Manifest oder WindowsCortanaPane ist. Cortana selbst ist in der getesteten Windows-Version vollständig deaktiviert.

Der Dienst login.live.com/ppsecure/deviceaddcredential.srf scheint Teil vom Login-System von Microsoft zu sein. Auffällig war hier, dass diese spezielle Adresse nur von einem der getesteten Geräte beim Login angefragt wurde. Da der Login anscheinend auch ohne diesen Dienst funktioniert und die übertragenen Daten nur schwer oder gar nicht verständlich sind, wurde er ebenfalls als kritisch eingestuft.

Ein letzter Dienst, der sehr schwer einzuschätzen ist, ist vortex.data.microsoft.com. Auf den meisten Testgeräten verhielt sich dieser sehr unauffällig, jedoch wurden auf einem Rechner einmal Informationen gesendet, welche wie eine HTTP-History aussehen. Aufgrund dieses Vorkommnisses wurde auch dieser Dienst als kritisch eingestuft.

4.3.3 SONSTIGE AUFFÄLLIGKEITEN

Viele Übertragungen scheinen ineffizient zu sein. Es werden z.B. bei der Weiterleitung über go.microsoft.com schon beim Anfragen der Zieladresse alle Daten mitgeschickt, welche anschließend ein zweites Mal zum eigentlichen Ziel gesendet werden. Es fielen auch viel IDs auf, welche im Header mehrmals auftauchten, und Anfragen, die bereits eine Antwort bekommen hatten, wurden unmittelbar danach ein weiteres Mal gestellt. An diesen und anderen Sachen fiel eine nicht effiziente Ressourcennutzung auf, wobei Microsoft durch manches eventuell einen Nutzen ziehen könnte. Da es in dieser Untersuchung aber um Datenschutz geht, wurde auf diesen Aspekt nicht weiter eingegangen.

Wie im vorherigen Abschnitt bereits erwähnt, traten auch Kommunikationen auf, die nur auf einer Maschine zu finden waren und nicht reproduziert werden konnten. Die Untersuchung dieser zeigte leider nicht, ob es sich hierbei um Fehlerberichte oder ähnliches handelt. Der Fall mit vortex.data.microsoft.com ist aber der Einzige, der als Sicherheitskritisch eingestuft wurde.

5 ZUSAMMENFASSUNG UND AUSBLICK

5.1 ZUSAMMENFASSUNG

Im Rahmen des Komplexpraktikums „Datenschutzfreundliche Technologien im Internet" wurden die Datenschutz-Eigenschaften von Windows 10 Enterprise LTSB am Beispiel der ZIH-Version der Technischen Universität Dresden untersucht. Dabei wurde auch verschlüsselte Kommunikation betrachtet. Durch die in dieser Arbeit beschriebene Herangehensweise wurden verschiedene Gruppen von Übertragungen verzeichnet und bzgl. ihrer Datenschutzfreundlichkeit unterschiedlich eingestuft. Viele dieser Gruppen umfassen nachvollziehbare und berechtigte Nachrichten und Datentransfers, andere hingegen sind im jeweiligen Kontext nicht erwünscht oder nicht nachvollziehbar.
Verglichen mit den Ergebnissen aus Untersuchungen anderer Versionen von Windows 10 in früheren Arbeiten (vgl. Abschnitt 2.2) konnte für Enterprise LTSB erheblich weniger Kommunikation nachgewiesen werden. Dies bezieht besonders Datenübertragungen ein, welche in verwandten Arbeiten wiederholt als datenschutzunfreundlich eingestuft wurden.

Da im Fokus des Praktikums eine Beurteilung bzgl. Datenschutz steht, kann die Mehrheit der nachgewiesenen Kommunikation als vertretbar eingestuft werden. Schließlich wurden nur in wenigen Fällen personenbezogene Daten zur Identität oder zum Verhalten des Nutzers übertragen.
Dennoch legen die in Kapitel 4 vorgestellten Ergebnisse nahe, dass durch verschiedene Maßnahmen durchaus eine Steigerung des Datenschutzes erreicht werden kann. Hierzu zählt vor allem das Unterbinden einzelner Dienste bzw. der Kommunikation mit bestimmten Servern. Zur Auswahl dieser Dienste bzw. Server kann die in Abbildung 4.2 vorgestellte Liste von Kommunikationsgruppen mit der zugehörigen Einstufung gemäß dem Ampelsystem herangezogen werden. Ein möglicher Ansatz hierfür ist das Unterbinden jedweder mit orange oder rot bewerteter Kommunikation. Dabei sind jedoch mögliche Langzeitfolgen wie eine eingeschränkte Funktion der Bing-Suche und mangelhafte Kompatibilität von Hard- und Softwarekomponenten zu erwarten.
Aufgrund dieser Annahme empfehlen die Autoren der vorliegenden Arbeit einen differenzierteren Ansatz. Dieser beinhaltet die Unterbindung aller in Abschnitt 4.3.2 vorgestellter kritischer Kommunikationen abzüglich derer, die den Diensten `go.microsoft.com`, `login.live.com` oder `bing.com` zugeordnet werden können.

Dies umfasst:

- `telecommand.telemetry.microsoft.com`
- `dmd.metaservice.microsoft.com`
- `vortex.data.microsoft.com`

Darüber hinaus empfiehlt es sich, Update-Dienste wie `sls.update.microsoft.com` und `definitionupdates.microsoft.com` über ein internes Content Delivery Network (CDN) der Technischen Universität Dresden zu realisieren.

Abschließend sollen die Grenzen der Aussagekraft der vorgestellten Ergebnisse aufgezeigt werden.
Das vorliegende Nutzungsszenario beschränkt sich auf Windows 10 Enterprise LTSB, welches in einer virtuellen Maschine installiert wurde. Diese war im lokalen Netzwerk lediglich mit einem Router verbunden. Neben der Standard-Software der verwendeten Windows-Version wurden keine zusätzlichen Programme installiert. Zudem wurde in den Untersuchungen ein eher unnatürliches Nutzungsverhalten simuliert, bei dem eine kleine Anzahl von Programmen und Betriebssystemfunktionen über einen Zeitraum von wenigen Monaten Verwendung fand. Darüber hinaus unterstützt die Burp Suite lediglich die Protokolle HTTP und HTTPS. Kommunikation über andere Protokolle kann nicht untersucht werden. Zudem ist die Burp Suite als intrusives Werkzeug einzustufen, da sie aktiv an der Weiterleitung von Nachrichten beteiligt ist. Auch dies kann die Analyse beeinflussen.
Die genannten Ergebnisse und Erkenntnisse gelten somit genau für dieses Nutzungsszenario. Eine weiter reichende Aussagekraft ist nur durch zusätzliche Untersuchungen mit komplexeren Nutzungsszenarien möglich. Darüber hinaus sei anzumerken, dass das Nichtfinden bedenklicher Kommunikation nicht mit dem Nichtvorhandensein ebendieser gleichzusetzen ist.

Unter Berücksichtigung der genannten Anpassungen und Grenzen der Aussagekraft kann Windows 10 Enterprise LTSB für die Verwendung im Kontext der Technische Universität Dresden empfohlen werden. Dabei sei jedoch zu beachten, dass jede dieser Anpassungen ein Eingriff in Funktionen und Dienste von Microsoft oder dessen Partnern ist und Langzeitwirkungen haben kann.

5.2 AUSBLICK

In dieser Arbeit wurden viele Untersuchungen angestellt. Das Themenfeld bietet aber noch deutlich mehr Potential zur tiefgreifenden Erforschung.

Bisher wurden Tests nur auf vier Geräten vorgenommen, wovon drei in einer virtuellen Maschine liefen. Mit mehr Aufwand könnte man deutlich mehr physische Geräte testen, um der realen Situation an der Technischen Universität Dresden etwas näher zu kommen. Die Geräte sollten auch verschiedene Setups aufweisen, da schon sehr ähnliche Versuche zu teils sehr unterschiedlichen Ergebnissen führten und so ein besserer Überblick über die tatsächlich resultierenden Änderungen getroffen werden könnte. In diesem Zusammenhang sollten auch die Auswirkungen der Einstellungen in einem Langzeittest genau untersucht werden (vgl. Abschnitt 2.2). Des Weiteren sollten auch andere Windows-Editionen getestet werden, wie Education, da viele Studenten diese benutzen und auch in der Universität damit arbeiten.

Auch die Version Enterprise LTSB des ZIH scheint noch Optimierungspotential zu haben. Neben den bereits genannten Änderungen, könnte die „Feedbackhäufigkeit“ in den Datenschutzeinstellungen von „Automatisch“ auf „Nie“ gesetzt werden und die „Diagnose- und Nutzungsdaten“ von „Einfach“ auf „Sicherheit“. Dadurch sollte sich der Netzwerkverkehr weiter reduzieren. Mit den neuen Einstellungen könnten weitere Untersuchungen durchgeführt werden, um zu ermitteln, wie sich die Änderungen auf die Traces, deren Bewertung und die Funktionalität des Systems auswirken.

Im Internet finden sich auch einige Tools, die die Datensicherheit deutlich erhöhen sollen. Mehrheitlich wird von derartigen Tools abgeraten. In diesem Zusammenhang wäre es jedoch sinnvoll, solche Tools einmal genauer zu untersuchen.

Erwähnt werden sollte an dieser Stelle auch, dass Windows 7 und Windows 8 mittlerweile möglicherweise ähnliche Wege gehen. Die Quellen [Upd] und [Sch15] wiedersprechen sich aber in diesem Punkt. Der Vorwurf lautet, dass in den Systemen durch Sicherheitsupdates Dienste mit ähnlicher Funktionalität, was das Überwachungsverhalten angeht, nachinstalliert wurden. Reale Untersuchungen gibt es hierzu aber noch nicht.

LITERATUR

[Adm15] Administrator. *Analysis of Windows 10: In its principle, it is a mere terminal to collect information about the user's fingers, eyes and voice!* Tschechisch. Hrsg. von AENews. Übers. von Google Translate. 2015. URL: http://aeronet.cz/news/analyza-windows-10-ve-svem-principu-jde-o-pouhy-terminal-na-sber-informaci-o-uzivateli-jeho-prstech-ocich-a-hlasu/.

[Ant15] Sebastian Anthony. *Windows 10 doesn't offer much privacy by default: Here's how to fix it*. Hrsg. von ArsTechnica. 2015. URL: http://arstechnica.com/information-technology/2015/08/windows-10-doesnt-offer-much-privacy-by-default-heres-how-to-fix-it/.

[Bag15] Jo Bager. *Windows 10: Neue Datenschutzbestimmungen – Windows wird zur Datensammelstelle*. Hrsg. von heise online. 2015. URL: http://www.heise.de/newsticker/meldung/Windows-10-Neue-Datenschutzbestimmungen-Windows-wird-zur-Datensammelstelle-2765536.html.

[Ber15] Jürgen Berkemeier. *Table Sort*. 16. Nov. 2015. URL: http://www.j-berkemeier.de/TableSort.html.

[Bot16] Ed Bott. *Windows 10 telemetry secrets: Where, when, and why Microsoft collects your data*. Hrsg. von ZDNet. 2016. URL: http://www.zdnet.com/article/windows-10-telemetry-secrets/.

[Bri15] Peter Bright. *Even when told not to, Windows 10 just can't stop talking to Microsoft*. Hrsg. von ArsTechnica. 2015. URL: http://arstechnica.com/information-technology/2015/08/even-when-told-not-to-windows-10-just-cant-stop-talking-to-microsoft/.

[Bur] *Burp Suite*. URL: https://portswigger.net/burp/.

[DR08] T. Dierks und E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246. http://www.rfc-editor.org/rfc/rfc5246.txt. RFC Editor, 2008.

[Jä15] Heini Järvinen. *Microsoft's new small print - how your personal data is (ab)used*. Hrsg. von EDRi.org. 2015. URL: https://edri.org/microsofts-new-small-print-how-your-personal-data-abused/.

[Kal] *Kali Linux*. URL: https://www.kali.org.

[Kir12] Vadim Kiryukhin. *vkBeautify*. 23. Aug. 2012. URL: http://www.eslinstructor.net/vkbeautify/.

[Mer14] Chris Merriman. *Microsoft's Windows 10 Preview has permission to watch your every move*. Hrsg. von theINQUIRER. 2014. URL: http://www.theinquirer.net/inquirer/news/2373838/microsofts-windows-10-preview-has-permission-to-watch-your-every-move.

[Opp09] R. Oppliger. *SSL and TLS: Theory and Practice*. Artech House information security and privacy series. Artech House, 2009.

[Pro15] Tony Prophet. *Introducing Windows 10 Editions*. Hrsg. von blogs.windows.com. 2015. URL: https://blogs.windows.com/windowsexperience/2015/05/13/introducing-windows-10-editions/.

[RP15] Verbraucherzentrale Rheinland-Pfalz, Hrsg. *Windows 10 – Überwachung bis zum letzten Klick*. 2015. URL: https://www.verbraucherzentrale-rlp.de/windows-10---Ueberwachung-bis-zum-letzten-klick-1.

[Sag16] Ivan Sagalaev. *highlight.js*. 23. Feb. 2016. URL: https://highlightjs.org/.

[Sch15] Hajo Schulz. *Angebliche „Schnüffel-Updates" für Windows 7 und 8.1*. 27. Aug. 2015. URL: http://www.heise.de/newsticker/meldung/Angebliche-Schnueffel-Updates-fuer-Windows-7-und-8-1-2792343.html.

[Seb15] Andreas Sebayang. *Unternehmen verzichten mit LTSB zunächst auf den Edge-Browser*. Hrsg. von Golem.de. 2015. URL: http://www.golem.de/news/windows-10-unternehmen-verzichten-mit-ltsb-zunaechst-auf-den-edge-browser-1506-114572.html.

[TW12] Andrew S. Tanenbaum und David Wetherall. *Computernetzwerke*. 5., aktualisierte Aufl. Pearson, 2012.

[Upd] *Warnung vor Microsoft Patchday: Heute kommen die Schnüffel-Updates für Windows 7 und 8*. 8. Sep. 2015. URL: http://www.chip.de/news/Warnung-vor-Microsoft-Patchday-Heute-kommen-die-Schnueffel-Updates-fuer-Windows-7-und-8_83057903.html.

[Vbo] *Virtual Box*. URL: https://www.virtualbox.org/.

[W3C15] W3C. *File API*. 21. Apr. 2015. URL: https://www.w3.org/TR/file-upload/.

[Wir] *Wireshark*. URL: https://www.wireshark.org.