Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Björn Budde

Analyse und Bewertung der Update-Mechanismen gängiger Betriebssysteme

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Björn Budde

Analyse und Bewertung der Update-Mechanismen gängiger Betriebssysteme

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter: Prof. Dr. Thomas Schmidt

Eingereicht am: 28. September 2017

Björn Budde

Thema der Arbeit
Analyse und Bewertung der Update-Mechanismen gängiger Betriebssysteme

Stichworte
Updates, Upgrade, Patch, Verteilte Systeme, Computersicherheit, Software, Softwareverteilung, Softwarelebenszyklus, Betriebsysteme, Computer Netzwerk, Windows, Linux, OS X, Informationssicherheit, IT-Sicherheit, Cybersicherheit

Kurzzusammenfassung
Durch die wachsende Verbreitung und Komplexität von Desktop-Betriebssystemen steigen die Risiken und die Anzahl an Fehlern. Diesem muss im Rahmen eines Zyklus zur Verwaltung und Wartung von Software entgegen getreten werden. Hierbei kommt einem funktionierenden und sicheren Update-Mechanismus eine hohe Bedeutung zu. Für die Nutzer besteht dieser aus der Updateprüfung, Updateübertragung und Updateausführung. In der vorliegenden Arbeit wurde dazu ein Überblick erstellt und eine Sicherheitsanalyse durchgeführt, um die notwendigen Informationen zu Beurteilung und Bewertung der Betriebssystem-Update-Mechanismen zu erfassen.
Björn Budde

Title of the paper
Analysis and Evaluation of the Update Mechanisms of popular Operating Systems

Keywords
Updates, Upgrade, Patch, Distributed Systeme, Computersicherheit, Software, Softwaredistribution, Software Development Livecycle, Operating System, Computer Network, Windows, Linux, OS X, Informations Security, IT-Security, Cybersecurity

Abstract
The growing distribution and complexity of desktop-operating-systems leads to higher risks and more errors. These can be countered on the basis of a software maintenance and support cycle. In this, a functional and secure update mechanism is extremely important. For the users, this consists of the update check, update transfer and update execution. In this thesis, an overview was created and a security analysis was performed to gather the necessary information for evaluation and assessment of the operating system update mechanisms.

Inhaltsverzeichnis

1 Einleitung

Seit dem Aufkommen von Computern steigt deren Anzahl stetig an und die zunehmende Leistungs- und Speicherfähigkeit führen „zu einer Etablierung der Personal Computer [...] im privaten, universitären und unternehmerischen Umfeld “ (Baun, 2017, S.22). Heutzutage besetzen diese Personal Computer (PC) bzw. Desktop Computer den Markt zusammen mit Embedded-, Mobile-, IOT-Geräten und Workstations. Pro Quartal werden derzeit ungefähr 60 Millionen PC ausgeliefert (vgl. Gartner, 2017).

Der Personal Computer zeichnet sich durch eine Interaktion mit dem Nutzer aus. Die Nutzung des PCs erfolgt durch eine grafisch-interaktive Benutzeroberfläche, welche Teil des Desktop-Betriebssystems ist (vgl. Brause, 2017, S.352). Das Betriebssystem abstrahiert dafür die Hardware, zur Nutzung durch Software, welche wiederum dem Anwender Funktionen zur Verfügung stellt (vgl. Brause, 2017, S.3). Gestiegene Anwenderanforderungen an die Software führen zu neuen Betriebssystemfunktionen.

Nutzer fordern die Funktionalität von Betriebssystemen mit unterschiedlichen Sprachen, Bedienkonzepten und Zielsetzungen auf variierender Hardware. Für alle Nutzungsfälle ist ein stabiler Betrieb zu gewährleisten. Die Erfüllung dieser Anforderungen führt zu komplexer Betriebssystemsoftware. Die Anzahl der Codezeilen, die den Kern des Betriebssystems umfassen, ist daher mit der Zeit stark gestiegen. So wuchs die Zahl der Zeilen an Code im Windows NT Kernel zwischen 1993 und 2003 von 4-5 Millionen auf 50 Millionen um das Zehnfache (vgl. Maraia, 2005, Figure I.1). Diese Software ist jedoch nicht fehlerfrei.

Auch Betriebssysteme müssen daher innerhalb eines Softwarelebenszyklus gewartet und verwaltet werden. Gerade auch um eine langfristige Bindung der Nutzer zu gewährleisten ist im Rahmen der regelmäßigen Wartung die Aktualisierung, also das Update von Betriebssystemsoftware notwendig. Zwischenzeitliche Verbesserungen an Software werden vom Kunden aus verschiedenen Gründen gefordert. Zu den Wichtigsten zählen:

- **Sicherheit** - Schutz vor Angriffen durch die Behebung von Sicherheitslücken (vlg. Leopold u. a., 2015, S.64)
- **Stabilität** - Verbesserter Betrieb durch die Behebung von Fehlern (vgl. Wolfgang Osterhage, 2009, S.21)
- **Funktionalität** - Neue Funktionen und in Folge eine höhere Produktivität
- **Abhängigkeiten** - Anpassung an geänderte Abhängigkeiten wie z.B. neue Hardware
- **Compliance** - Geänderte Vorgaben, wie neue Gesetze und Richtlinien

1.1 Ziel

Die vorliegende Arbeit untersucht die Update-Mechanismen der drei größten Desktop- Betriebssysteme Windows, MacOS und Linux. Laut netmarketshare (2017) decken diese Betriebssysteme versionsübergreifend über 96% des Marktes ab. Es soll ein Verständnis für die Unterschiede und Gemeinsamkeiten der Update-Mechanismen entwickelt werden, um einen Überblick über die verwendeten Technologien zu geben. Die Update-Mechanismen sollen besonders auf ihre Sicherheit untersucht werden. Hierzu wird der Updateprozess in die Phasen Updateprüfung, Updateübertragung und Updateausführung geteilt. Ziel ist es, eventuelle Schwachstellen aufzudecken, um die Sicherheit der Nutzer zu erhöhen. Abschließend werden Vorschläge zur Verbesserung diskutiert.

1.2 Zielgruppe

Zum Verständnis der vorliegenden Arbeit wird auf Grundlagenwissen in Rechnernetzen und verteilten Systemen sowie der Informationssicherheit gesetzt. Im Bezug auf Rechnernetze setzt diese Arbeit Wissen über die Kommunikation im Internet und die beteiligten Systeme voraus. Im Besonderen sind hier das TCP/IP Protokoll, symmetrische und asymmetrische Verschlüsselung, SSL/TLS und das ISO/OSI-Schichtenmodell zu nennen. Weiter werden Grundkenntnisse in Betriebssystemen und Softwareentwicklung benötigt. Zudem sind Erfahrungen im Einsatz der untersuchten Betriebssysteme von Vorteil. Notwendige Begriffe werden im Kapitel *Grundlagen und Einführung* näher erläutert.

Zum Nachvollziehen der Ergebnisse werden Kenntnisse in der Auswertung von Netzwerkmitschnitten benötigt. Systemeigentümer können durch die gegebenen Schlüsse eine Bewertung

ihrer eigenen Systeme durchführen. Weiterhin kann die vorliegende Arbeit bei der Systembeschaffung einen Überblick über die Sicherheit der Update-Mechanismen liefern.

1.3 Voraussetzungen und Abgrenzungen

Die im Rahmen der vorliegenden Arbeit durchgeführte Untersuchung beschränkt sich auf die verbreitetsten Desktop-Betriebssysteme. Neben diesen finden sich Update-Mechanismen beispielsweise in IOT-, Mobile- und Server-Betriebssystemen. Sie sind nicht Teil dieser Untersuchung, auch wenn es große Schnittmengen zwischen den Systemtypen gibt und ähnliche Software eingesetzt wird. Ausgeschlossen von der Untersuchung ist auch weitere Software, welche auf den Systemen installiert ist und nicht zum Betriebssystem zählt bzw. vom Betriebssystem-Updateprozess berücksichtigt wird. Die vorliegende Arbeit untersucht den Update-Mechanismus auf der Seite des Clients. Dies beinhaltet nicht die internen Entwicklungsabläufe beim Updatehersteller.

1.4 Struktur der Thesis

Kapitel 2 stellt die Grundlagen zum Verständnis der folgenden Kapitel dar und definiert die Updatebegrifflichkeiten, Netzwerkarchitekturen, Sicherheits- und Angriffsbegriffe sowie Distributionsmethoden. Außerdem wird der zur Untersuchung durchgeführte Test erläutert.

In *Kapitel 3* wird Windows als Betriebssystem mit der größten Verbreitung untersucht.

Kapitel 4 zeigt mit MacOS den größter Mitbewerber von Windows im Markt der Desktop-Betriebssysteme.

In *Kapitel 5* wird Ubuntu Linux als einer der größten Vertreter von Open-Source Software als Betriebssystem betrachtet.

Kapitel 6 fasst die vorangegangenen Ergebnisse übergreifend zusammen, bewertet sie und gibt Lösungsmöglichkeiten.

Kapitel 7 schließt mit einer Ergebnisdiskussion und gibt einen Ausblick auf weitere Entwicklungsmöglichkeiten und Ziele.

Nicht jeder Nutzer ist für notwendige Updates empfänglich.

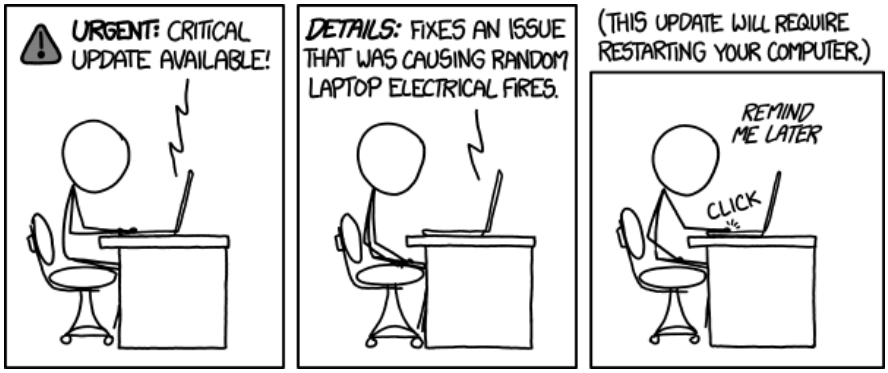

Abbildung 1.1: Randall Munroe, Update, 2014, https://xkcd.com/1328/

2 Grundlagen und Einführung

Betriebssysteme liefern dem Nutzer, wie in Kapitel 1 dieser Arbeit beschrieben, unterschiedliche Funktionalitäten, um Aufgaben zu erfüllen. Hierfür werden diverse Systemkomponenten benötigt. Diese Komponenten sind als Software zu bezeichnen und können eigenständige Funktionalitäten darstellen oder als Teil eines Softwaresystems wirken.

Wie jedes Produkt unterliegt auch die Betriebssystemsoftware einem Lebenszyklus. Um diesen darzustellen gibt es verschiedene Modelle. Beispielhaft sei der in Abbildung 2.1 gezeigte, in NIST (2008) beschriebene System Development Life Cycle (SDLC) genannt[1]. Dieser bietet eine gute Übersicht zur Einordnung der Bedeutung von Update-Mechanismen, ohne die Phasen im Lebenszyklus zu weit aufzugliedern.

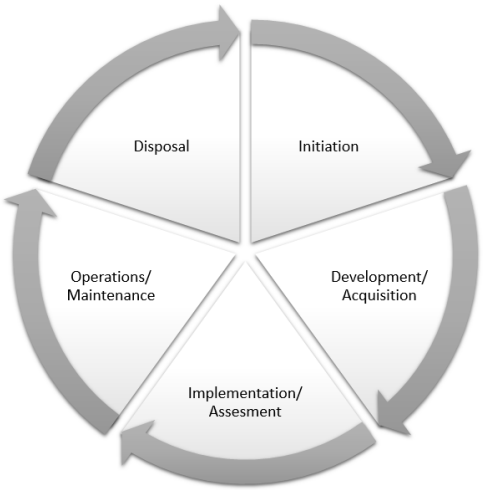

Abbildung 2.1: System Development Life Cycle

[1] Zur agilen Entwicklung passende Modelle sind unter: http://www.ambysoft.com/essays/agileLifecycle.html zu finden (letzter Zugriff: 20.09.17)

Der SDLC besteht aus 5 Phasen (vgl. NIST, 2008, S.6):

1. Initiation - Erstellung der Bedarfs- und Zieldefinition
2. Development/Acquisition - Systementwicklung oder Beschaffung
3. Implementation/Assessment - Systemtests und Einführung
4. Operations/Maintenance - Systembetrieb und Anpassung
5. Disposal - Aussteuerung des Systems

Die Wartungs- und Verwaltungsphase (Operations/Maintenance) nimmt in der zeitlichen Betrachtung den größten Teil des gezeigten Software- bzw. Betriebssystemlebenszyklus ein. Einem effektiven Update-Mechanismus kommt gerade in dieser Phase eine hohe Bedeutung zu. Er gewährleistet Sicherheit im fortlaufenden Betrieb und eine effiziente Verwaltung. Zudem kann auch in der letzten Phase der Aussonderung (Disposal) ein Update-System helfen, den Kunden zu einem neuen Softwareprodukt zu bewegen.

Aufgrund der zunehmenden Komplexität von informations- und datenverarbeitenden Systemen und der damit steigenden Angriffsfläche erhöht sich der Bedarf an Updates. Zudem steigen mit der Zeit seit Systemauslieferung auch meist die Anforderungen, um am Markt gegen Konkurrenzprodukte bestehen zu können. All dies führt zu einer erhöhten Anzahl an Updates und einer notwendigen Strukturierung und Verwaltung durch ein Update-System.

Nachfolgend werden wesentliche Grundlagen auf der zum Verständnis der Arbeit notwendigen Ebene erläutert. Durch die historische Entwicklung und einen heterogenen, durch Marketing beeinflussten Markt existieren unterschiedliche, sich überschneidende und teilweise widersprechende Terminologien. Diese Begriffe werden nachfolgend definiert und voneinander abgegrenzt.

2.1 Updates

Ein Update bezeichnet eine Informations- oder Datensammlung zur Zustandsüberführung eines Informationssystems von einem unerwünschten Ausgangszustand in einen Zielzustand. Die auf dem Betriebssystem vorliegenden Daten können in Systemdaten, welche die Software ausmachen, und Nutzerdaten, die verarbeitet werden, unterschieden werden. Ein Update verändert die Software, ist also eine Änderung der Systemdaten. Schon seit Beginn der Nutzung

von digitalen Systemen bestand die Notwendigkeit der Änderung von Systemdaten. Seit der Nutzung von Lochkartencomputern existiert zum Beispiel der Begriff des Patches, um Informationen kontrolliert zu verändern und Fehler zu beheben. Ursprünglich geschah dies durch manuelle Veränderung des Lochkartenmusters mit Aufklebern (engl. Patch) und Stanzen (vgl. Walkinshaw, 2017, S.73).

Die zur Zustandsüberführung notwendigen Informationen werden meist vom Softwarehersteller zusammengefasst und erreichen dann als Paket den Nutzer. Beim Hersteller sind bis zur Erstellung dieses Informationspaketes die in Abbildung 2.2 gezeigten Schritte zu durchlaufen.

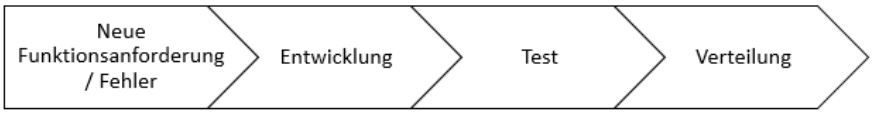

Abbildung 2.2: Update Ablauf

2.1.1 Updatebegriffe

Aktuell sind Begriffe wie Update, Patch, Upgrade, Bugfix und Hotfix, auf moderne Software bezogen, für ähnliche Verfahren im Gebrauch. Daher werden folgenden Definitionen empfohlen und gelten für die vorliegende Arbeit:

Update
Der Begriff Update bezeichnet eine Informationssammlung zur geplanten und kontrollierten Zustandsüberführung eines Informationssystems und dient als Sammelbegriff für die nachfolgenden Bezeichnungen. Ein Update bezeichnet zudem eine Menge an Fehlerkorrekturen, wenn diese aufgrund von gegenseitigen Abhängigkeiten gemeinsam ausgeführt werden. Ein Update kann verschiedene Systemkomponenten betreffen.

Upgrade
Ein Upgrade enthält neben Fehlerkorrekturen neue Funktionen und Features. Daher werden Upgrades teilweise auch als Feature Update bezeichnet. In der Vergangenheit erfolgte die Abgrenzung zwischen Update und Upgrade zusätzlich über die Kosten für ein Upgrade. Durch die geänderten Bezahlmodelle für Betriebssystemsoftware kann diese Definition heutzutage aber nicht mehr angewendet werden (vgl. Tate, 2013). Je nach Risiko und Schutzbedarf verzichten Nutzer teilweise auf Upgrades mit dem Ziel keine unbekannten Fehler in das System

einzuführen. Weiterhin können bei einem Upgrade auch Funktionalitäten entfernt werden (vgl. Hohmann, 2003, S.222).

Patch

Ein Patch wird zur Verbesserung der Sicherheit erstellt und eingespielt (vgl. Lenhard, 2017, S.29). Patches dienen der langfristig stabilen Behebung von Sicherheitslücken (vgl. Rohr, 2015, S.47) und werden intensiv getestet, um die Einführung neuer Sicherheitslücken und Fehler zu verhindern. Während die Installation von Upgrades nicht essentiell ist, sind Patches für einen sicheren Betrieb zwingend notwendig (vgl. IT-Grundschutz Bundesamt für Sicherheit in der Informationstechnik, 2016, M 2.273). Ein Patch ist in sich geschlossen und betrifft meist einen abgeschlossenen Systemteil. Teilweise werden die Begriffe Security Update und Patch synonym verwendet.

Bugfix

Bugfixes sind Patches, die der Fehlerbehebung dienen und auf Verbesserungen der Stabilität und Leistung im Betrieb abzielen. Generell zählen Bugfixes, wie Patches auch, zum normalen Herstellerservice.

Hotfix

Ein Hotfix ist ein auf einen spezifischen Fehler zugeschnittenes Update (vgl. Thomas M. Thomas and Donald Stoddard, 2012, S.97), welches im Falle eines Fehlers oder einer Sicherheitslücke hoher Kritikalität, also hohem Risiko, kurzfristig verteilt wird. Durch die hohe Kritikalität resultiert ein umgehender Umsetzungsbedarf von Gegenmaßnahmen. Das Risiko beim Systembetrieb mit kritischem Fehler oder Sicherheitslücke fällt höher aus, als das Einspielen eines weniger umfangreich getesteten Hotfix. Aufgrund der zeitlichen Anforderung nach schneller Verteilung werden Hotfixes eventuell auch über alternative Verteilungskanäle vertrieben und können von Dritten stammen.

Downgrade

Ein Downgrade beschreibt das Zurückführen des Systems auf einen ursprünglichen Systemzustand. Dies beinhaltet meist nicht die auf dem System vorliegenden persönlichen Daten (Nutzdaten). Gründe für ein Downgrade sind häufig mit einem Update eingeführte Fehler und Abhängigkeiten. Nicht alle Systeme unterstützen Downgrades, da durch diesen Vorgang auch ein unsicherer Ausgangszustand erreicht werden kann, wenn beispielsweise Änderungen an Nutzerdaten erfolgt sind, welche von früheren Versionen der Systemsoftware nicht unterstützt werden.

Kumulierte Updates

Kumulierte Updates sind aus einer Menge an Patches und Updates zusammengestellte Pakete (vgl. Bott u. a., 2017, S.532), welche durch die Bündelung Ressourcen schonen und den Transportaufwand verringern sollen. Ein bekanntes Beispiel für diese Art von Updates sind die Microsoft Service Packs (vgl. Microsoft, 2017a). Ziel ist es, dem Nutzer einen einfachen Zugriff auf die Ressourcen zu geben und die Sicherheit, beispielsweise durch die Updateausführung vor der ersten Verbindung mit dem Internet, zu erhöhen.

2.2 Verteilte Systeme

Um Anwendern in Rechnernetzen Funktionen anbieten zu können, werden häufig verteilte Systeme eingesetzt, da ein einzelnes System die Systemanforderungen nicht erfüllen kann. Besonders gilt dies für die Anforderungen nach Rechenleistung, Speicher und Ausfallsicherheit. Verteilte Systeme sind „eine Ansammlung unabhängiger Computer, die den Benutzern wie ein einzelnes kohärentes System erscheint“ (Tanenbaum und van Steen, 2008, S.19). Daher erkennt der Nutzer, wie auch bei einem Update-System, meist nicht, dass es sich um einen Systemverbund handelt. Verteilte Systeme stellen aktuell den Standard für Systeme der Größenordnung eines Update-Verteilungsmechanismus dar.

2.2.1 Netzwerkarchitekturen

Die Verfügbarkeit von Updates ist für die Clients Sicherheits- und Qualitätskriterium eines Betriebssystems zugleich. Grundlage einer funktionierenden Systemstruktur ist die Wahl einer passenden Netzwerkarchitektur. Grundsätzlich sind hier die Client-Server-Struktur und Peer-to-Peer Netzwerke zu unterscheiden.

Client-Server-Struktur

Wird die klassische Definition der Client-Server-Struktur zugrunde gelegt, so betrifft diese im Grundsatz eine durch den Client initiierte Verbindung an den Server, mit der ein bestimmter Service angefordert wird (vgl. Tanenbaum und van Steen, 2008, S.55). Die Verbindungen laufen zu einem Server, welcher den gesuchten Service bereitstellt. Um die Qualität des Services zu verbessern, läuft dieser aber nicht mehr auf einem Server, sondern wird von einem verteilten System bereit gestellt. Innerhalb dieses Systems werden die benötigten Informationen zum Beispiel von einer zentralen, verwaltenden Instanz verteilt. Die Knoten des verteilten Systems befinden sich in Clientnähe (Edge-Computing, Content-Delivery-Network) um die Netzlast zu verringern und zu verteilen (vgl. Tanenbaum und van Steen, 2008, S.615f). Zwischen den

Netzknoten und dem Endanwender besteht trotzdem eine Client-Server Beziehung, da die innere Struktur des verteilten Systems von außen nicht sichtbar ist.
Vorteil der Client-Server Struktur ist die einfachere Kontrolle des Systems an einer zentralen Stelle, welche von den Betriebssystemherstellern benötigt wird.
Nachteil ist die hohe Abhängigkeit von der fehlerfreien Funktion des zentralen Systems und die hohe Last an dieser Stelle.

Peer-to-Peer

P2P-Rechnernetze zeichnen sich durch die Kommunikation aller Teilnehmer untereinander aus. Der angebotene Service wird durch sich im Netzwerk befindlichen Netzwerkknoten angeboten. Hierzu wird eine spezifischen Anfrage, beispielsweise nach einer Information, im Netz gestellt. Die Netzknoten prüfen nun individuell, ob sie diese Anfrage erfüllen können. Bei erfolgreicher Antwort kann dann der Informationsaustausch zwischen den beteiligten Knoten durchgeführt werden.
Vorteil der P2P-Struktur ist die Sicherheit vor Ausfällen. Im Netz kann die Last auf die Knoten verteilt werden. Netzknoten können bis zu einer kritischen Menge dem Netzwerk bei- und austreten, ohne dass der Service ausfällt.
Nachteil ist die problematische Kontrolle des Netzwerkes, da die Knoten meist gleichberechtigt den angebotenen Service ausführen.

2.2.2 Distribution

Distributionsmethoden in Rechnernetzen unterscheiden sich in Push- und Pull-Verfahren. Diese werden nachfolgend erläutert.

Push

Im Push-Verteilungsverfahren informiert eine meist zentrale Instanz die Clients über verfügbare Updates. Je nach Implementation können Clients ohne eigene Wahlmöglichkeit geupdated werden. Da für diese Art der Umsetzung die zentrale Stelle die Informationen über alle verbundenen Clients halten muss, ist das Verfahren nicht überall einsetzbar (vgl. Dustdar u. a., 2013, S.237). Gerade in Firmennetzwerken finden sich Systeme, welche Updates verteilen und erzwingen, um keine unsicheren Systeme in Ihrem Netzwerk zuzulassen und die Verwaltung von einer zentralen Stelle zu ermöglichen.

Pull

Das Pull-Verfahren beschreibt die Kontaktaufnahme des Empfängers (Client) zum Sender

(Server) zur Überprüfung des Informationsstandes (vgl. Dustdar u. a., 2013, S.236). Auf den Update-Mechanismus bezogen ist dies die aktive Suche des Clients nach verfügbaren Updates. Hierfür ist vorab eine Stelle bekannt, bei der Updateinformationen abgefragt werden können. Anschließend entscheidet der Client über Umfang und Ausführung des Updates und führt die gewählte Aktion aus.

2.3 Sicherheit

Gesellschaftlich und politisch wird mit Sicherheit die „Abwesenheit von bzw. [der] Schutz vor Gefahren [...] assoziiert"(Glaeßner, 2002, S.3). Die Übertragung dieser Definition auf komplexe, digitale Prozesse und Systeme führt zu einem Bedarf, den Schutz und verbleibende Risiken zu verwalten.

Um im betrieblichen Umfeld Sicherheit zu gewährleisten, werden Informationssicherheits- und Risikomanagementsysteme (ISMS und RMS) eingesetzt. Diese dienen der Entwicklung, Umsetzung und Kontrolle verschiedener Vorgaben. Generell kann zwischen technischen Vorgaben und rechtlichen Vorgaben unterschieden werden. Technische Vorgaben werden im IT-Grundschutz (vgl. Bundesamt für Sicherheit in der Informationstechnik, 2016) definiert. Rechtliche Vorgaben und Umsetzungsempfehlungen ergeben sich aus der ISO/IEC (2014) Reihe 27000 und fortlaufende. Eine Besonderheit für viele Systeme stellt die Verarbeitung personenbezogener Daten dar. Für sie gelten weitere, z.B. in der Datenschutz-Grundverordnung (siehe Europäische Union, 2016) angegebenen Vorgaben.

Zuerst müssen die zu schützenden Werte oder Güter bekannt sein. Dies sind in digitalen Prozessen die Daten und Informationen. Den Werten sind Schutzziele zugeordnet. Sie unterliegen durch Bedrohungen und Schwachstellen einem Risiko. Für dieses Risiko muss nun eine Risikobeurteilung durchgeführt werden (vgl. Königs, 2017, S.52f). Die Beurteilung umfasst Höhe und Eintrittswahrscheinlichkeit eines Schadens. Danach können in der Risiko-Behandlung technische oder prozedurale Maßnahmen entwickelt und den Vorgaben entsprechend ausgewählt werden, um ein gewünschtes Sicherheitsniveau zu erreichen.

Viele Definitionen und Konventionen der Sicherheit entstammen dem angloamerikanischen Sprachraum. Dort ist die begriffliche Trennung der im Englischen als *Safety and Security* bezeichneten Funktionssicherheit und Informationssicherheit gängig (vgl. Eckert, 2014, S.6). Die Funktionssicherheit beschreibt die im Betrieb gewährleistete Sicherheit gegen Fehler. Die

Informationssicherheit beschreibt dagegen die Sicherheit vor Angriffen. Angriffe sind vorsätzliche Handlungen um sich „sich Vorteile zu verschaffen bzw. einen Dritten zu schädigen“ (vgl. IT-Grundschutz, Bundesamt für Sicherheit in der Informationstechnik, 2016, S.98).

2.3.1 Schutzziele

Im Rahmen von die Informationssicherheit betreffenden Systemuntersuchungen wird zur Beurteilung häufig auf das CIA-Schema zurückgegriffen(vgl. National Institute of Standards and Technology, 2004, S.2). Dieses Schema beschreibt Confidentiality, Integrity und Availability als Schutzziele für Informationen und Daten. Zusätzlich kann dieses Modell um weitere Schutzziele ergänzt werden (vgl. Kap.1.2 Eckert, 2014). In dieser Untersuchung wird zur Bewertung der Update-Mechanismen die Authentizität zusätzlich zu den drei klassischen Schutzziele im CIA-Schema betrachtet. Diese werden nachfolgend vorgestellt.

Vertraulichkeit/Confidentiality beschreibt die Eigenschaft der Informationen, nur für den gewünschten, limitierten Empfängerkreis erkennbar zu sein. Vertraulichkeit ist aus technischer Sicht zu unterscheiden zwischen dem Zustand der Übertragung und der Speicherung. Vertraulichkeit auf dem Übertragungsweg (*in transport*) umfasst die übertragenen Informationen selber und wird beispielsweise durch die Nutzung von SSL/TLS gewährleistet. Zudem kann die Verschlüsselung auch entstehenden Metadaten wie Ursprung, Ziel, Größe und Übertragungsdauer beinhalten. Auf die Speicherung (*at rest*) bezogen, kann die Vertraulichkeit zum Beispiel durch Dateiverschlüsselung mit Verfahren wie AES erreicht werden.

Integrität/Integrity bezeichnet die Unversehrtheit der vorliegenden Informationen. Dabei ist für die Sicherheitsbetrachtung besonders die Erkennung unberechtigt modifizierter Inhalte wichtig. Datenintegrität beschreibt umfassender die Erkennung jedweder unerlaubter Änderung der vorliegenden Daten. Die Integrität wird häufig durch die Nutzung von Hash-Verfahren zugesichert.

Verfügbarkeit/Availability steht für die Zugriffsmöglichkeit auf die gewünschten Informationen in einem definierten Zeitraum. Zudem beschreibt die Verfügbarkeit die Gewährleistung des Zugriffs ohne unautorisierte Beeinträchtigung (vgl. Eckert, 2014, S.12). Die Gewährleistung von lokalen Systemressourcen stellt aufgrund der Leistungsfähigkeit aktueller Systeme fast kein Problem mehr dar. Zur Verbesserung der Verfügbarkeit der Serverressourcen werden verteilte Systeme (siehe Abschnitt 2.2.1) eingesetzt.

Authentizität/Authenticity beschreibt den eindeutigen Beweis der den vorliegenden Informationen zugeordneten Identität. Diese beinhaltet zudem den Nachweis von Echtheit und Herkunft. Zum Beweis der Identität muss eine erfolgreiche Authentifizierung durchgeführt werden. Hierzu können die Faktoren Besitz (Public-Private-Keypair), Wissen (Passwort) oder biometrische Eigenschaft (Irismuster) genutzt werden. Für die Kommunikation zwischen Softwaresysteme erfolgt häufig der Nachweis von Besitz oder Wissen, da zum Beweis einer biometrischen Eigenschaft der Nutzer benötigt wird.

2.3.2 Angriffe

Angriffe beschreiben beabsichtigte, nicht autorisierte Handlungen, die das System gefährden (vgl. Sorge u. a., 2013, Kap.3.3). Nachfolgend werden sie aufgrund ihres Vorgehens unterschieden.

Passive Angriffe verletzen die Vertraulichkeit der Kommunikation, ohne die Integrität der Information zu gefährden. Hierzu zählen zum Beispiel das Abhören des Nachrichteninhaltes und die Gewinnung von der Verbindung zuzuordnenden Metadaten (Merkmale der Daten).

Aktive Angriffe verändern oder verfälschen die Informationen. Häufig haben aktive Angriffe eine Verletzung der Integrität zur Folge. Das Vortäuschen von Integrität ist also in der Konstruktion aktiver Angriffe von hoher Bedeutung. Zur Gruppe der aktiven Angriffe zählen beispielsweise die Veränderung, Wiederholung und Löschung von Daten.

2.4 Test

Um die Update-Mechanismen der Betriebssysteme vergleichen zu können, werden alle Tests in der gleichen Umgebung durchgeführt. Ziel ist die Aufzeichnung und Analyse des Betriebssystem-Updates. Hierfür wird der Netzwerkverkehr gespeichert. Updates von Software Dritter wurden in der Bewertung nicht berücksichtigt, sind aber zum Teil in den Netzwerkmitschnitten (Traces) zu finden.

2.4.1 Anforderungen

Als Anforderungen an die Update-Mechanismen gelten die Erfüllung der in Abschnitt 2.3.1 genannten Schutzziele Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität. Zur Untersuchung der Anforderungserfüllung wurde der Updateprozess wurde in die Phasen Updateprüfung, Updateübertragung und Updateausführung geteilt.

In der Phase der Updateprüfung wird erwartet, dass der Client sich Informationen zu den aktuell verfügbaren Updates beschafft, und ermittelt welche dieser Updates für den Client benötigt werden. Dies muss geschehen, ohne dass Rückschlüsse auf den aktuellen Updatestand des Clients erfolgen können.

Die Updateübertragung dient der sicheren Beschaffung von Updates. Hierbei ist zu beachten, dass die Updates aus sicheren Quellen stammen und wieder keine unerwünschten Informationen den Updatestand des Clients erkennbar machen. Aufgrund der Datenmenge ist auch die Qualität der Verfügbarkeit für die Beurteilung der Updateübertragung essentiell.

In der Phase der Updateausführung wird das Schutzziel der Integrität und Authentizität besonders betrachtet. Zu installierende Updates dürfen nicht verändert worden sein und müssen aus sicheren Quellen stammen.

2.4.2 Durchführung

Die Tests wurden in einem 2,4 GHz 802.11b/g WLAN Netzwerk durchgeführt. Der Anschluss ist mit 25 Mbit/s mit dem Internet verbunden. Die Standardeinstellungen des Systems (z.B. auch der DNS-Server) wurden beibehalten.

Der Netzwerkverkehr wurde über die Software Wireshark mitgeschnitten, gespeichert, einer ersten Begutachtung unterzogen und anschließend in Form von .pcap Dateien exportiert. Um die Aufzeichnung übersichtlich zu halten, wurden alle Mitschnitte gefiltert, damit nur den Host betreffende Packete gespeichert wurden.

2.4.3 Auswertung

Aufgrund der Datenmengen von mehreren hunderttausend Paketen wurden Analysetools zur Auswertung genutzt. Beispielhaft für die Analyse von .pcap-Dateien sind hier *pcapfex*, *EtherApe*, *rumint* und *tcpextract* zu nennen. Die Anwendung dieser Tools brachte jedoch aufgrund ihrer Beschränkungen nur wenige Erkenntnisse. Auch Cloud/Online-Lösungen wurden begutachtet. Hierzu zählt *PacketTotal*, welches die Ergebnisse der Untersuchungen grafisch ansprechend darstellt, aber eine Beschränkung auf 50MB Dateigröße und einer Einstimmung zur Veröffentlichung aller hoch geladenen Dateien vorraussetzt.

Die Auswertung erfolgte schließlich mit *tshark* und *Network Miner* Für weitere Tools zur Analyse wird an dieser Stelle auf das Wireshark-Wiki[2] und eine private Sammlung von PCAP-Tools[3] verwiesen. Außerdem wurde *Wireshark* für die Rekonstruktion der TCP-Streams, die Objekterkennung und den Export eingesetzt.

[2] https://wiki.wireshark.org/Tools
[3] Siehe http://xiaming.me/awesome-pcaptools/(letzterZugriff:20.09.17) (letzter Zugriff: 20.09.17)

3 Windows

3.1 Grundlagen

Windows 10 kommt mit einem Marktanteil von knapp 28% (netmarketshare, 2017) auf die größte Verbreitung der drei zu vergleichenden Betriebssysteme. Trotzdem ist aktuell noch Windows 7 mit ca. 48% das weltweit verbreitetste Betriebssystem für Desktop Computer. Ein möglicher Grund hierfür liegt in der Vergangenheit. Aufgrund des großen Erfolges von Windows XP und der Ankündigung von Microsoft, den Support für dieses Produkt zum 08. April 2014 einzustellen[1], sind viele Nutzer auf das zu dem damaligen Zeitpunkt stabilste Folgesystem Windows 7 umgestiegen. Windows 8 wurde durch die für die Nutzer ungewohnten Änderungen, wie der Touch-Orientierten-Bedienung, nicht wie gewünscht am Markt angenommen. Am 14. Januar 2020 endet der für Unternehmen wichtige erweiterte Support für Windows 7[2]. Spätestens zu diesem Zeitpunkt und durch Systemumstellungen im direkten Vorfeld ist mit einem starken Anstieg in der Verbreitung von Windows 10 zu rechnen.

Auch wenn die Sicherheitsanforderungen an ein System sich durch eine starke Verbreitung nicht ändern, so steigt doch das Risiko und der Schaden durch die hohe Anzahl an gefährdeten Instanzen. Weiterhin muss sich Microsoft, durch die hohe Anzahl an Clients, besonderen Infrastrukturproblemen stellen (Siehe Kap. 2.2.1).

Für die Installation und Nutzung von Windows 10 muss der Nutzer den Lizenzbedingungen zustimmen. Zu Updates wird folgendes genannt.
*„**6. Updates.** Die Software sucht in regelmäßigen Abständen nach System- und App-Updates, lädt diese für Sie herunter und installiert sie. [...] Microsoft muss möglicherweise Ihr System aktualisieren, um Ihnen diese Updates bereitstellen zu können. Durch die Annahme dieses Vertrags erklären Sie sich mit dem Erhalt dieser Arten von automatischen Updates ohne zusätzliche Be-*

[1] Offizielle Ankündigung mit Details: https://www.microsoft.com/en-us/windowsforbusiness/end-of-xp-support (letzter Zugriff: 12.09.17)
[2] Weitere Angaben: https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet (letzter Zugriff: 12.09.17)

nachrichtigung einverstanden.“ (vgl. Microsoft, 2017b)

Mit der Annahme der Bedingungen gibt der Nutzer weitreichende Rechte, die den Download und die weitere Installation ohne weitere Nutzerbestätigung betreffen, ab. Hierzu zählt besonders die Erlaubnis zum automatischen Installieren von Updates ohne Eingriff des Nutzers. Es folgt, dass Microsoft ohne Widerspruchsrecht des Nutzers, im Rahmen von Upgrades auch Features entfernen kann. Updates werden von Microsoft im Rahmen des Windows as a Service Servicemodelles angeboten.

3.1.1 Windows as a Service - WaaS

Windows as a Service (WaaS) ist die aktuelle Bezeichnung für das Servicemodell von Microsoft Windows 10. Die Nutzung des Betriebssystems wird als Dienstleistung angeboten und die Betriebssystemsoftware soll dafür kontinuierlich mit Updates versorgt werden und aktuell bleiben. Hierfür ist es notwendig, die Clients in einen regelmäßigen Updatezyklus zu bringen. Grundlage dazu ist die Vereinfachung der Verteilung und Anwendung sowie die Erhöhung der Akzeptanz der Nutzer für anstehende Updates. Bis zur Veröffentlichung von Windows 10 erschienen neue Windows Versionen in unregelmäßigen Abständen als sogenanntes Major Releases. Zwischen den Major Releases wurden Minor Releases veröffentlicht. Ältere Versionen verloren nach einiger Zeit dann den Status der unterstützen und mit Updates versorgten Produkte.

WaaS enthält die Absicht zur halbjährlichen Bereitstellung eines als Feature Updates bezeichneten Upgrades. Im Gegensatz zu einem Major Release mit sehr vielen neuen Features im Abstand von mehreren Jahren werden neue Funktionen in kleineren Schritten eingeführt. MS entfernt jedoch regelmäßig mit der Einführung eines Feature Updates Anwendungen und Funktionen. Üblich ist hierbei, dass dies vorab angekündigt wird[3]. Weiterhin werden auszusteuernde Bestandteile vorzeitig mit dem Status "Deprecated"versehen. Diese Teile sind zwar aktuell noch nutzbar, werden aber in zukünftigen Versionen entfernt oder ersetzt.

Weiterhin stellt Microsoft (MS) monatlich Qualitätsupdates bereit. Sie stellen zumeist eine Sammlung an Patches dar (Security Update). MS veröffentlicht Security Updates üblicherweise am zweiten Dienstag im Monat. Zudem werden weitere Informationen über die einzelnen Pat-

[3] Beispielsweise: https://support.microsoft.com/en-us/help/4034825/features-that-are-removed-or-deprecated-in-windows-10-fall-creators-up (letzter Zugriff: 12.09.17)

ches auf der Website von MS publiziert[4]. Diese Veröffentlichungen beinhalten Informationen, welches Teilsystem betroffen ist und ob Einschränkungen vorliegen. In einer Selbsteinschätzung werden die den Patches zu Grunde liegenden Sicherheitslücken nach Schweregrad (Severity) und Auswirkungen (Impact) bewertet. Durch die Veröffentlichung der Informationen, welche für Systemadministratoren teilweise essentiell sind, wird aber auch Angreifern und Erstellern von Schadsoftware die Arbeit erleichtert (vgl. Oliveria, 2006). Teilweise können durch diese Informationen und das Reverse Engineering von Updates bisher unbekannte Schwachstellen ausfindig gemacht werden. Die zu diesem Zeitpunkt noch nicht geupdateten Systeme sind alle potenziell für diese Schwachstelle anfällig. Die automatische Ausnutzung dieser grundlegenden System- bzw. Prozessschwachstelle wurden bereits untersucht (vgl. Brumley u. a., 2008).

3.1.2 Branches

Branches, auch Channel/Kanäle genannt, dienen der Unterteilung der Endanwender in unterschiedliche Nutzergruppen. Die Kanäle sind auf unterschiedliche Nutzeranforderungen ausgerichtet. Updates werden anhand ihrer Eigenschaften einem oder mehreren Kanälen zugeteilt und an die Clients ausgeliefert. Die Clients können sich für die Teilnahme an einem bestimmten Channel entscheiden. Eingeschränkt wird diese Wahl jedoch zum Teil von der Lizenz des Nutzers. Administratoren können für die von ihnen verwalteten Rechner Gruppenvorgaben treffen um die Zugehörigkeit zu einem Channel vorzugeben.

Zu Juli 2017 erfolgte eine Umstellung der Bezeichnungen und das Servicemodell enthält nun die nachfolgenden dargestellten Kanäle.[5].

Long-Term Servicing Channel

Der LTSC dient der Bereitstellung einer langfristig stabilen und sicheren Betriebsumgebung für Software. Dies ist beispielsweise für Hersteller von Finanzsoftware oder Systeme der Mess- und Medizintechnik aufgrund der hohen Anforderungen nach Verfügbarkeit bzw. Zuverlässigkeit wichtig. Der LTSC ist vom Bezug neuer Features im Rahmen des normalen Zyklus ausgenommen. Weiterhin fehlen den LTSC Systemen einige Systemkomponenten wie Cortana, Edge oder der Windows Store (vgl. Bott u. a., 2017, S.774). Das Beziehen von Updates aus dem LTSC ist nur mit Enterprise Lizenz möglich.

[4] Security Tech Center: https://portal.msrc.microsoft.com/en-us/security-guidance (letzter Zugriff: 14.09.17)
[5] Weitere Informationen unter https://blogs.technet.microsoft.com/windowsitpro/2017/07/27/waas-simplified-and-aligned/ (letzter Zugriff: 21.09.17)

Semi-Annual Channel

Bis Mitte 2017 war dieser Channel als Current Branch (CB) oder Current Branch for Business (CBB) bekannt. Der Semi-Annual Channel (SAC) unterteilt sich in die beiden Gruppen Targeted und Broad[6]. **Targeted** dient dem Test der fertigen Feature Updates in der Zielumgebung der Clients. In der Gruppe **Broad** erhalten schließlich der Großteil der Anwender, mit Ausnahme des Long-Term Servicing Channel, die Updates.

Windows Insider Preview

Im Insider Programm erhalten die Nutzer vor der Veröffentlichung in anderen Channels Zugriff auf Updates. Die Nutzer können so frühzeitig eventuelle Probleme im Betrieb dieser geupdateten Komponenten in ihrem System erkennen und an MS zurück spiegeln. Diese sollen dann vor einer Ausführung auf vielen Clients behoben werden. Falls nicht durch den Administrator untersagt, kann die Teilnahme am Windows Insider Programm vom Nutzer festgelegt werden.

Innerhalb dieses Channels sind weitere Abstufungsmöglichkeiten vorgesehen. **Release Preview** steht für einen früheren Bezug der im Semi-Annual Channel veröffentlichten Feature Updates. Die Gruppe **Slow** erhält zusätzlich zu den in Release Preview veröffentlichten Updates einige geringfügige Änderungen früher. **Fast** erhält vorliegende Software und Updates am schnellsten. Hierzu zählen viele Zwischenschritte hin zum nächsten Semi-Annual Channel Release.

Ein Wechsel zwischen den Channels ist nicht immer möglich. Grundsätzlich gilt: Ein Wechsel auf einen schnelleren Channel ist immer möglich, ein Wechsel auf einen langsameren Channel erfordert Warten auf das nächste Release (siehe Tabelle 3.1):

Channel	Insider Preview	Semi Annual	Long Term
Erscheinungshäufigkeit	Monatlich	6 Monate	2-3 Jahre
Servicefrist	-	18 Monate	10 Jahre

Tabelle 3.1: Microsoft Channel Übersicht

[6] Siehe: https://blogs.technet.microsoft.com/surface/2017/07/28/the-windows-semi-annual-channel-and-targeted-deployment/ (letzter Zugriff: 21.09.17)

3.2 Software

Windows 10 bringt ein komplexes, aus verschiedenen Komponenten bestehendes Updatesystem mit. Zusätzlich zu den Systemupdates verteilt MS darüber auch Updates, die eigene Softwareprodukte betreffen (bspw. MS Edge). Im Gegensatz zu früheren Windows Versionen werden die bisher verfügbaren Updates für ein Produkt im Regelfall kumuliert (vgl. Bott u. a., 2017, S.532). Nachfolgend werden die wichtigsten Komponenten des Windows-Update-Mechanismus beschrieben.

3.2.1 Windows Update Service

Unter MS Windows läuft ein Update Service zur Verwaltung der Updates. Dieser, als wuaserv firmierende Windows Update Service ist regelmäßig aktiv und sucht nach neu verfügbaren Updates (Pull-Verfahren). Liegen Updates vor, so werden diese im Normalfall automatisch vom Client heruntergeladen. Alternativ können Updates aber auch manuell heruntergeladen werden. Neben der Prüfung und Beschaffung von Updates initiiert der Windows Update Service auch die Installation der Updatepakete.

3.2.2 Windows Server Update Service - WSUS

WSUS ist eine von MS angebotene Lösung zur Verwaltung und Verteilung von Software, Updates und Einstellungen auf zugewiesenen Computern. Hierfür stellt WSUS eine zentrale Server-Komponente dar. Mit WSUS können Updates gehostet werden, damit der Download aus dem Internet für die verbundenen Rechner nur einmal erfolgen muss. Dadurch kann eine dem Client-Server-Modell entsprechende, einfach zu verwaltende, lokale Updatearchitektur geschaffen werden[7]. Zudem kann mit WSUS auch signierte Software Dritter verteilt werden.

3.2.3 Windows Update Delivery Optimization - WUDO

WUDO ist eine mit Windows 10 eingeführte P2P-Funktionalität (siehe Absatz 2.2.1) zur Verteilung von Updates und ergänzt die klassischen Client-Server-Verteilstruktur. Historisch gesehen steht es in der Tradition von *BranchCache*, welches schon seit Windows 7 eine ähnliche Funktion in Zusammenarbeit mit WSUS anbietet und weiterhin existiert[8]. Während für diese Lösung

[7] Genaue Erläuterung zum Einsatz unter https://technet.microsoft.com/en-us/library/cc708448(v=ws.10).aspx (letzter Zugriff: 20.09.17)

[8] Dokumentation unter: https://docs.microsoft.com/en-us/windows/deployment/update/waas-branchcache (letzter Zugriff: 20.09.17)

aber aus Anwendersicht weitere Software benötigt wird, funktioniert WUDO in Zusammenarbeit mit einem MS zugehörigen verteilten System. Dieses verwaltet alle bekannten Netzknoten und liefert dem Client die zur Übertragung des Updates notwendigen Informationen (vgl. Team, 2017)

Um das Verhalten von WUDO den vorliegenden Anforderungen anzupassen existieren verschiedene Optionen[9]. Hierzu zählen zum Beispiel die Cache Größe und Dateiquellen (LAN, Group, Internet...). Wird WUDO deaktiviert, so kann diese Instanz nicht mehr als Cache dienen.

Laut MS dient WUDO der Verteilung von Teilpaketen der Updates. Die zusammengesetzten Updates werden unabhängig von Ihrer Herkunft (offizieller Server oder lokaler Cache) auf Authentizität geprüft[10].

3.3 Test

3.3.1 Testablauf

Der Test wurde auf einem Rechner mit Windows 10 Pro Lizenz in der Version 1607 durchgeführt. Der Rechner gehört zum allgemeinen, öffentlichen Semi-Annual Channel (siehe Abschnitt 3.1.2). Für den Test wurden Wireshark und USBPCap installiert. Mit Wireshark wurde ein Netzwerkmitschnitt eines Updatevorganges durchgeführt.

Ausgelöst wurde die Updateprüfung über *Update und Sicherheit* in den Microsoft Systemeinstellungen. Nach der Suche der Updates erfolgte den Updatevorgang betreffend keine weitere Nutzernachfrage oder Eingabe. Die in Auflistung 3.1 gezeigten Updates wurden heruntergeladen und installiert.

```
1 - Definitionsupdate fuer Windows Defender -KB22676 (Definition 1.251.421.0)
2 - Update fuer Windows 10 Version 1607 fuer x64-basierte Systeme (KB4033637)
3 - Update fuer Windows 10 Version 1607 fuer x64-basierte Systeme (KB4023057)
4 - Windows-Tool zum Entfernen boesartiger Software fuer Windows 8, 8.1, 10 und
5               Windows Server 2012, 2012 R2, 2016 x64 Edition -August 2017 (KB890830)
6 - 2017-08 Kumulatives Update fuer Windows 10 Version 1607 fuer x64-basierte Systeme
7               (KB4034658)
8 - 2017-08 Sicherheitsupdate fuer Adobe Flash Player fuer Windows 10 Version 1607
9               fuer x64-basierte Systeme (KB4034662)
```

[9] Dokumentation unter: https://docs.microsoft.com/en-us/windows/deployment/update/waas-delivery-optimization (letzter Zugriff: 21.09.17)

[10] https://privacy.microsoft.com/de-DE/windows-10-windows-update-delivery-optimization

```
10 - 2017-08 Update fuer Windows 10 Version 1607 fuer x64-basierte Systeme (KB44035631)
11 - Windows 10 Creators Update-Datenschutzeinstellungen fuer x64-basierte Systeme
12                 (KB4013214)
13 - Microsoft .NET Framework 4.7 fuer Windows 10 Version 1607 und Windows Server 2016
14                 fuer x64 (KB3186568)
```

Auflistung 3.1: Verfügbare Windows Updates

3.3.2 Analyse und Auswertung

Der Mitschnitt umfasst ca. 1,9 GB in Form von knapp unter 1,87 Millionen aufgezeichneten Pakete. Der Updatevorgang dauerte etwas mehr als 2 Stunden. Die Windows Updates werden auf der Festplatte unter *<Windows Directory>\softwaredistribution\download* temporär gespeichert. Die abgelegten Updatepakete wurden auch im Netzwerkmitschnitt erkannt.

Übersicht

Im Netzwerkmitschnitt konnten unterschiedliche Dateitypen ermittelt werden. Als Wichtigste sind hier folgende zu nennen. **Appx**/AppxBundle-Dateien sind Universal Windows Platform Apps, welche auf Portierbarkeit ausgerichtet sind, signiert werden und als primäres Dateiformat im Windows Store zum Einsatz kommen. Weiterhin Dateien im **CAB**-Format, das Microsoft Cabinet Format um Pakete zu erstellen und zu komprimieren. **CER**-Dateien sind Zertifikate welche zum Code signieren, verifizieren oder zur Verschlüsselung mit SSL/TLS genutzt werden. Weiterhing waren **EXE** (executable), also ausführbare Dateien und **MSP**/(Microsoftpatch) Dateien zum Update von Softwarepaketen im Mitschnitt zu erkennen.

Die folgenden Hosts lieferten im Rahmen des Netzwerkmitschnittes die meisten Pakete an den Client (siehe Tabelle 3.2). Insgesamt gab es aktive Verbindungen zu 136 Hosts.

IP	Paketanzahl	Name
88.134.181.98	319397	download.windowsupdate.com, download.windowsupdate.com.edgesuite.net
13.107.4.50	267415	7.tlu.dl.delivery.mp.microsoft.com, b1.download.windowsupdate.com
88.134.181.107	222484	download.windowsupdate.com, download.windowsupdate.com.edgesuite.net

Tabelle 3.2: Windows Update Service Server

Während die Domain windowsupdate.com durch eine WHOIS-Abfrage Microsoft zuzuordnen ist[11], kann edgesuite.net der Firma Akamai zugeordnet werden[12]. Eindeutig ist hier zu sehen, dass Microsoft für die Verteilung seiner Updates neben der eigenen Infrastruktur ein Content-Delivery-Network (CDN) einsetzt. Die aufgeführten IP-Adressen betreffend kann festgestellt werden, dass sowohl die 88.134.181.98 als auch die 88.134.181.107 beim Anschlussprovider (Kabel Deutschland) zu finden sind. Die Server des Akamai CDN befinden sich also in Anwendernähe. Die 13.107.4.50 wird in Redmont verortet und gehört zu Microsoft.

Netzwerkverbindung

Bei bestehender Netzwerkverbindung testet Windows 10 die Interneterreichbarkeit durch den Aufruf von http://www.msftconnecttest.com/connecttest.txt (siehe Tabelle 3.3). Dem Server gegenüber identifiziert sich der anfragende Dienst durch den User-Agent Microsoft NCSI (Network Connectivity Status Icon)[13].

Paket Nr.	Quelle	Ziel	Protokoll	Kommentar
18	192.168.0.42	192.168.0.1	DNS	Query
20,23,24	192.168.0.42	13.107.4.52	TCP	Handshake
25,26	192.168.0.42	13.107.4.52	HTTP	GET-Request+Response
27-30	192.168.0.42	13.107.4.52	TCP	Close

Tabelle 3.3: Frames Erreichbarkeitsprüfung

Updateprüfung

Vor dem Updatedownload werden verschiedene SSL/TLS Verbindungen aufgebaut. Auflistung 3.2 zeigt die im Netzwerkmitschnitt beim SSL/TLS-Handshake in den Zertifikaten erkennbaren Servernamen nach Häufigkeit sortiert. Die am häufigsten auftretenden *.do.dsp.mp.microsoft.com Domains werden für „peer discovery and peer management“(vgl. Team, 2017) des WUDO Service genutzt. Nach einem Microsoft Technet Eintrag[14] können die *.update.microsoft.com Domains MS und spezifisch dem Update Service zugeordnet werden. *.akamized.net-Domains

[11] Weitere Domains können den Microsoft Firewall Konfigurationsempfehlungen entnommen werden: https://technet.microsoft.com/en-us/library/bb693717.aspx (letzter Zugriff: 21.09.17)

[12] Weitere Details in der Entwicklerbeschreibung von Akamai: https://developer.akamai.com/learn/Overview/Client_Edge_Servers_Origin.html (letzter Zugriff: 21.09.17)

[13] Weitere Informationen unter https://blogs.technet.microsoft.com/networking/2012/12/20/the-network-connection-status-icon/ (letzter Zugriff: 21.09.17)

[14] Siehe im Detail https://technet.microsoft.com/en-us/library/bb693717.aspx

sind Teil des Akamai Content-Delivery-Networks. Welcher der gezeigten Verbindungen die Funktionalität der Updateprüfung darstellt, wurde aufgrund der SSL/TLS Verschlüsselung an dieser Stelle nicht weiter untersucht, da für eine Untersuchung die Kenntnis des Kommunikationsinhalt grundlegend wäre.

```
$ tshark -r win_update.pcap -T fields -e ssl.handshake.extensions_server_name \
-Y ssl.handshake.extensions_server_name | sort | uniq -c | sort -rg
158 cp401-prod.do.dsp.mp.microsoft.com
 73 login.live.com
 44 geover-prod.do.dsp.mp.microsoft.com
 41 disc401-prod.do.dsp.mp.microsoft.com
 39 fe3.delivery.mp.microsoft.com
 32 tsfe.trafficshaping.dsp.mp.microsoft.com
 18 v10.vortex-win.data.microsoft.com
 11 sls.update.microsoft.com
 11 fe2.update.microsoft.com
  9 settings-win.data.microsoft.com
  9 img-prod-cms-rt-microsoft-com.akamaized.net
  8 store-images.s-microsoft.com
[...]
```

Auflistung 3.2: Gekürzte Analyse Windows SSL/TLS Server

Beispiel 1 - Microsoft Malicious Software Removal Tool

Beginnend bei Paket Nr. 357 412 im Windows Update Netzwerkmitschnitt wurde ein Update von *windowsupdate.com* (88.134.181.98) geladen. Wie in Auflistung 3.3 erkennbar ist User-Agent Feld als *Microsoft-Delivery-Optimization/10.0* gesetzt. Dies deutet darauf hin, dass die Kommunikation vom WUDO-Service (Siehe 3.2.3) initiert wurde. Das Update wurde unverschlüsselt geladen, die Kommunikation erfolgte über den Server Port 80. Der Server gab sich in der Antwort als *Microsoft-IIS/8.5* aus. Dies ist schlüssig, da Internet Information Services (IIS) das Microsoft Web Server Produkt darstellt.

```
Frame 357412: 400 bytes on wire (3200 bits), 400 bytes captured (3200 bits)
Internet Protocol Version 4, Src: 192.168.0.42, Dst: 88.134.181.98
Transmission Control Protocol, Src Port: 50713, Dst Port: 80, \
Seq: 1, Ack: 1, Len: 346
Hypertext Transfer Protocol
GET /c/msdownload/update/software/uprl/2017/08/windows-kb890830-x64-v5.51_[...].exe
Cache-Control: no-cache
Connection: Keep-Alive
Accept: */*
User-Agent: Microsoft-Delivery-Optimization/10.0
```

```
11 Host: download.windowsupdate.com
```

Auflistung 3.3: Gekürzter GET-Request aus Windows Netzwerkmitschnitt TCP-Stream 1018

Bei dem Update handelt es sich nach der Dateibeschreibung um das Microsoft Malicious Software Removal Tool in der Version 5.51[15]. Das Tool enthält zwei Signaturen (SHA-1 und SHA-256). Beide sind auf die Microsoft Corporation ausgestellt, von der Microsoft Code Signing PCA signiert und am 27. Juli 2017 ausgestellt. Der Zertifizierungspfad lässt sich mit dem Microsoft SignTool verifizieren und bis zur Microsoft Root Certificate Authority nachvollziehen (siehe Auflistung 3.4).

```
1  SignTool verify /pa /v D:windows-kb890830-x64-v5.51_[...].exe
2  Verifying: D:windows-kb890830-x64-v5.51_[...].exe
3
4  Signature Index: 0 (Primary Signature)
5  Hash of file (sha1): 00DAE6E7723763C6C06D052DF079CEAA4F26C0AF
6
7  Signing Certificate Chain:
8      Issued to: Microsoft Root Certificate Authority
9      Issued by: Microsoft Root Certificate Authority
10     Expires:   Mon May 10 01:28:13 2021
11     SHA1 hash: CDD4EEAE6000AC7F40C3802C171E30148030C072
12         Issued to: Microsoft Code Signing PCA
13         Issued by: Microsoft Root Certificate Authority
14         Expires:   Tue Sep 01 00:29:32 2020
15         SHA1 hash: 3CAF9BA2DB5570CAF76942FF99101B993888E257
16             Issued to: Microsoft Corporation
17             Issued by: Microsoft Code Signing PCA
18             Expires:   Thu Nov 02 22:17:17 2017
19             SHA1 hash: 98ED99A67886D020C564923B7DF25E9AC019DF26
20 [...]
21 Successfully verified: D:windows-kb890830-x64-v5.51_[...].exe
22 Number of files successfully Verified: 1
```

Auflistung 3.4: Signaturüberprüfung Beispiel 1

Beispiel 2 - Update and Privacy Experience

Paket 401 608 leitet mit einem HTTP-GET-Request den Download einer .cab-Datei (Siehe 3.3.2) vom Host *b1.download.windowsupdate.com* (13.107.4.50) ein (siehe Auflistung 3.5). Das Update behandelt die *Update and Privacy Experience (UPX).*

[15] Weitere Informationen unter https://support.microsoft.com/en-us/help/890830 (letzter Zugriff: 21.09.17)

```
Frame 401608: 366 bytes on wire (2928 bits), 366 bytes captured (2928 bits)
Internet Protocol Version 4, Src: 192.168.0.42, Dst: 13.107.4.50\\
Transmission Control Protocol, Src Port: 50807, Dst Port: 80, \
Seq: 291, Ack: 553, Len: 312\\
Hypertext Transfer Protocol
GET /d/crup/2017/05/windows10.0-kb4013214\-x64\_426d[...]b244.cab HTTP/1.1
Cache-Control: no-cache\\
Connection: Keep-Alive\\
Accept: */*\\
User-Agent: Microsoft-Delivery-Optimization/10.0\\
Host: b1.download.windowsupdate.com\\
```

Auflistung 3.5: Gekürzter GET-Request aus Windows Netzwerkmitschnitt TCP-Stream 1112

Die Datei enthält durch Microsoft nicht näher spezifiziert Änderungen der UPX in Vorbereitung auf ein folgendes Feature Update[16]. Das Update ist mit einem Microsoft Windows Zertifikat signiert. Das Zertifikat lässt sich verifizieren, auf die Microsoft Root Certificate Authority zurückführen und nutzt SHA-1 (siehe Auflistung 3.6).

```
SignTool verify /pa /v D:windows10.0-kb4013214-x64_[...].cab
Verifying: D:windows10.0-kb4013214-x64_[...].cab

Signature Index: 0 (Primary Signature)
Hash of file (sha1): 738DBE509AAF091BF08D10B502E90F5B141C7402

Signing Certificate Chain:
    Issued to: Microsoft Root Certificate Authority
    Issued by: Microsoft Root Certificate Authority
    Expires:   Mon May 10 01:28:13 2021
    SHA1 hash: CDD4EEAE6000AC7F40C3802C171E30148030C072
        Issued to: Microsoft Code Signing PCA
        Issued by: Microsoft Root Certificate Authority
        Expires:   Tue Sep 01 00:29:32 2020
        SHA1 hash: 3CAF9BA2DB5570CAF76942FF99101B993888E257
            Issued to: Microsoft Corporation
            Issued by: Microsoft Code Signing PCA
            Expires:   Thu Nov 02 22:17:17 2017
            SHA1 hash: 98ED99A67886D020C564923B7DF25E9AC019DF26
[...]
Successfully verified: D:windows10.0-kb4013214-x64_[...].cab
Number of files successfully Verified: 1
```

Auflistung 3.6: Signaturüberprüfung Beispiel 2

[16] Veröffentlichungsinformationen unter https://support.microsoft.com/en-us/help/4013214 (letzter Zugriff: 21.09.17)

3.3.3 Bewertung

Der Update-Mechanismus von MS Windows 10 ist aktuell als größtenteils sicher zu bewerten. Problematisch ist jedoch, dass durch die Übertragung der Updates im Klartext Informationen über Updatestand ableitbar sind.

Updateprüfung

Die Prüfung auf neue Updates findet über eine SSL/TLS-Verbindung verschlüsselt statt. Dies entspricht der Sicherheitsanforderung nach Vertraulichkeit. Auffällig ist, dass Verbindungen zu sehr vielen Servern erfolgen, welche aufgrund ihrer Hostnamen (login.live.com, *.update.microsoft.com, settings-win.data.microsoft.com) den Schluss zulassen, dass unterschiedliche Services angeboten werden (siehe Auflistung 3.2).

Updateübertragung

In den zwei Beispielen kann erkannt werden welche spezifischen Updates heruntergeladen wurden, da die Übertragung der Updates unverschlüsselt und damit nicht vertraulich durchgeführt wird. Aus der Beschaffung dieses Updates lässt sich darauf schließen, dass diese aktuell noch nicht auf dem Rechner vorhanden sind. Die Windows Updates sind mit einer Knowledge Base (KB) Nummer im Dateinamen gekennzeichnet. Auch wenn die Identitätsfeststellung von Dateien über verschiedene andere Verfahren, wie ein Vergleich von Hash-Werten, möglich ist, erleichtert die KB-Nummer diesen Vorgang deutlich. Zur Feststellung des Updatestandes eines Clients müssen nur die HTTP-GET-Requests mitgeschnitten und auf heruntergeladene Updates gefiltert werden.

Updateausführung

Die heruntergeladenen Dateien werden vor ihrer Ausführung auf gültige Signaturen überprüft und sind neben einem SHA-256, auch mit einem SHA-1 Zertifikat signiert, welches zur Verifikation genutzt wird und gültig ist (siehe Auflistung 3.4 und 3.6). Das SHA-1 Hash-Verfahren ist spätestens nach der Veröffentlichung der ersten berechneten Kollisionen (vgl. Stevens u. a., 2017), nicht mehr als sicher anzusehen, auch wenn für einen Angriff auf ein Zertifikat ein anderer Angriff (Second Preimage Attack) erfolgreich durchgeführt werden muss (vgl. Rechberger, 2010). Aktuell erfüllt dieses Vorgehen daher noch die Schutzziele der Integrität und Authentizität.

4 MacOS

4.1 Grundlagen

MacOS (macOS, OS X) wurde in der ersten Version im Jahre 2001 von der Firma Apple eingeführt[1]. Im Frühjahr 2009 hat macOS Linux als Hauptkonkurrent von Windows im Bereich Desktop-Betriebssysteme abgelöst[2]. Im Gegensatz zu Microsoft stellt Apple in großen Absatzmengen Hardware wie Notebooks und Mobilgeräte her. MacOS läuft auf den Desktop PCs, Workstations und Notebooks. Die von Apple hergestellten Tablets und Smartphones werden mit dem Mobile-Betriebssystem iOS, welches im Unterschied zu MacOS eine auf Touch-Interaktion ausgerichtete Bedienoberfläche nutzt, ausgestattet.

Apples Geschäftskonzept unterscheidet sich grundlegend von dem anderer Mitbewerber wie Microsoft. MacOS ist ausschließlich auf der von Apple vorgesehen Hardware lauffähig. Vorteil dieser engen Verbindung von Software und Hardware ist die vereinfachte Verwaltung und Wartung. Nachteile sind die fehlende Flexibilität und Abhängigkeit der Anwender von Apple.

Über den Websupport von Apple können weitere Informationen zu Sicherheitsupdates beschafft werden[3]. Auch neue Softwareversionen von MacOS werden von Apple online angekündigt.

4.1.1 Releases

MacOS erscheint in unregelmäßigen Abständen[4]. Seit 2011 wurden aber jährlich neue Feature Updates, auch Major Versions genannt, veröffentlicht. Weiterhin erscheinen ungefähr alle 2-3 Monate kumulierte Updates, von Apple als Minor Versions bezeichnet. Nutzer können sich für das Beta-Test Programm von Apple anmelden, um Updates früher im Rahmen einer öffentlichen Testphase zu erhalten.

[1] Siehe: https://web.archive.org/web/20010410204117/http://www.apple.com:80/macosx/ (letzter Zugriff: 21.09.17)
[2] Siehe: https://en.wikipedia.org/wiki/Usage_share_of_operating_systems (letzter Zugriff: 21.09.17)
[3] Siehe https://support.apple.com/de-de/HT201222 (letzter Zugriff 21.09.17)
[4] Weitere Informationen unter https://robservatory.com/a-useless-analysis-of-os-x-release-dates/ (letzter Zugriff: 21.09.17)

4.2 Software

Apple nutzt in MacOS in der Standardeinstellung den Mac App Store zur Prüfung, Verteilung und Installation von Updates[5]. Weiterhin können Updates über das CLI-Tool *softwareupdate* bezogen werden. Neben der automatischen Suche und Anwendung der Updates können Anwender die Softwarepakete ebenso manuell von der Apple-Support-Website beschaffen.

4.2.1 Mac App Store

Der Mac App Store wurde zu Beginn des Jahres 2011 in MacOS in Form eines Software Updates eingeführt (vgl. Newsroom, 2010). Über den App Store konnte zuerst insbesondere Software von anderen Herstellern bezogen werden, welche dort eingestellt wurde. Apple übernimmt in diesem Fall die Verteilung und Ausführung der Updates für Hersteller und Kunden.

Für über den App Store vertriebene Software gibt Apple unterschiedliche, technische und rechtliche Vorgaben, welche von den Herstellern erfüllt werden müssen, vor. Hierzu zählt unter anderem, dass alle eingereichten Applikationen mit einem, durch Apple ausgestellten Zertifikat signiert werden müssen.

Apple bietet dem Nutzer in den Konfigurationsoptionen des Mac App Store die Möglichkeit Updates automatisch und ohne weitere Eingabe zu installieren. Im Gegensatz zu automatischen Prüfung auf neue Updates ist dies aber nicht standardmäßig aktiviert.

4.2.2 MacOS Server

MacOS Server ist ein Softwarepaket, welches im Mac App Store angeboten wird und dem Anwender die Nutzung von Server-Funktionalitäten bietet. Hiermit können die Fähigkeiten von Microsoft WSUS (siehe Abschnitt 3.2.2) zum lokalen Hosten von Updates nachgebildet werden. Möglich ist dies durch das Hosten eines *Software Update Servers* oder eines *Cache Service*[6].

[5] Weitere Informationen zur Installation von Updates unter MacOS: https://support.apple.com/de-de/HT201541 (letzter Zugriff: 20.09.17)

[6] Weitere Details siehe: https://support.apple.com/macos/server (letzter Zugriff 21.09.17)

4.3 Test

Im Rahmen des Tests wurde die Prüfung, Download und Verteilung über den Mac App Store Untersuchung. Das CLI-Tool *softwareupdater* war nicht Teil dieses Tests, da die Standardeinstellung für diese Aufgabe den Mac App Store vorsieht.

4.3.1 Testablauf

Das Update wurde über den Mac App Store gestartet. Der Netzwerkmitschnitt umfasst knapp über 840.000 Netzwerkpakete in einem Zeitraum von 27 Minuten. Der verwendete Rechner nutzte MacOS in der Version 10.10.5. Im der GUI des Software Updater wurden die in Abbildung 4.1 gezeigten vier Updates aufgeführt und installiert.

```
1 - Remote Desktop Client Update 3.8.4
2 - Remote Desktop Client Update 3.9.3
3 - Safari 10.1.2
4 - Security Update 2017-003 10.10.5
```

Auflistung 4.1: Verfügbare MacOS Updates

4.3.2 Analyse und Auswertung

Übersicht

Die meisten Pakete wurden von den in Tabelle 4.1 aufgeführten Hosts empfangen.

IP	Paketanzahl	Name
17.253.55.210	347659	swcdn.apple.com, swcdn.apple.com.akadns.net
17.253.57.207	339198	swcdn.apple.com, swcdn.apple.com.akadns.net
17.253.57.205	5033	swcdn.apple.com, swcdn.apple.com.akadns.net

Tabelle 4.1: MacOS Update Service Server

Der Hostname swcdn.apple.com ist eindeutig der Firma Apple zuzuordnen. Der zweite Hostname swcdn.apple.com.akadns.net ist über seinen Namen und durch eine Whois-Auskunft

der Firma Akamai zuzuordnen. Zudem gibt Apple an das Content-Distribution-Network von Akamai zu nutzen[7].

Updateprüfung

Nach dem Starten des Software Updates wird eine Domain Name System (DNS) Anfrage nach der Domain swscan.apple.com gestellt, um die IP-Adresse des Zielservers zu erhalten und die Updateprüfung durchzuführen. Die Antwort liefert die unterschiedlichen Hostnames und Ziel IPs.

```
1  Frame 348: 210 bytes on wire (1680 bits), 210 bytes captured (1680 bits)
2  Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.42
3  User Datagram Protocol, Src Port: 53, Dst Port: 59607
4  Domain Name System (response)
5  Flags: 0x8180 Standard query response, No error
6  Questions: 1
7  Answer RRs: 4
8  Queries
9     swscan.apple.com: type A, class IN
10       Name: swscan.apple.com
11          Type: A (Host Address) (1)
12          Class: IN (0x0001)
13
14    swscan.apple.com: type CNAME, class IN, cname swscan-cdn.apple.com.akadns.net
15    swscan-cdn.apple.com.akadns.net: type CNAME, class IN, \
16                         cname swdist.apple.com.edgekey.net
17    swdist.apple.com.edgekey.net: type CNAME, class IN, cname e5977.e9.akamaiedge.net
18    e5977.e9.akamaiedge.net: type A, class IN, addr 23.216.203.197
```

Auflistung 4.2: Gekürzte DNS-Response aus MacOS Netzwerkmitschnitt Packet 348

Der in Auflistung 4.2 genannte Server mit der Adresse 23.216.203.197 sendete im Zeitraum des Netzwerkmitschnittes 599 Pakete und empfing 422 Pakete. Der Netzwerkverkehr mit diesem Server ist über ein SSL-Zertifikat gesichert und verschlüsselt. Daher kann an dieser Stelle keine Aussage über den Inhalt der Kommunikation getroffen werden.

Beispiel 1 - Security Update 2017-003 10.10.5

Eines der erwarteten Updates war das *Security Update 2017-003 10.10.5* (Siehe Auflistung 4.1 Zeile 4). Aufgrund des HTTP-GET-Requests kann dieser Download gut nachvollzogen werden. Der Client gibt sich im User Agent als Software Update mit dem Zusatz „CFNetwork/720.5.7

[7] Dokumentation: https://support.apple.com/de-de/HT200149 (letzter Zugriff: 21.09.17)

Darwin/14.5.0 “aus. CFNetwork ist das Apple Network Framework[8] und Darwin Apples Open-Source Projekt für grundlegende Systemkomponenten[9].

```
Frame 418048: 385 bytes on wire (3080 bits), 385 bytes captured (3080 bits)
Internet Protocol Version 4, Src: 192.168.0.42, Dst: 17.253.57.207
Transmission Control Protocol, Src Port: 49471, Dst Port: 80, \
Seq: 1, Ack: 1, Len: 319
Hypertext Transfer Protocol
GET /content/downloads/46/42/091-01555/[...]/SecUpd2017-003Yosemite.pkg HTTP/1.1
Host: swcdn.apple.com
Accept: */*
Connection: keep-alive
Accept-Encoding: gzip, deflate
User-Agent: Software Update(unknown version) CFNetwork/720.5.7 Darwin/14.5.0(x86\64)
```

Auflistung 4.3: Gekürzter GET-Request aus MacOS Netzwerkmitschnitt TCP-Stream 124

Das Security Update 2017-003 ist ein kumuliertes Update und enthält eine Vielzahl an Patches bzw. Sicherheitsupdates um Sicherheitslücken zu beheben[10]. Mit dem Systemtool *pkgutil* konnte, wie in Auflistung 4.4 gezeigt, die Signatur des Updatepaketes dem System gegenüber verifiziert werden. Die Signaturkette führt bis zur Apple Root CA, welche von MacOS akzeptiert wird. Dieses Root Zertifikat kann man zudem auch von Apple direkt beziehen,[11] um eine systemunabhängige Prüfung außerhalb des MacOS Systems durchzuführen.

```
$ pkgutil --check-signature ./SecUpd2017-003Yosemite.pkg
Package "SecUpd2017-003Yosemite.pkg":
Status: signed Apple Software
Certificate Chain:
1. Software Update
   SHA1 fingerprint: 1E 34 E3 91 C6 44 37 DD 24 BE 57 B1 66 7B 2F DA 09 76 E1 FD
------------------------------------------------------------------------
2. Apple Software Update Certification Authority
   SHA1 fingerprint: FA 02 79 0F CE 9D 93 00 89 C8 C2 51 0B BC 50 B4 85 8E 6F BF
------------------------------------------------------------------------
3. Apple Root CA
   SHA1 fingerprint: 61 1E 5B 66 2C 59 3A 08 FF 58 D1 4A E2 24 52 D1 98 DF 6C 60}
```

Auflistung 4.4: Signaturprüfung SecUpd2017-003Yosemite.pkg

[8] Siehe https://developer.apple.com/documentation/cfnetwork (letzter Zugriff 20.09.17)
[9] Weiteres unter https://developer.apple.com/library/content/documentation/MacOSX/Conceptual/OSX_Technology_Overview/SystemTechnology/SystemTechnology.html (letzter Zugriff 20.09.17)
[10] Weiteres unter: https://support.apple.com/de-de/HT207922 (letzter Zugriff 20.09.17)
[11] Download von https://www.apple.com/certificateauthority/ möglich (letzter Zugriff 20.09.17)

Beispiel 2 - Remote Desktop Client Update 3.9.3

Weiterhin konnte aus dem Netzwerkmitschnitt mit Hilfe des Wireshark Object Export das *Remote Desktop Client Update 3.9.3* hergestellt werden. Der Request und Download erfolgte analog zu Beispiel 1, jedoch von einer anderen IP-Adresse.

```
Frame 453258: 382 bytes on wire (3056 bits), 382 bytes captured (3056 bits)
Internet Protocol Version 4, Src: 192.168.0.42, Dst: 17.253.57.205
Transmission Control Protocol, Src Port: 49481, Dst Port: 80, \
Seq: 1, Ack: 1, Len: 316
Hypertext Transfer Protocol
GET /content/downloads/37/16/031-21283/[...]/RemoteDesktopClient.pkg HTTP/1.1\r\n
Host: swcdn.apple.com\r\n
Accept: */*\r\n
Connection: keep-alive\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Software Update (unknown version) CFNetwork/720.5.7 Darwin/14.5.0(x86_64)
```

Auflistung 4.5: Gekürzter GET-Request aus MacOS Netzwerkmitschnitt TCP-Stream 124

Das Remote Desktop Client Update 3.9.3 betrifft die Fernwartungssoftware von Apple und dient der Fehlerbehebung (vgl. Abschnitt 2.1.1 Bugfix)[12]. Mit dem Tool *pkgutil*[13] wurde die Signatur des Updates erfolgreich geprüft (siehe Abbildung 4.6).

```
$ pkgutil --check-signature ./RemoteDesktopClient.pkg
Package "RemoteDesktopClient.pkg":
Status: signed Apple Software
Certificate Chain:
1. Software Update
   SHA1 fingerprint: 1E 34 E3 91 C6 44 37 DD 24 BE 57 B1 66 7B 2F DA 09 76 E1 FD
------------------------------------------------------------------------
2. Apple Software Update Certification Authority
   HA1 fingerprint: FA 02 79 0F CE 9D 93 00 89 C8 C2 51 0B BC 50 B4 85 8E 6F BF
------------------------------------------------------------------------
3. Apple Root CA
   SHA1 fingerprint: 61 1E 5B 66 2C 59 3A 08 FF 58 D1 4A E2 24 52 D1 98 DF 6C 60
```

Auflistung 4.6: Signaturprüfung RemoteDesktopClient.pkg

[12] Weitere Informationen unter: https://support.apple.com/de-de/HT206178 (letzter Zugriff: 20.09.17)
[13] Man-Page unter: https://developer.apple.com/legacy/library/documentation/Darwin/Reference/ManPages/man1/pkgutil.1.html (letzter Zugriff: 21.09.17)

4.3.3 Bewertung

Der MacOS Update-Mechanismus wird als aktuell sicher bewertet, trotzdem gibt es Verbesserungsbedarf. Der Mac App Store nutzt zur Updateprüfung eine sichere verschlüsselte Verbindung. Der Updatedownload erfolgt jedoch unverschlüsselt und erlaubt es den Updatestand des Client einzuschätzen. Weiterhin verwendet Apple für Signaturen einen langfristig nicht mehr sicheren Algorithmus.

Updateprüfung

Seit OS X 10.8.5 erfolgt die Prüfung auf neue Updates über eine SSL/TLS-Verbindung mit dem Host https://swdist.apple.com[14]. Wie im Abschnitt Updateprüfung gezeigt kann dieses Verhalten bestätigt werden. Das Vorgehen entspricht dem geforderten Sicherheitsniveau. Durch die Verschlüsselung kann die Kommunikation, auf die CIAA-Schutzziele bezogen, Vertraulich durchgeführt werden.

Updateübertragung

Die erwarteten Updatepakete konnten durch unverschlüsselte HTTP-GET Requests im Netzwerkmitschnitt erkannt werden. Diese Requests können als Quelle für Schlüsse über den Updatestand des Clients dienen. Das Schutzziel der Vertraulichkeit wird an dieser Stelle nicht ausreichend erfüllt. Durch Hostnamen und IP-Adressen der beteiligten Server kann bestätigt werden, dass Apple ein Content-Delivery-Network einsetzt um die Verfügbarkeit der Updates zu verbessern.

Updateausführung

Beide in den Beispielen untersuchten Pakete sind mit einem gültigen Zertifikat signiert und konnten verifiziert werden. Über den Mac App Store bezogene Updates werden nach Aussage von Apple auf gültige Signaturen überprüft[15]. Weiterhin werden Anwendungen in MacOS vor dem Starten durch das Systemtool *Gatekeeper* auf gültige Signaturen überprüft. Eine Änderung der Dateien, beispielsweise durch Einfügen von Schadcode, würde den Hash-Wert der Software verändern. In der Folge wäre auch die auf dem ursprünglichen Hash-Wert basierende Signatur ungültig. Die doppelte Signaturprüfung bei der Installation und Ausführung der Software erfüllt aktuell jedoch noch die Schutzziele Integrität und Authentizität.

[14] Weiter Informationen unter: https://support.apple.com/de-de/HT200149 (letzter Zugriff: 21.09.17)
[15] Weitere Informationen zur automatischen und manuellen Prüfung unter https://support.apple.com/en-us/HT202369 (letzter Zugriff: 20.09.17)

5 Linux

5.1 Grundlagen

Ubuntu ist mit nach eigenen Angaben ungefähr 40 Millionen Desktop Nutzern[1] eine der meist verbreiteten Linux Distributionen. Basierend auf der Linux Distribution Debian nutzt Ubuntu beispielsweise deren Paketsystem. Weiterhin werden für Debian geupdatete Pakete, im Rahmen des Entwicklungszyklus, auch den Ubuntu-Nutzern zur Verfügung gestellt.

Als einer der größten Vertreter von Open-Source Software wird Ubuntu daher stellvertretend für diese Gruppe betrachtet. Open-Source Software definiert sich durch die Möglichkeit, den Software Quelltext zu erhalten, zu modifizieren und zu verbreiten. Die genaue Definition dieser Rechte erfolgt über Lizenzen[2].

Im Gegensatz zu vielen anderen Distributionen, welche alleine von Vereinen oder Freiwilligen gepflegt werden, erfolgt ein großer Teil des Supports und der Entwicklung für Ubuntu durch die Firma Canonical[3].

Ubuntu nutzt, wie auch Debian, den Linux Kernel. Der Linux Kernel ist ein monolithischer Kernel, der in verschiedenen Open-Source Projekten als Betriebssystemkernel genutzt wird. Die Entwicklung des Kernels findet, losgelöst von der Entwicklung der anderen Betriebssystemkomponenten, im Linux Kernel Projekt statt, welches von Freiwilligen und Kernel-Gründer und Entwickler Linus Torvalds und getragen wird. Die Verteilung und das Update des Kernels und weiterer Betriebssystembestandteile obliegen aber den Distributionsverantwortlichen.

[1] Weitere Angaben von Canonical: https://insights.ubuntu.com/about/ (letzter Zugriff: 21.09.17)
[2] Übersicht über viele Lizenzen: https://opensource.org/licenses (letzter Zugriff: 14.09.17)
[3] Offizielle Beschriebung des Verhältnisses zwischen Canonical und Ubuntu https://www.ubuntu.com/about/canonical-and-ubuntu (letzter Zugriff: 20.09.17)

5.1.1 Releases

Neue Versionen von Ubuntu erscheinen in einem zeitbasierten 6-monatigen Zyklus und werden als **Regular** bezeichnet[4]. Diese Versionen werden neun Monate mit Updates versorgt. Upgrades mit neuen Features werden grundsätzlich nur mit einer neuen Ubuntu Version veröffentlicht. Die Software, welche aus inoffiziellen Quellen bezogen wird, kann jedoch selbstständig neue Versionen und Features veröffentlichen.

Im zweijährigen Rhythmus erscheint statt der Regular Version eine **Long-Term Support** (LTS) Version, welche für einen Zeitraum von fünf Jahren mit Security Updates versorgt wird. Upgrades mit neuen Features sind davon grundsätzlich ausgenommen. Updates zur Unterstützung neuer Hardware zählen hingegen zum Support.

Weiterhin kann im kommerziellen Ubuntu Advantage Program kostenpflichtig verlängerter Support (Extended Security Maintenance) bezogen werden. LTS Versionen durchlaufen ein erweitertes Testfenster und haben das Ziel zur langfristig in Unternehmen genutzt zu werden. Teilweise besteht hier der Bedarf nach stabilen Systemen, welche von neuen Features unbeeinflusst laufen. Dies betrifft besonders auch die angepassten Ubuntu Systeme für Server, IOT und Cloud.

Release	Regular	Long-Term Support
Erscheinungshäufigkeit	6 Monate	2 Jahre
Servicefrist	9 Monate	5 Jahre

Tabelle 5.1: Ubuntu Release Übersicht

Auf Ubuntu aufbauend werden weitere angepasste Distributionen angeboten. Einige davon werden offiziell anerkannt und durch die Ubuntu Community unterstützt[5]. Eine der größten[6], nicht durch die Ubuntu Foundation oder Canonical unterstützten Distributionen ist Linux Mint. Die meisten dieser Distributionen folgen dem Ubuntu Release Zyklus.

[4] Erläuterung dieser Vorgehensweise unter: https://wiki.ubuntu.com/TimeBasedReleases (letzter Zugriff: 21.09.17)
[5] Weiteres unter https://wiki.ubuntu.com/DerivativeTeam/Derivatives (letzter Zugriff: 21.09.17)
[6] Übersicht über die Popularität der Linux Distributionen: http://distrowatch.com/dwres.php?resource=popularity (letzter Zugriff: 22.09.17)

5.1.2 Repositories

Paketquellen werden im Linux Sprachgebrauch meist als Repositories bezeichnet. Diese enthalten eine Sammlung aller zugeordneten Pakete. Die Repositories unterscheiden sich nach Unterstützung und Lizenz (siehe Tabelle 5.2[7]).

Im **Main** Repository enthaltene Pakete nutzen nur Open-Source Code. Weiterhin muss die Lizenz die Veränderung und Weitergabe der Software unter der gleichen Lizenz ermöglichen. Über das Main Repository beziehbare Pakete werden von Canonical gepflegt. Dies garantiert dem Nutzer eine sichere und langfristige Versorgung mit Updates.

Universe steht für das von der Community unterstützte Repository für Open-Source Software. Weniger häufig genutzte Pakete für Open-Source Software werden hier verwaltet. Canonical stellt keinen Support und Updates werden durch die Community erbracht.

Das **Restricted** Repository ist die Paketquelle für alle von der Lizenz, bzw. Verfügbarkeit des Source-Codes eingeschränkten, aber durch Canonical unterstützten Pakete. Besonders sind zum Betrieb des Betriebssystems Gerätetreiber der Hersteller notwendig. Diese enthalten firmeneigenen Quellcode, welcher Closed-Source ist. Updates werden teilweise durch die Hersteller erbracht.

Alle weiteren Packete, welche durch die Community unregelmäßig mit Updates versorgt werden, finden sich im Repository **Multiverse**. Die Software aus diesem Repository kann unter verschiedenen Lizenzen liegen. Aufgrund des unklaren Supportstatus der Pakete in diesem Repository ist von der Verwendung abzuraten. Populäre Beispiele sind Multimedia-Codecs.

Canonical Partner Repositories bieten Zugang zu Closed-Source-Software und enthalten im Gegensatz zum Restricted Repository auch Applikationen. Dieses Repository wird von Canonical betrieben und den kommerziellen Partnern teilweise gewartet.

Personal Package Archives sind Repositories zum Hosten von Softwarepaketen, welche als zusätzliche Quellen eingebunden werden können. Der Vertrieb der Software erfolgt ohne Einfluss der Community oder Canonical und ist ausschließlich vom Besitzer des PPAs abhängig. Dieser ist auch für Updates verantwortlich. Da die Updates außer der Zugehörigkeit zum

[7] Weiteres im Ubuntu Policy Manual: http://people.canonical.com/~cjwatson/ubuntu-policy/policy.html/ch-archive.html (letzter Zugriff: 14.09.17)

spezifischen PPA nicht überprüft werden, muss der Anwender dem PPA-Besitzer vertrauen. Beispiele für bekannte PPAs sind LibreOffice und Wine.

	Main	Universe	Restricted	Multiverse
Canonical Support	x		x	
Community Support		x		x
Open Source	x	x		
Restricted License			x	x

Tabelle 5.2: Übersicht Packetquellen

5.2 Software

5.2.1 Advanced Packaging Tool - APT

Das Advanced Packaging Tool (APT) ist eine von Debian übernommene Applikation zur Paketverwaltung. Zu den grundlegenden Aufgaben gehören die Suche, das Installieren und Entfernen von Programmpakten. Hierfür nutzt APT unterschiedliche Quellen (siehe Abschnitt 5.1.2). APT bezeichnet das Laden der Inhalte der verbundenen Repositories als Update. Der Download und die Installation der neuen Pakete wird als Upgrade bezeichnet. Zur Sicherung der Authentizität prüft APT die Signaturen der Repositories. Weiterhin werden bei der Installation die Abhängigkeiten von anderen Paketen geprüft und, falls möglich, behoben.

Ein Distributions-Upgrade beinhaltet zusätzlich das intelligente Auflösen von Abhängigkeiten, um neue Versionen zu installieren. Hierbei können auch Pakete entfernt werden.

5.2.2 Software Update

Um den Nutzern eine vereinfachte Bedienung zu ermöglichen, bietet Ubuntu die Software *Update Manager*[8], welcher die Funktionalität der Konsolenanwendung APT in einer GUI darstellt. Diese führt automatisch ein Update der in den Repositories vorliegenden Informationen durch und zeigt mögliche Software-Updates an.

[8] Softwarebeschreibung unter https://launchpad.net/update-manager (letzter Zugriff: 22.09.17)

5.3 Test

5.3.1 Testablauf

Im Rahmen der Untersuchung wurde ein Update von Ubuntu 17.10 durchgeführt. Zum Netzwerkmitschnitt zählen mehr als 375.000 Pakete, die 460 MB umfassen. Der Mitschnitt dauerte 17 Minuten.

5.3.2 Analyse und Auswertung

Übersicht

Neben den erwarteten .deb-Dateien finden sich im Netzwerkmitschnitt noch nicht gekennzeichnete .pkg und .txt Dateien. Die Pakete werden unverschlüsselt mit einem HTTP-GET-Request über Port 80 geladen.

In Tabelle 5.3 sind die Hosts gezeigt, welche die meisten Pakete an den Client sendeten. Die *.ubuntu.com Domains sind der Firma Canonical zugeordnet, die *.gnome.com Domains gehören der Gnome Foundation.

IP	Paketanzahl	Name
91.189.91.26	33927	security.ubuntu.com, us.archive.ubuntu.com
209.132.180.168	465	proxy.gnome.org, extensions.gnome.org
91.189.91.23	427	security.ubuntu.com, us.archive.ubuntu.com

Tabelle 5.3: Ubuntu Update Service Server

Updateprüfung

Die Updateprüfung findet unverschlüsselt im Klartext statt. Zur Sicherung der Authentizität sind die Dateien signiert. Weiterhin werden die in den authentifizierten Dateien angegebenen Hash-Werte überprüft. Mit einem HTTP-GET-Request an us.archive.ubuntu.com/ubuntu/dists/artful/InRelease wird die folgende Datei geladen (Auflistung 5.1).

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Origin: Ubuntu
Label: Ubuntu
Suite: artful
Version: 17.10
Codename: artful
Date: Tue, 05 Sep 2017 11:18:04 UTC
Architectures: amd64 arm64 armhf i386 ppc64el s390x
Components: main restricted universe multiverse
Description: Ubuntu Artful 17.10
MD5Sum:
bdf128a92d0f99ebdd6f7f4d5433fe01        573843633 Contents-amd64
3ef4e5ff1dc8b7fd30d79a71988616a4         36093575 Contents-arm64.gz
[...]
SHA256:
6761fe44a96cefcedef3d8bf7eba445992b4e23b3526df2f5651cd424c4dff73
                                        573843633 Contents-amd64
13afe325bdad4d69dfffaca22b4c3af1c807d2c612eb1c8eb8c39796350857ed
                                        36093575 Contents-arm64.gz
[...]
97656bcd1ae8d986bc4996928209d15c6ff0e6484745b854a1c154fc1e56cb9a
                                        1129784 main/binary-amd64/Packages.xz
[...]
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.11 (GNU/Linux)

iQIcBAEBCgAGBQJZrojWAAoJEDtP5qzAsh8ycSkP/RFqAyC+422+koVlg8aAZJGw
hh1vV4SsjOLvgsOtf52hPZMU1PrQ4UM/0eU7PB4BKoh6zaM6V1hQjgXJ7edO/irM
uFZZ3J/XfK2NakoYVWuYtrI00aPD7a7zPicMKNMDY3kj188Mce2+vQiFosgF36q+
GbMnuvuM59H/Y4AjbtA0VXY/AluHOkuOnp6WcGRKnNk7RxYob8wU/h0DdqxcIegc
m1TvLgfGBPBlsTXpVwQJiFnJObbM9q7iJFZlWgLn+pbnJBPfoX0fawfCi8sdY0Ms
o64Ey85ejc+30fUXFKTAiWU4gQX3Ombl5Pn6YKeQmB4zDvks66sLvAvO5XExs5Yv
dmpjT3sUiiQZGzw5QnCDrrwKsPOuDjhep6DvhoxSf5HbIuI8S3w51HjlHI0ggua0
W9aqUjJWLSs8ga8Tr6nDgv9iXw7IAbzOg5mCKGFujF3Qo4+G+c1sL2Ru7/pz3EdD
3Ar+woQlIylhyj60IjJFf8TqslNQDJCkxSjJu21F5jAyqgBW+Lh61D2oFQoNFJ+4
C6Mf/P2jsodxBUesX6fKN5ugCFJ1blwo3OagzYyeC5ECDzpFMZjsz0dZGup9cOcJ
HAU1T5zuBYVf7JKUknj+ozEDgb04xAOgOjU+xgiVHBlNrhb/+BgDvCRsCJldjl8a
sl9chBhL4U57U4wP16FI
=Zxng
-----END PGP SIGNATURE-----
```

Auflistung 5.1: Datei InRelease (gekürzt)

Die gezeigte Datei enthält die den Repositories und Architekturen zugeordnete Hash-Werte, Größen und Bezeichnungen sowie eine Signatur. Der Client entscheidet nach seinen Einstellungen und der Prozessorarchitektur (bspw. amd64) welche Dateien heruntergeladen werden.

Die nachfolgende Tabelle 5.4 wurde aus den HTTP-GET-Requests an den Host us.archive.ubuntu.com ermittelt und zeigt die heruntergeladenen Repository Packages. Der Request erfolgte anhand des in Auflistung 5.1 gezeigten Hash-Wertes des Repositories. Die Dateien sind mit dem XZ-Verfahren[9] komprimiert.

Hash(short)	Size in Byte	Name	Package Nr.
97656bcd[...]	1129784	main/binary-amd64/Packages.xz	465
34134e8d[...]	1126096	main/binary-i386/Packages.xz	1577
f1f36fc8[...]	8700	restricted/binary-i386/Packages.xz	1578
08ad3461[...]	2826300	universe/dep11/Components-amd64.yml.xz	2647
7d2400a1[...]	7843672	universe/dep11/icons-64x64.tar.gz	3155
7317b3d2[...]	146672	multiverse/binary-i386/Packages.xz	3420
040df273[...]	153604	multiverse/binary-amd64/Packages.xz	3688

Tabelle 5.4: Ubuntu Repository Package Download

Beispielhaft wurde die in Tabelle 5.4 gezeigte, erste heruntergeladene Datei untersucht. Die heruntergeladenen Dateien können anhand ihres Hash-Wertes auf Integrität geprüft werden (siehe Auflistung 5.2 Zeile 2 und Tabelle 5.4 erster Eintrag).

```
1 $ sha256sum Packages.xz
2 97656bcd1ae8d986bc4996928209d15c6ff0e6484745b854a1c154fc1e56cb9a  Packages.xz
```

Auflistung 5.2: SHA-256 HASH Packages.xz

Die heruntergeladene Datei enthält eine Liste der in diesem Repository verfügbaren Pakete. Als Auszug sind in Auflistung 5.3 die Informationen zu den in den folgenden Beispielen 1 und 2 untersuchten Paketen aufgeführt. Hierzu zählen besonders die zur Verifikation der Integrität notwendigen Hash-Werte und die für die Installation notwendige Angabe der Abhängigkeiten (siehe Absatz 5.2.1), aber auch allgemeine Informationen wie der Paketverantwortliche (Maintainer) und die Kurzbeschreibung des Paketes.

[9] Für weitere Informationen siehe https://tukaani.org/xz/format.html (letzter Zugriff: 21.09.17)

```
[...]
Package: linux-image-generic
Maintainer: Ubuntu Kernel Team <kernel-team@lists.ubuntu.com>
Architecture: amd64
Source: linux-meta
Version: 4.12.0.12.13
Depends: linux-image-4.12.0-12-generic, linux-image-extra-4.12.0-12-generic,
linux-firmware
Recommends: thermald
Filename: pool/main/l/linux-meta/linux-image-generic_4.12.0.12.13_amd64.deb
Size: 2222
MD5sum: 1f61921f369af4ebf2cc8ff6e91fe578
SHA1: e62f7ed97be8927b8f8448dd42602cc826ba165b
SHA256: 21120b12eac6af79419df9405235af0117cdb1c1a61b61dca9193ce338f9db57
Description: Generic Linux kernel image
Description-md5: 6d632579c673704f44b290b16e7dbfd1
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Origin: Ubuntu
Supported: 9m
[...]
Package: gcc-6-base
Maintainer: Ubuntu Core developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian GCC Maintainers <debian-gcc@lists.debian.org>
Architecture: amd64
Source: gcc-6
Version: 6.4.0-4ubuntu1
Breaks: gcc-4.4-base (<< 4.4.7), gcc-4.7-base (<< 4.7.3), gcj-4.4-base (<< 4.4.6-9~),
gcj-4.6-base (<<    4.6.1-4~), gnat-4.4-base (<< 4.4.6-3~), gnat-4.6 (<< 4.6.1-5~)
Filename: pool/main/g/gcc-6/gcc-6-base_6.4.0-4ubuntu1_amd64.deb
Size: 16600
MD5sum: e1c53bdd520b1da5a923a82e32e9d7ab
SHA1: 939b70f2308a249a4e6acfe155eefe5ad39c630c
SHA256: c1601edccb5075db408fff69efaf3d2aa59730f804a9c48781a97baa02475c47
Description: GCC, the GNU Compiler Collection (base package)
Multi-Arch: same
Homepage: http://gcc.gnu.org/
Description-md5: b6e93638a6d08ea7a18929d7cf078e5d
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Origin: Ubuntu
Supported: 9m
[...]
```

Auflistung 5.3: Gekürzter Auszug aus Packages.xz

Beispiel 1 - linux-image-generic

Unter den Updates befinden sich für das System essentielle Bestandteile. Hierzu gehört das Linux Kernel Image. Der Download dieser Datei wird mit einem HTTP-GET-Request and die

Domain us.archive.ubuntu.com über Port 80 gestartet.

```
1 Frame 349385: 218 bytes on wire (1744 bits), 218 bytes captured (1744 bits)
2 Internet Protocol Version 4, Src: 10.0.2.15, Dst: 91.189.91.26)
3 Transmission Control Protocol, Src Port: 52284, Dst Port: 80,
4 Seq: 11270, Ack: 141652805, Len: 164)
5 Hypertext Transfer Protocol
6 GET /ubuntu/pool/main/l/linux-meta/linux-image-generic_4.12.0.12.13_amd64.deb
7 HTTP/1.1
8 Host: us.archive.ubuntu.com
9 User-Agent: Debian APT-HTTP/1.3 (1.5~beta1)
```

Auflistung 5.4: Gekürzter GET-Request aus Linux Netzwerkmitschnitt TCP-Stream 124

Durch Bestimmung des Hash-Wertes und Vergleich mit dem in Auflistung 5.5 Zeile 2 und Auflistung 5.3 Zeile 17 angegebenen Werten kann die Integrität des Paketes bestätigt werden.

```
1 $ sha256sum linux-image-generic_4.12.0.12.13_amd64.deb
2 21120b12eac6af79419df9405235af0117cdb1c1a61b61dca9193ce338f9db57
```

Auflistung 5.5: SHA-256 HASH linux-image-generic_4.12.0.12.13_amd64.deb

Beispiel 2 - gcc

Die Abkürzung GCC steht für die GNU Compiler Collection, ein Compiler für diverse Programmiersprachen. Als Compiler stellt GCC einen wichtigen Systembestandteil dar und ein Kompromittierung dieses Paketes würde sämtliche, compilierte Software gefährden.

```
1 Frame 166445: 206 bytes on wire (1648 bits), 206 bytes captured (1648 bits)
2 Internet Protocol Version 4, Src: 10.0.2.15, Dst: 91.189.91.26
3 Transmission Control Protocol, Src Port: 52280, Dst Port: 80,
4 Seq: 7527, Ack: 49818153, Len: 152
5 Hypertext Transfer Protocol
6 GET /ubuntu/pool/main/g/gcc\-6/gcc\-6\-base\_6.4.0\-4ubuntu1\_amd64.deb HTTP/1.1
7 Host: us.archive.ubuntu.com
8 User-Agent: Debian APT-HTTP/1.3 (1.5~beta1)
```

Auflistung 5.6: Gekürzter GET-Request aus Linux Netzwerkmitschnitt TCP-Stream 16

Analog zu Beispiel 1 kann hier die Integrität durch einen Vergleich der Hash-Werte verifiziert werden (siehe Auflistung 5.7 Zeile 2 und Auflistung 5.3 Zeile 39).

```
1 $ sha256sum gcc-6-base_6.4.0-4ubuntu1_amd64.deb
```

```
2 c1601edccb5075db408fff69efaf3d2aa59730f804a9c48781a97baa02475c47
```

Auflistung 5.7: SHA-256 HASH gcc-6-base_6.4.0-4ubuntu1_amd64.deb

5.3.3 Bewertung

Der Update-Mechanismus von Ubuntu ist als ausreichend sicher zu bewerten. Ubuntu unterscheidet sich durch den Verzicht auf Code-Signaturen grundlegend von den anderen Betriebssystemen. Trotzdem kann wird durch Nutzung Vertrauenskette das Schutzziel der Authentizität für den Update-Mechanismus erfüllt. Durch die unverschlüsselte Datenübertragung bei der Updateübertragung ist die Vertraulichkeit, bezogen auf den Updatestand jedoch nicht gegeben.

Updateprüfung

Die Updateprüfung von Ubuntu erfolgt im Klartext. Zu beachten ist jedoch, dass die Festlegung der zu installierenden neuen Pakete auf dem Client getroffen wird. Benötigt werden dafür lediglich die Übersicht der Paketquellen und die Inhalte der Repositories. Da jedoch alle Clients mit gleichen Repository Einstellungen und Architektur die gleichen Informationen erhalten, können daraus keine Informationen über den Updatestand abgeleitet werden.

Die Repository-Übersichten werden anhand ihres Hash-Wertes heruntergeladen. Dies erschwert einen Angriff, da ein Angreifer zusätzlich die passenden Hash-Werte extrahieren müsste.

Obwohl die Vertraulichkeit in der Kommunikation nicht gegeben ist, erfüllt der Updatestand, also die schützenswerte Information, in dieser Phase weiterhin das Schutzziel der Vertraulichkeit.

Updateübertragung

Der Download der neuen Softwarepakete erfolgt unverschlüsselt über HTTP. Die Vertraulichkeit ist dabei nicht mehr gegeben. Die Pakete werden aus dem gewünschten Repository bezogen. Aus diesen Requests können die Updatepakete und die Rechnerarchitektur abgeleitet werden. Dies ermöglicht Rückschlüsse auf den Systemzustand und Updatestand.

Um die Verfügbarkeit zu verbessern, besteht die Möglichkeit, Kopien der Repositories selber zu hosten.

Updateausführung

Grundsätzlich werden durch APT nur neue Pakete installiert und keine Änderungen an Paketen verarbeitet. Durch die Überprüfung der Hash-Werte kann die Integrität indirekt auch die Authentizität der heruntergeladenen Pakete gewährleistet werden. Hierfür werden die neu zu installierenden Pakete gehasht und mit den in der Übersicht des Repositories erhaltenen Werten abgeglichen (siehe Auflistung 5.3). Die Integrität der Paketübersicht ist ebenfalls durch den Hash-Wert Vergleich mit der signierten Liste der Repositories, gesichert. Hierdurch ergibt sich eine zusammenhängende Vertrauenskette (chain-of-trust), die das neue Paket sichert.

Ubuntu führt vor der Ausführung der Software keine weitere Prüfung (bspw. in Form einer Signaturverifikation des Codes) durch. Neben den technischen Herausforderungen stellt für diese Maßnahme aber vor allem die Philosophie der Open-Source Bewegung ein Hindernis dar. Ziele der Open-Source Bewegung sind unter anderem die freie Verteilung, Veränderung von Software und dem zugehörigen Source Code (vgl. Open Source Initiative, 2017). Jedwede Veränderung am Code würde zu einer defekten Signatur führen. In Folge müsste der Code also neu, mit einem vorliegenden Schlüssel signiert werden. Da diese Vorgänge aber an unterschiedlichen Stellen in der Community durchgeführt werden, müsste eine Schlüssel- bzw. Zertifikatsverwaltung genutzt werden. Über viele Open-Source Projekte hinweg ist diese Entwicklung aktuell nicht absehbar.

6 Zusammenfassung und Bewertung

Ziel der Untersuchung war die Frage nach der Sicherheit der Update-Mechanismen der untersuchten Betriebssysteme. Alle untersuchten Systeme nutzen in Teilen sichere Verfahren. Trotzdem sind die Mechanismen nicht fehlerfrei und durchgängig sicher.

Um in den unterschiedlichen Phasen Aussagen über das Systemverhalten geben zu können wurde ein Netzwerkmitschnitt des Updatevorganges der drei Betriebssysteme durchgeführt. Zusätzlich wurden die im Prozess verwendeten Dateien und Prozesse untersucht und beobachtet.

Hierzu kann festgestellt werden, dass alle untersuchten Betriebssysteme durch ihren Update-Mechanismus Informationen über den aktuellen Updatestand des Systems preisgeben. Das ist problematisch, da mit dem Wissen über den Updatestand ein Angriff zielgerichtet konstruiert werden kann. Das Schutzziel der Vertraulichkeit wird von allen Systemen nicht ausreichend erfüllt. Nachfolgend werden die Updatephasen aufgeführt und Probleme der einzelnen Betriebssysteme erläutert.

6.1 Updateprüfung

Alle drei untersuchten Systeme nehmen zur Updateprüfung Kontakt zu einem externen System auf. Bei Windows und MacOS erfolgt diese Kommunikation zur Updateprüfung verschlüsselt über SSL/TLS. Da die ausgetauschten Informationen nicht bekannt sind, kann das Updateprüfungsverfahren für diese beiden Systeme nicht weiter untersucht werden. Durch die Verschlüsselung wird das Schutzziel der Vertraulichkeit erfüllt.

Unter Ubuntu läuft die Updateprüfung offen ab. Trotzdem kann ein potentieller Angreifer aus den sichtbaren Informationen keine ausreichenden Rückschlüsse auf den aktuellen Systemzustand ziehen, da unterschiedliche Ubuntu-Systeme die gleichen Informationen vom Updateserver erhalten. Daher erfüllt auch Ubuntu während der Updateprüfung das Ziel.

Bei allen drei Systemen ist eine Veränderung der in der Updateprüfung versendeten Informationen nach aktuellem Stand nicht trivial und praktikabel. Dies liegt an der Verschlüsselung von Windows und MacOS und der Vertrauenskette aus Signatur und Hash-Werten die Ubuntu nutzt.

6.2 Updateübertragung

Die Updateübertragung aller untersuchten Systeme erfolgt unverschlüsselt über das HTTP Protokoll. Die Übertragung im Klartext, nach Anforderung der Daten mit HTTP-GET-Requests stellt ein Risiko dar.

Ein Angreifer, welcher Zugriff auf den Netzwerkverkehr hat, kann in dieser Phase genaue Informationen über den Updatestand eines Systems sammeln. Grundlage ist die Annahme, das ein Update zur Zeit des Downloads noch nicht auf dem System eingespielt ist. Kombiniert man dies mit den Informationen, welche spezifische Sicherheitslücke durch das Update geschlossen wurde, so kann, abhängig von der Sicherheitslücke, ein Angriff konzipiert werden.

Besonders gefährdet sind Systeme mit Sicherheitslücken, die zu einer Remote Code Execution führen, welche über das Netzwerkinterface erreicht wird, da davon auszugehen ist, dass ein Angreifer welcher das Netzwerk überwacht, auch Zugriff auf diese Schnittstelle hat. Ein solcher Angriff könnte in Netzwerken mit nicht vertrauenswürdigen Teilnehmern wie öffentliche WLANs durchgeführt werden.

Die Lösung für die fehlende Transportsicherung im Update-Mechanismus kann die Einführung von HTTPS (HTTP over SSL/TLS) sein. Hierfür müssten nur geringfügige Systemanpassungen vorgenommen werden. Zu beachten ist jedoch, dass durch die Nutzung von Verschlüsselung beim Datentransport eine geringfügig höhere Rechnerlast auf Seiten des Clients und Servers entsteht. Auch wenn dies nicht für alle Systeme von Bedeutung ist, müssen bei Updatesystemen die hohe Anzahl an Clients, mehrere Millionen, und die Updategröße, mehrere Hundert Megabyte, berücksichtigt werden.

6.3 Updateausführung

Wie auch durch Vorgaben empfohlen prüfen Windows 10 und MacOS prüfen vor der Ausführung von Softwarepaketen, ob diese signiert sind (vgl. Bundesamt für Sicherheit in der

Informationstechnik, 2016, M 4.177). Die Signatur muss auf die im System vorliegenden Herstellerzertifikate oder die importierten Zertifikate Dritter zurückführbar sein. Beide Hersteller geben dem Nutzer die Einstellungsmöglichkeit, ob ausschließlich signierte und verifizierte Software ausgeführt werden darf.

Eine Signaturüberprüfung beim Starten von Software erfolgt unter Ubuntu grundsätzlich nicht. Die Linux Open-Source-Community setzt sich für freien Vertrieb und Modifikation von quelloffener Software ein. Die Modifikation von signiertem Code ist aber nicht möglich, ohne die Signatur ungültig zu machen. In Folge müsste also modifizierter Code neu vom Nutzer signiert werden.

Essentiell für die Sicherheit von Signaturen ist die Wahl des Signaturalgorithmus. Microsoft und Apple verwenden den SHA-1 Algorithmus zum Signieren von Code. SHA-1 gilt jedoch nicht mehr als sicher. Hierzu schreibt Eckert (2014) „An der langfristigen Sicherheit von SHA-1 [muss] ernsthaft gezweifelt werden“ (S.389). Microsoft signiert daher seine Updatepakete zusätzlich mit einem zweiten SHA-256 Zertifikat (Dual-Sign). Viele Unternehmen reagieren auf die zunehmende Problematik der weiteren Nutzung unsicherer Algorithmen wie MD5 oder SHA-1 und empfehlen eine Aussteuerung alter Zertifikate und Algorithmen und Umstellung auf neuere Hash-Verfahren wie SHA-256 für die Nutzung in Signaturen.

6.4 Bewertung

Windows, Linux und MacOS unterscheiden sich, grundsätzlich erheblich in den Nutzerzahlen. Windows 10 hat 400 Millionen Nutzer[1]. MacOS wird versionsübergreifend von ungefähr 100 Millionen Anwendern[2] genutzt, Ubuntu erreicht etwa 40 Millionen Desktop-Nutzer[3]. Auch, wenn diese Zahlen, da sie von den Herstellern stammen und nicht verifiziert werden können skeptisch zu betrachten sind, so ist doch eine ungefähre Einordnung möglich. Aus dieser Betrachtung muss der Schluss zu ziehen sein, dass an die Updateverteilungssysteme von Windows und MacOS höhere Anforderungen im Betrieb gestellt werden, als an Ubuntu. Aufgrund der hohen Nutzerzahl ist es daher für diese Systemhersteller besonders nötig Ihre Update-Mechanismen effizient zu gestalten (vgl. Gkantsidis u. a., 2006).

[1] Eigene Angaben: https://news.microsoft.com/bythenumbers/windows-ten (letzter Zugriff: 22.09.17)
[2] Aussagen aus Interview unter: https://techcrunch.com/2017/04/04/apple-pushes-the-reset-button-on-the-mac-pigerro/(letzter Zugriff: 22.09.17)
[3] Selbstdarstellung von Ubuntu: https://insights.ubuntu.com/about/ (letzter Zugriff: 22.09.17)

Für den Updatedownload wird die Nutzung von SSL/TLS gefordert. Dies würde einen zusätzlichen Rechenaufwand zur Verschlüsselung erzeugen. Ebenso müsste die Verteilung der SSL/TLS Zertifikate auf sämtliche Stellen, insbesondere auch das CDN, verwaltet werden.

Die aus der Untersuchung der Updateausführung geforderte Umstellung auf sicherere Zertifikatsalgorithmen, zum Signieren von Code, ist ein langwieriger Prozess. Wird ein alter Zertifikatsalgorithmus für ungültig erklärt, läuft so signierte Software nicht mehr. Gerade bei langlebigen Softwareprodukten welche keine Updates erhalten, stellt das ein Problem dar. Daher ist eine lange Übergangsphase notwendig. Microsoft plant diese bereits (vgl. Microsoft Corporation, 2017, S.5). Apple gibt derzeit nur SHA-1 als beim Signieren von Code unterstütztes Hash-Verfahren an (vgl. Apple, 2017).

7 Fazit

Diese Arbeit zeigt eine erweiterte Einführung in die Update-Mechanismen der drei großen Desktop-Betriebssysteme Windows, Linux und Mac. Die Updateprozesse wurden analysiert, bewertet und besonders auf ihre Sicherheit untersucht. Hierzu folgte die Unterteilung des Prozesses in Updateprüfung, Updateübertragung und Updateausführung. In diesen Phasen wurde dann die Erfüllung der Schutzziele Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität bewertet.

Die Sicherheit der Update-Mechanismen ist aktuell nicht ausreichend und könnte durch einfache Maßnahmen deutlich verbessert werden.

Es konnte festgestellt werden, das alle Systeme in der ersten Phase der Updateprüfung als sicher anzusehen sind. Hierfür sorgt unter Windows und MacOS die Verschlüsselung, welche die Vertraulichkeit bei der Prüfung garantiert. Die Ubuntu-Updateprüfung ist so konzipiert, dass aus den gesammelten Daten keine Rückschlüsse auf den aktuellen Systemzustand getroffen werden können.

Zudem konnte gezeigt werden, dass es im Rahmen der zweiten Phase der Updateübertragung deutlichen Verbesserungsbedarf gibt. Das Herunterladen der Updates erfolgt durch alle untersuchten Systeme unverschlüsselt über eine HTTP-Verbindung. Dies kann durch einen Angreifer genutzt werden, um Informationen über das System zu gewinnen und einen Angriff durchzuführen. Hier kann durch den Einsatz von SSL/TLS die Vertraulichkeit und damit die Sicherheit deutlich erhöht werden.

Sowohl Microsoft mit Windows als auch Apple mit MacOS nutzen Content-Distribution-Networks (CDN) um die Verfügbarkeit der Updates zu gewährleisten. Hierzu muss beachtet werden, dass alle Systeme sich deutlich in der Anzahl der Nutzer unterscheiden. Der Bedarf der Nutzung eines CDN besteht aktuell bei Ubuntu nicht. Alle Systeme erfüllen das Ziel der Verfügbarkeit.

Weiterhin ist zu beobachten, dass bei der Updateausführung (der dritten Phase) alle Systeme Verfahren verwenden, um die Integrität und Authentizität der Updatepakete sicherzustellen. Windows und MacOS nutzen zusätzliche Schutzmaßnahmen um Software vor der Ausführung generell auf ihre Authentizität zu prüfen. Beide nutzen aktuell noch Signaturen mit SHA-1 Hashes. Diese sind nicht mehr langfristig sicher und sollten durch Nachfolgestandards ersetzt werden, gerade da die Umstellung auf einen neuen Signaturalgorithmus einen langen Übergangszeitraum erfordert.

Das Ziel der Untersuchung konnte erfolgreich erreicht werden. Langfristig ist eine Betrachtung der Veränderung der Betriebssystem-Update-Mechanismen erforderlich, um Aussagen über durch die Hersteller getroffene Anpassungen und Änderungen treffen zu können. Weiterhin bietet jedes einzelne Betriebssystem Möglichkeiten, den Update-Mechanismus in Details zu prüfen. Hierzu zählen bezüglich der Sicherheit besonders Untersuchungen der Update-Mechanismen gegen Resistenz vor Man-in-the-Middle Angriffen, welche unter anderem die Aspekte Certificate-Pinning, DNS-Spoofing, SSL/TLS-Stripping/Downgrade behandeln könnten. Auf das Themengebiet der verteilten Systeme bezogen wäre die Untersuchung des Content-Delivery-Networks und der Windows P2P Funktionalität eine mögliches Ziel.

Der Trend zur Nutzung von App Stores, in denen sämtliche Software gebündelt verwaltet wird, verschiebt die wahrnehmbare Linie zwischen Systemsoftware des Betriebssystemherstellers und Software anderer Hersteller. Das Vertrauen der Nutzer verlagert sich demnach vom Softwarehersteller zum App Store Betreiber. Die Sicherheit der Softwarepakete im App Store kann jedoch nur begrenzt vom Betreiber gewährleistet werden. Hier muss also im Rahmen von Vorgaben und automatischen Prüfungen ein Sicherheitsniveau durchgesetzt werden.

Eine weitere aktuelle Entwicklung im Bereich der Betriebssysteme ist die stark steigende Anzahl an nicht Desktop-Systemen wie beispielsweise Mobil- und Internet-of-Things Geräten. Auch und gerade in diesen Systemen werden Update-Mechanismen benötigt, da aufgrund der Lebensdauer und Einsatzzeit ein erhöhter Wartungsbedarf besteht. Hierfür werden angepasste Updatekonzepte benötigt (vgl. Beyerstedt, 2017). In diesen Bereichen besteht starker Bedarf, nach der Untersuchung der verwendeten Verfahren.

Neue Update-Mechanismen sollten die in dieser Arbeit aufgezeigten Probleme bei der Erfüllung der Schutzziele berücksichtigen und durchgehend Transportverschlüsselung sowie sichere Algorithmen verwenden.

Tabellenverzeichnis

Abbildungsverzeichnis

Auflistungsverzeichnis

Literaturverzeichnis

[Apple 2017] Apple: *Code Signing Guide - Code Signing Requirement Language*. 2017. – URL https://developer.apple.com/library/content/documentation/Security/Conceptual/CodeSigningGuide/RequirementLang/RequirementLang.html. – Zugriffsdatum: 2017-09-18

[Baun 2017] Baun, Christian: *Betriebssysteme Kompakt*. Berlin : Springer Vieweg, 2017

[Beyerstedt 2017] Beyerstedt, Jannik: *Sichere und robuste Firmware-Updates von IoT-Geräten*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2017

[Bott u. a. 2017] Bott, Ed ; Siechert, Carl ; Stinson, Craig: *Windows 10 für Experten*. Heidelberg : dpunkt.verlag, 2017

[Brause 2017] Brause, Rüdiger: *Betriebssysteme - Grundlagen und Konzepte*. Berlin : Springer Vieweg, 2017

[Brumley u. a. 2008] Brumley, David ; Poosankam, Pongsin ; Song, Dawn ; Zheng, Jiang: Automatic Patch-Based Exploit Generation is Possible: Techniques and Implications. In: *2008 IEEE Symposium on Security and Privacy,*, URL http://www.truststc.org/pubs/381.html, April 2008

[Bundesamt für Sicherheit in der Informationstechnik 2016] Bundesamt für Sicherheit in der Informationstechnik: *BSI-Grundschutz Katalog*. 2016. – URL http://www.bsi.de/gshb/deutsch/index.htm. – Zugriffsdatum: 2017-09-22

[Dustdar u. a. 2013] Dustdar, S. ; Gall, H. ; Hauswirth, M.: *Software-Architekturen für Verteilte Systeme: Prinzipien, Bausteine und Standardarchitekturen für moderne Software*. Springer Berlin Heidelberg, 2013 (Xpert.press). – URL https://books.google.de/books?id=WS_3BQAAQBAJ. – ISBN 9783642555992

[Eckert 2014] Eckert, Claudia: *IT-Sicherheit : Konzepte - Verfahren - Protokolle*. München : De Gruyter Oldenbourg, 2014

[Europäische Union 2016] Europäische Union: *EU-Datenschutz-Grundverordnung*. URL http://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32016R0679. – Zugriffsdatum: 2017-09-21, 2016

[Gartner 2017] Gartner: *Gartner Says Worldwide PC Shipments Declined 4.3 Percent in Second Quarter of 2017*. 2017. – URL http://www.gartner.com/newsroom/id/3759964. – Zugriffsdatum: 2017-09-16

[Gkantsidis u. a. 2006] Gkantsidis, Christos ; Karagiannis, Thomas ; VojnoviC, Milan: Planet Scale Software Updates. In: *SIGCOMM Comput. Commun. Rev.* 36 (2006), August, Nr. 4, S. 423–434. – URL http://doi.acm.org/10.1145/1151659.1159961. – ISSN 0146-4833

[Glaeßner 2002] Glaessner, Gert-Joachim: Sicherheit und Freiheit. In: *Verwundbarkeit hochindustrieller Gesellschaften - Innere Sicherheit - Demokratie*. 2002

[Hohmann 2003] Hohmann, L.: *Beyond Software Architecture: Creating and Sustaining Winning Solutions*. Pearson Education, 2003 (Addison-Wesley Signature Series (Fowler)). – URL https://books.google.de/books?id=0slJ8zynjCEC. – ISBN 9780132465946

[ISO/IEC 2014] ISO/IEC: *ISO/IEC 27000 Information technology — Security techniques — Information security management systems — Overview and vocabulary*. 2014. – URL https://www.iso.org/committee/45306/x/catalogue/. – Zugriffsdatum: 2017-09-20

[Königs 2017] Königs, Hans-Peter: *IT-Risikomanagement mit System*. Wiesbaden : Springer Fachmedien, 2017

[Lenhard 2017] Lenhard, Thomas H.: *Datensicherheit - Technische und organisatorische Schutzmaßnahmen*. Wiesbaden : Springer Fachmedien, 2017

[Leopold u. a. 2015] Leopold, Helmut ; Bleier, Thomas ; Skopik, Florian: *Cyber Attack Information System - Erfahrungen und Erkenntnisse aus der IKT-Sicherheitsforschung*. Xpert.press, 2015. – ISBN 9783662443057

[Maraia 2005] Maraia, Vincent: *The Build Master: Microsoft's Software Configuration Management Best Practices*. Addison-Wesley Professional, 2005. – ISBN 0321332059

[Microsoft 2017a] Microsoft: *Description of the standard terminology that is used to describe Microsoft software updates*. 2017. – URL https://support.microsoft.com/en-us/help/824684/description-of-the-standard-terminology-that-is-used-to-describe-micro. – Zugriffsdatum: 2017-09-20

[Microsoft 2017b] Microsoft: *Microsoft-Software-Lizenzbestimmungen, Windows-Betriebssystem.* 2017. – URL https://www.microsoft.com/en-us/Useterms/OEM/Windows/10/UseTerms_OEM_Windows_10_German.htm. – Zugriffsdatum: 2017-09-20

[Microsoft Corporation 2017] Microsoft Corporation: *Guidance to SHA-1 Hashing Algorithm Deprecation for the Microsoft Trusted Root Certificate Program.* 2017. – URL https://social.technet.microsoft.com/wiki/contents/articles/32288.windows-enforcement-of-sha1-certificates.aspx. – Zugriffsdatum: 2017-09-18

[National Institute of Standards and Technology 2004] National Institute of Standards and Technology: *Fips Pub 199.* 2004. – URL http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.199.pdf. – Zugriffsdatum: 2017-09-17

[netmarketshare 2017] netmarketshare: *Desktop Operating System Market Share.* 2017. – URL https://www.netmarketshare.com/operating-system-market-share.aspx. – Zugriffsdatum: 2017-09-16

[Newsroom 2010] Newsroom, Apple: *Press Release - Apple's Mac App Store to Open on January 6.* 2010. – URL https://www.apple.com/newsroom/2010/12/16Apples-Mac-App-Store-to-Open-on-January-6/. – Zugriffsdatum: 2017-09-20

[NIST 2008] NIST: *Security Considerations in the System Development Life Cycle.* 2008. – URL http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-64r2.pdf. – Zugriffsdatum: 2017-09-16

[Oliveria 2006] Oliveria, Paul: *Patch Tuesday… Exploit Wednesday.* 2006. – URL http://blog.trendmicro.com/trendlabs-security-intelligence/patch-tuesday-exploit-wednesday/. – Zugriffsdatum: 2017-09-15

[Open Source Initiative 2017] Open Source Initiative: *The Open Source Definition.* 2017. – URL https://opensource.org/osd. – Zugriffsdatum: 2017-09-18

[Rechberger 2010] Rechberger, Christian: *Second-Preimage Analysis of Reduced SHA-1.* S. 104–116. In: Steinfeld, Ron (Hrsg.) ; Hawkes, Philip (Hrsg.): *Information Security and Privacy: 15th Australasian Conference, ACISP 2010, Sydney, Australia, July 5-7, 2010. Proceedings.* Berlin, Heidelberg : Springer Berlin Heidelberg, 2010. – URL https://doi.org/10.1007/978-3-642-14081-5_7. – ISBN 978-3-642-14081-5

[Rohr 2015] Rohr, Matthias: *Sicherheit von Webanwendungen in der Praxis.* Wiesbaden : Springer Fachmedien, 2015

[Sorge u. a. 2013] Sorge, Christoph ; Gruschka, Nils ; lacono, Luigi L.: *Sicherheit in Kommunikationsnetzen.* München : Oldenbourg Verlag, 2013

[Stevens u. a. 2017] Stevens, Marc ; Bursztein, Elie ; Karpman, Pierre ; Albertini, Ange ; Markov, Yarik: *The first collision for full SHA-1.* 2017. – URL https://shattered.io/static/shattered.pdf. – Zugriffsdatum: 2017-09-18

[Tanenbaum und van Steen 2008] Tanenbaum, Andrew S. ; Steen, Maarten van: *Verteilte Systeme - Prinzipien und Paradigmen.* München : Pearson Studium, 2008

[Tate 2013] Tate, Ryan: *Apple Just Ended the Era of Paid Operating Systems.* 2013. – URL https://www.wired.com/2013/10/apple-ends-paid-oses/. – Zugriffsdatum: 2017-09-16

[Team 2017] Team, Microsoft Windows S.: *Windows Server Blog - Why WSUS and SCCM managed clients are reaching out to Microsoft Online.* 2017. – URL https://blogs.technet.microsoft.com/windowsserver/2017/01/09/why-wsus-and-sccm-managed-clients-are-reaching-out-to-microsoft-online/. – Zugriffsdatum: 2017-09-20

[Thomas M. Thomas and Donald Stoddard 2012] Thomas M. Thomas and Donald Stoddard: *Network Security First-step.* Cisco Press, 2012

[Walkinshaw 2017] Walkinshaw, Neil: *Software Quality Assurance: Consistency in the Face of Complexity and Change.* Cham : Springer Natur, 2017

[Wolfgang Osterhage 2009] Wolfgang Osterhage: *Abnahme komplexer Software-Systeme.* Xpert.press, 2009

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 28. September 2017 Björn Budde