FACHHOCHSCHULE WEDEL

SEMINARARBEIT

in der Fachrichtung

Wirtschaftsingenieurwesen – IT Management

Thema:

Das Qubes OS

Eingereicht von:	Sherwin Kotval Poppenbütteler Berg 102 22399 Hamburg Tel.: +49 177 50 666 36 E-Mail: sherwinkotval@gmail.com Matrikelnummer: 103530
Erarbeitet im:	1. Fachsemester
Betreuer (FH Wedel):	Prof. Dr. Michael Anders Fachhochschule Wedel Feldstraße 143 22880 Wedel Tel.: +49 4103 8048 24 E-Mail: an@fh-wedel.de

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Ort, Datum Unterschrift (Vor-und Nachname)

Inhalt

Abbildungsverzeichnis

1. Einleitung

Heutzutage ist es von besonderer Relevanz persönliche Daten vor Unbekannten geheim zu halten und sich selbst vor Angriffen aus dem In- und Ausland zu schützen.

Als Edward Snowden im Sommer 2013 enthüllte, dass Geheimdienste weltweit Überwachungs- und Spionageaktionen, sowohl im In- als auch Ausland vollziehen, erschrak dies die Gesellschaft. Niemandem war das Ausmaß der Überwachung und Datensammlung der Geheimdienste bewusst. Maßgeblich beteiligt waren der US-amerikanische Geheimdienst „National Security Agency“ (NSA) und Großbritanniens „Gouvernement Communication Headquarters“ (GCHQ) an diesen Überwachungsaktionen. Mit ihren Verbündeten, dem kanadischen, australischen und neuseeländischen Geheimdiensten, bilden sie den Kern der weltweiten Spionage, die sogenannten „Five Eyes“. Die Geheimdienste aus Deutschland, Frankreich, Belgien, Japan und Südkorea arbeiteten ebenfalls mit diesen fünf Nationen zusammen. Alle profitieren voneinander, da Informationen geteilt und erhalten wurden. Ziel ist es offiziell jegliche Form der elektronischen Kommunikation zu überwachen, um so Verdächtige zu finden, welche den Geheimdiensten bisher unbekannt waren. Ausspioniert wurden jedoch nicht nur verdächtige Gruppierungen oder Einzelpersonen, sondern auch einzelne Unternehmen und Spitzenpolitiker verschiedenster Nationen, darunter auch die deutsche Kanzlerin Angela Merkel. Gesammelt wurden beispielsweise Metadaten aus Telefongesprächen und E-Mails, Standortdaten von Mobiltelefonen und Aktivitäten auf sozialen Netzwerken. Mithilfe von Programmen wie Tempora verschaffte sich die GCHQ Zugriff auf alle Daten, welche über die transatlantischen Glasfaserkabel in die USA geschickt wurden. Die NSA hingegen nutze das Programm PRISM um Informationen von inländischen Internetriesen, wie Google, YouTube, AOL, Apple, Microsoft und Facebook zu erspähen und zu analysieren. Es gibt noch zahlreiche andere Programme und Organisationen die maßgeblich zur Datensammlung und Informationsbeschaffung solcher Geheimdienste beitragen (Beuth, 2013).

Aufgrund der Veröffentlichung von Staatsgeheimnissen und „Top Secret“ Dokumenten wurde Edward Snowden in den USA angeklagt und per Haftbefehl gesucht. Da er sich bereits vor der Veröffentlichung des Materials in Hongkong befand, forderte die USA Hongkong auf ihn auszuliefern. Nach etlichen Asyl-Anträgen in verschiedensten Ländern, schaffte es Edward Snowden ins Exil nach

Russland, wo er sich wohl heute noch befindet (Schrettl, 2017; ZEIT ONLINE GmbH, 2013).

Seit dieser Zeit sind einige neue Gesetze und Regelungen in Kraft getreten, die die Datensammlung und deren Aufbewahrung neu regeln. Unternehmen wie Google, Apple und Microsoft begannen neben den neuen Gesetzen und Regelungen außerdem ihre Verschlüsselungstechniken auszubauen, um die Daten ihrer Kunden besser zu schützen (Beuth, 2013).

Heutzutage schützen sich auch Normalverbraucher immer mehr. Das Nutzen des Tor-Netzwerkes zum Anonymen-Surfen im Internet oder der Versand verschlüsselter E-Mails sind in der allgemeinen Bevölkerung angekommen und stellen keine Seltenheit mehr da. Laut einer Statistik der Statista GmbH aus Hamburg haben sich die Nutzer des Tor-Netzwerkes allein in Deutschland zwischen November 2016 und November 2017 fast verdreifacht (Brandt, 2017).

Abbildung 1: Tor Nutzung in Deutschland 2017 (Brandt, 2017)

Neben den steigenden Möglichkeiten sich zu schützen, ist ebenfalls die Anzahl der verschiedenen Schadsoftwares, auch „Malware“ genannt, in die Höhe geschnellt. Malware sind Schadsoftware die den Computer oder das Mobilgerät des Nutzers befallen. Sie sollen den Zugriff auf persönliche Informationen, Passwörter, Konten und anderen Dingen ermöglichen, die normalerweise auf dem Computer eines Nutzers geschützt sind. Unter der Kategorie Malware fallen viele Arten der

Schadsoftware, beispielsweise Spyware, Adware, Viren, Trojaner, Würmer, Randomware, Browser-Hijacker und viele mehr (AVAST Software, o.J.).
Das AV-Test Institut registriert nach eigenen Angaben etwa 350.000 neue Schadprogramme und potentiell unerwünschte Anwendungen täglich.

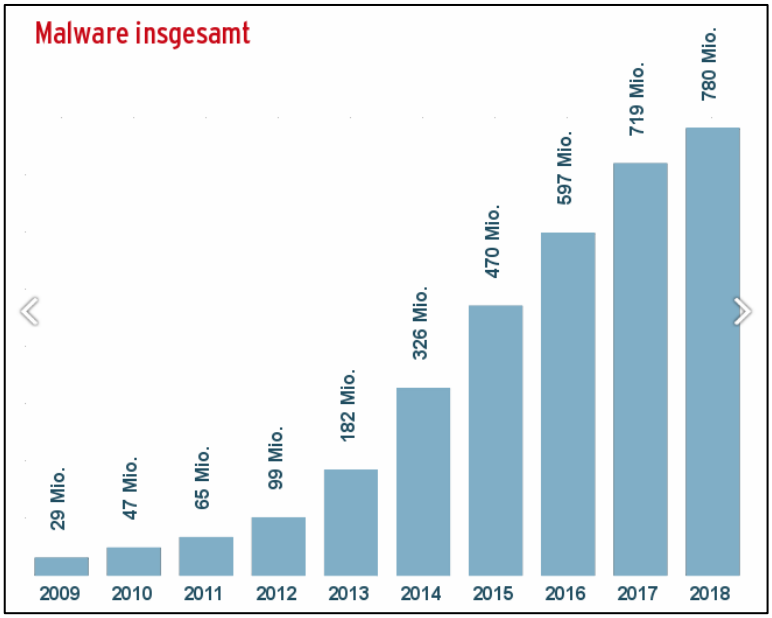

Abbildung 2: Malware insgesamt 2009-2018 (AV-TEST GmbH, 2018)

2017 waren es laut der AV-Test GmbH etwa 719 Millionen registrierte Programme für Windows Betriebssysteme, die als Malware klassifiziert werden konnten. Bis Ende 2018 sollen es etwa 780 Millionen werden. Das Ansteigen solcher Software ist nicht nur bezüglich Windows-Betriebssystemen zu beobachten. Ähnliche Anstiege lassen sich ebenso bei Android-Betriebssystemen und Mac OS finden (AV-TEST GmbH, 2018).
Um den oben genannten Faktoren wie Überwachung, Spionage und Manipulation durch Software zu entgehen, sind alternative Betriebssysteme wie das „Qubes OS“ als mögliche Lösung entwickelt worden.

2. Hintergrund

Das Qubes OS ist ein Betriebssystem ähnlich wie Microsoft Windows, Mac OS X, Android, iOS oder Tails Os. Es zeichnet sich durch eine ausgeprägte Sicherheitsorientierung von anderen Betriebssystemen ab. Diese Sicherheit gewährleistet das Qubes OS mithilfe von Virtualisierung und dem Xen Hypervisor. Das Qubes OS ist eine Open-Source Software, zertifiziert durch die „GNU GPLv3“. Die Version 1 des Betriebssystems Qubes OS erschien 2012 und die aktuellste Version 4.0 im März 2018 (The Qubes OS Project, 2018b). Um zu verstehen wie genau das Qubes OS Sicherheit für den Nutzer ermöglicht, ist es zunächst wichtig einige Aspekte zu erläutern.

2.1 GNU und GNU-GPL

Wie bereits erwähnt ist das Qubes OS von der GNU General Public Licence, auch GNU-GPL genannt, zertifiziert. Die GNU-GPL ist Teil des GNU-Projektes. Das GNU-Projekt wurde von Richard Stallman im Januar 1984 als Reaktion auf die sich immer weiter verbreitende Software-Kommerzialisierung gegründet. Jeder Rechnernutzer benötigte damals eine proprietäre Software um den Rechner überhaupt starten zu können. Das Ziel war es ein Unix-ähnliches Betriebssystem zu schaffen, welches alle Aspekte der Open-Source Idee vertritt (GNU-Projekt, 2018a).

Abbildung 3: Richard Stallman (2015) (Wikimedia Foundation Inc., 2018)

Diese Aspekte oder auch „Freiheiten“ sind Folgende: Das Programm darf für jeglichen Zweck, ob kommerziell oder privat genutzt werden (Freiheit 0). Das Programm und dessen Funktionen dürfen untersucht und studiert werden und auf eigene Bedürfnisse angepasst werden. Voraussetzung hierfür ist das mitverbreiten

des Quellcodes (Freiheit 1). Das Programm darf kopiert und erneut veröffentlicht werden (Freiheit 2). Das Programm darf verbessert werden und wieder veröffentlicht werden, um der Öffentlichkeit zu helfen. Voraussetzung ist hierbei erneut das mitverteilen des Quellcodes (Freiheit 3) (GNU-Projekt, 2018b).
Um andere Softwareentwickler ebenfalls dabei zu unterstützen freie Software zu entwickeln und zu verbreiten, schuf Richard Stallman 1989 die GNU-GPL. Diese Softwarelizenz beinhaltet die eben aufgeführten vier Freiheiten. Sie ist die bekannteste Lizenz für Open-Source Software und wurde 1991, unter GNU GPLv2, um weite Teile des Linux-Kernels und vielen weiteren alleinstehenden Open-Source-Programmen erweitert. Seit 2007 ist die aktuellste Version unter dem Namen GNU-GPLv3 veröffentlicht worden (Jaeger, o.J.).
Um eine Verständnisbasis zu schaffen wird im Folgenden zunächst auf „Virtual Machines“ eingegangen.

2.2 Virtual Machines

Um zu verstehen wie das Qubes OS funktioniert ist es wichtig zu verstehen, was Virtual Machines (VMs) sind, wie diese funktionieren und was der Hypervisor für eine Funktion erfüllt.
Computer, Laptops und Mobiltelefone werden oft mit vorinstallierten Betriebssystem verwendet. Vereinfacht beschrieben stellt ein Betriebssystem die Verbindung zwischen Hardwarekomponenten und dem Nutzer bzw. der von ihm/ihr genutzten Programme her. Moderne Hardwarekomponenten sind jedoch so leistungsstark, dass sie durch Virtualisierung mehrere Betriebssysteme gleichzeitig ausführen könnten. Eine VM ist ein Programm, welches einen eigenständigen Rechner mit eigenständigen Hardwarekomponenten simuliert (Wolski & Rup, 2017). Man könnte also sagen, dass eine VM ein Computer innerhalb eines Computers ist. Auf einer VM kann ein Betriebssystem integriert werden, welches wiederrum andere Programme wie beispielsweise Texteditoren oder Bild- und Videobearbeitungsprogramme beinhalten kann. Mehrere VMs können simultan auf einem Rechner laufen. Hierfür wird ein Hypervisor benötigt. Die VMs sind von anderen Teilen des Rechners und anderen VMs abgeschottet, sodass Programme die auf einer VM laufen keinen Zugang auf andere VMs haben (Microsoft, o.J.). Die Idee dahinter ist bereits Jahrzehnte alt.
Mitarbeiter der IBM hatten die Idee Performance Tests für neue Features durchzuführen. Dabei sollten diese Features im Idealfall sowohl im aktivierten als

auch im nicht-aktivierten Zustand auf demselben Rechner getestet werden. Dies sollte eine bessere Vergleichbarkeit realisieren. 1965 begannen also Forscher des IBM Cambridge Scientific Center an dieser Idee zu forschen. Sie entwickelten ein Schema, mit dem es möglich war, die Maschine, also den Computer, virtuell in mehrere kleine Teilstücke aufzuspalten. Diese Teilstücke sollten selbstständig, ihre zur Verfügung stehenden Hardwareressourcen verwalten, damit mehrere Einstellungen simultan getestet werden konnten (Kohlbrenner, Morris & Morris, o.J.). Erstellt und gesteuert wurden die Teilstücke durch ein Kontrollprogramm. Dieses Kontrollprogramm war der erste Hypervisor. Genutzt wurde der Hypervisor zunächst in diversen Server Betriebssystemen von IBM, unteranderem CP-40/CMS, CP-67/CMS. 1972 wurde der Hypervisor in die VM/370 Software integriert. Dies war die erste proprietäre VM-Software für Serverrechner der IBM. Integriert wurde diese Software auf den S/370 Servern. Die Kombination aus Server und Betriebssystem wurde an Universitäten und Unternehmen verkauft, da sich mithilfe dieser Technologie mehrere Nutzer, gleichzeitig auf einem Server, einwählen und arbeiten konnten ohne andere Nutzer zu beeinträchtigen oder zu beeinflussen. IBMs Nachfolger namens z/VM wurde in den 80er Jahren veröffentlicht und konnte nicht nur als Hypervisor sondern auch als alleinstehendes Betriebssystem verwendet werden (Bitner & Greenlee, 2012). Mittlerweile gibt es unzählige VM-Software-Anbieter, sowohl proprietär also auch Open-Source. Die bekanntesten auf diesem Gebiet sind *VMware*, *Oracle VM Virtualbox*, *Parallels*, *QEMU*, *Windows Virtual PC* und viele mehr (Fitzpatrick, 2010; Orgera, 2018).

2.3 Hypervisor

Wie bereits erwähnt wird ein Hypervisor benötigt, um auf der einen Seite Anfragen der VMs an die Hardware weiterzuleiten und auf der anderen Seite die VMs zu erstellen bzw. einzurichten und die Hardware-Ressourcenaufteilung zu verwalten. Der Hypervisor ist also der Verwalter der Virtualisierungsebene. Mithilfe des Hypervisors können außerdem mehrere VMs zur selben Zeit betrieben werden. Ein sehr großer Vorteil ist außerdem, dass auf den verschiedenen VMs auch unterschiedliche Betriebssysteme Anwendung finden können. Weitere Vorteile sind Isolation der VMs untereinander, das leichte Nutzen von Backup-VMs und solchen VMs, die nur kurzzeitig verwendet werden und nach der Verwendung wieder in den Ausgangszustand zurückgesetzt werden (Rouse, 2013).

Es wird zwischen zwei Hypervisor-Klassen differenziert, dem Typ-I-Hypervisor und Typ-II-Hypervisor.

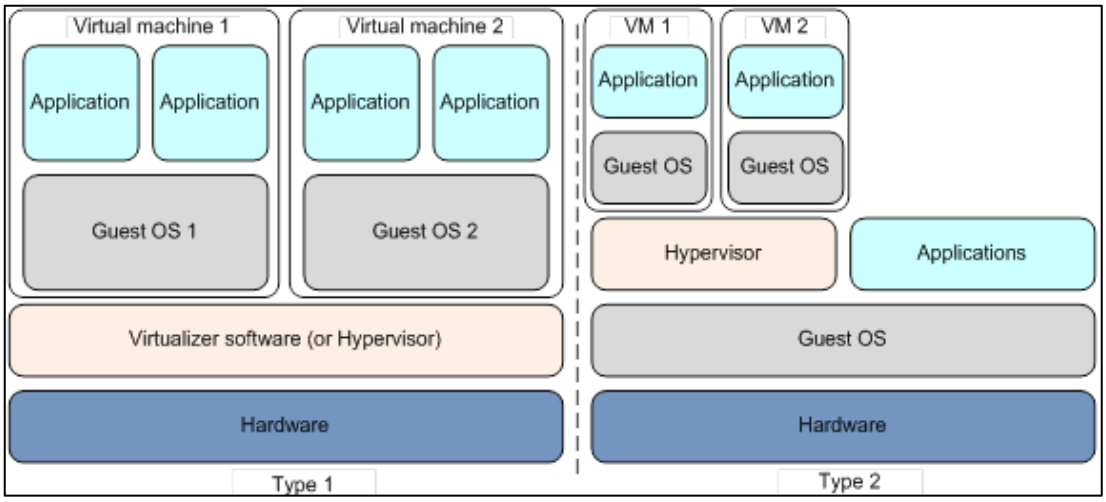

Abbildung 4: Hypervisor Klassen (Arm Limited, o.J.)

Der Typ-I-Hypervisor (Abbildung 4, links) bildet das Bindeglied zwischen der Hardware und den VMs. Hierbei bedarf es keinem vorinstallierten Betriebssystem auf dem Rechner. Auf den VMs werden anschließend Betriebssysteme installiert um Applikationen auszuführen. Es ist jedoch wichtig, dass der Hypervisor die nötigen Treiber integriert hat, damit die Kommunikation mit der Hardware einwandfrei funktioniert (Arm Limited, o.J.; Lipinski, 2014).

Um einen Typ-II-Hypervisor (Abbildung 4, rechts) zu nutzen bedarf es einem vorinstallierten Host-Betriebssystem. Auf dieses wird der Hypervisor installiert, der wiederrum die darüber liegenden VMs steuert. Auf dem Host-Betriebssystem können parallel zu den VMs auch eigene Applikationen laufen. Ein Anwendungsbeispiel wäre es Windows als Host-Betriebssystem zu nutzen, und Microsoft Virtual PC als Typ-II-Hypervisor (Arm Limited, o.J.; Lipinski, 2014).

Für das Qubes OS, wird ein Typ-I-Hypervisor verwendet. Die Hypervisor Software stammt aus dem Hause Xen und ist bereits im Qubes OS integriert.

2.4 Xen

Einleitend zu diesem Kapitel, wird zunächst auf die Geschichte des Xen-Projekts eingegangen und anschließend auf den Xen-Hypervisor.

Abbildung 5: Ian Pratt (Pratt, 2018)

Das Xen-Projekt entstand aus dem etwa 1990 gestarteten Forschungsprojekt XenoServer an der University of Cambridge. Maßgeblich beteiligt waren Ian Pratt und Simon Crosby (Xen Project, 2013). Der XenoServer war als weltweit nutzbarer Server gedacht, der den Code von Software-Entwicklern ausführen und testen sollte. Jeder Nutzer des Servers sollte seine eigenen Kapazitäten erhalten (Akritidis, 2008).
Die Nutzer-Systeme sollten außerdem voneinander getrennt sein, damit kein auszuführender Code einen anderen Nutzer schaden oder beeinträchtigen kann. Damit das Konzept sich wirtschaftlich trägt, sollten die Entwickler für die zum ausführen des Codes verwendeten Ressourcen zahlen. Aus diesem Projekt entwickelte sich der Xen-Hypervisor. Der Quellcode zum Xen-Hypervisor wurde 2002 veröffentlicht, um Hilfe von anderen Entwicklern zu erhalten. Das Ziel war es den Hypervisor effizienter und effektiver zu gestalten. 2004 wurde die erste Version von Xen veröffentlicht: Xen 1.0. Im selben Jahr wurde Xen 2.0 veröffentlicht und im selben Zuge ein Unternehmen namens XenSource, Inc. um das Projekt herum gegründet. Drei Jahre später wurde das Unternehmen von der Citrix Systems, Inc. für 500 Millionen US Dollar erworben. 2010 wurde Xen 4.0 veröffentlicht und 2011 Xen für Cloud-Computing. Im Jahr 2013 wurde Xen für mobile Geräte mit ARM Prozessoren veröffentlicht, was es ermöglichte Xen auch auf Laptops zu nutzen. Die aktuellste Version ist Xen 4.10.1. Diese erschien im Mai 2018 (Xen Project, 2013, Xen Project, 2017).

Für diese Arbeit von besonderer Bedeutung ist der Xen-Hypervisor. Der Xen-Hypervisor ist ein Typ-1-Hypervisor. Wie bereits beschrieben wurde, stellt ein Typ-1-Hypervisor das Bindeglied zwischen Hardware und VMs dar. Der Hypervisor erlaubt den Betrieb mehrerer Gastsysteme auf einem Hostsystem, also das Nutzen mehrerer VMs auf einem Rechner. Außerdem teilt der Hypervisor die Ressourcen der VMs ein. Dies bedeutet, dass der Hypervisor jeder VM einen bestimmten Anteil

an Hardware Ressourcen zuweist, sodass jedes System für sich arbeiten kann, ohne von anderen Systemen gestört oder verlangsamt zu werden. Der VM wird also gestattet über eigene Prozessoren und Festplatten zu verfügen. Der Xen-Hypervisor sowie der XenServer und weitere Komponenten von Xen sind Open-Source und mit der GPLv2 lizensiert. Seit einiger Zeit bietet Xen zahlreiche Funktionen für Unternehmen an und wird vor allem auch im Cloud-Computing Umfeld immer beliebter (Xen Project, 2013, Xen Project, 2017).

Nun, da alle nötigen Informationen vorliegen, wird im folgenden Kapitel das Qubes OS vorstellt. Als erstes werden ein Einrichtungsbeispiel und die Konfigurationen vorgestellt. Außerdem wird erklärt was für die Nutzung des Qubes OS benötigt wird und wo man das Betriebssystem herunterladen kann.

3. Qubes OS

Wie bereits erwähnt ist dieses Betriebssystem auf Sicherheit ausgelegt und erreicht dies durch Virtualisierung und Isolation der Qubes. Qubes werden im Qubes OS die VMs genannt. Die Virtualisierung wird mithilfe des Xen und dessen Hypervisors ermöglicht. Hierdurch werden die VMs nicht nur erzeugt, sondern auch gesteuert. Auf den VMs lassen sich die integrierten Linux-Distributionen Fedora, Debian oder Whonix aufspielen, ebenso ist es möglich Windows oder andere Linux-Systeme zu integrieren (The Qubes OS Project, 2018b, The Qubes OS Project, 2018c).

Abbildung 6: Joanna Rutkowska (Rutkowska, o.J.)

Joanna Rutkowska und die Mitglieder des Invisible Things Lab sind die Programmierer und Herausgeber dieses Betriebssystems. Joanna Rutkowska hat mehr als 10 Jahre Erfahrung auf dem Gebiet der Computersicherheit und arbeitete neben dem Qubes OS an verschiedenen anderen Projekten bezüglich Virtualisierung und Sicherheit mit (Rutkowska, o.J.a). Sie hat etwa 14 Artikel und wissenschaftliche Paper zu diesen Themen veröffentlicht, unter anderem auch über „Rootkits“ und „Stealth Malware“ (Rutkowska, o.J.b).

3.1 Struktur

Die Sicherheit im Qubes OS wird mithilfe von VMs gewährleistet. Diese werden Qubes genannt, sind voneinander unabhängig. Sie sind in keiner Art und Weise mit einander verbunden, es sei denn der Nutzer möchte es so. Der dahinterliegende Sinn ist, sobald eines der Qubes durch Schadsoftware infiziert sein sollte, diese nur Daten aus derselben Qube befallen können und keinen Zugang zu anderen Qubes und deren Daten haben. Außerdem können Qubes nicht nur die Hardware-Ressourcen einteilen, sondern einzelnen VMs, auch Domains genannt den Zugriff auf Hardwarekomponenten untersagen oder erlauben. Dies kann mithilfe von sogenannten Zwischendomains ermöglicht werden. So kann also beispielsweise einer Domain der Zugang ins Internet erlaubt werden, während einer anderen Domain dieser Zugang verwehrt wird. Sicherheit wird außerdem durch das Verschlüsseln des Datenträgers gewährleistet. Dies geschieht durch die integrierte

LUKS (Linux Unified Key Setup) Software. Luks ist eine Erweiterung des für Linux häufig genutzten dm-crypt (Invisible Things Lab, 2018; Thoma, 2014). Wie genau die zugrunde liegende Verschlüsselung von LUKS funktioniert, würde den Rahmen dieser Arbeit übersteigen und ist darüber hinaus auch kein Gegenstand des Seminars.
Das nachfolgende Bild stellt die Struktur der Qubes, des Xen Hypervisor und der Zwischendomains dar.

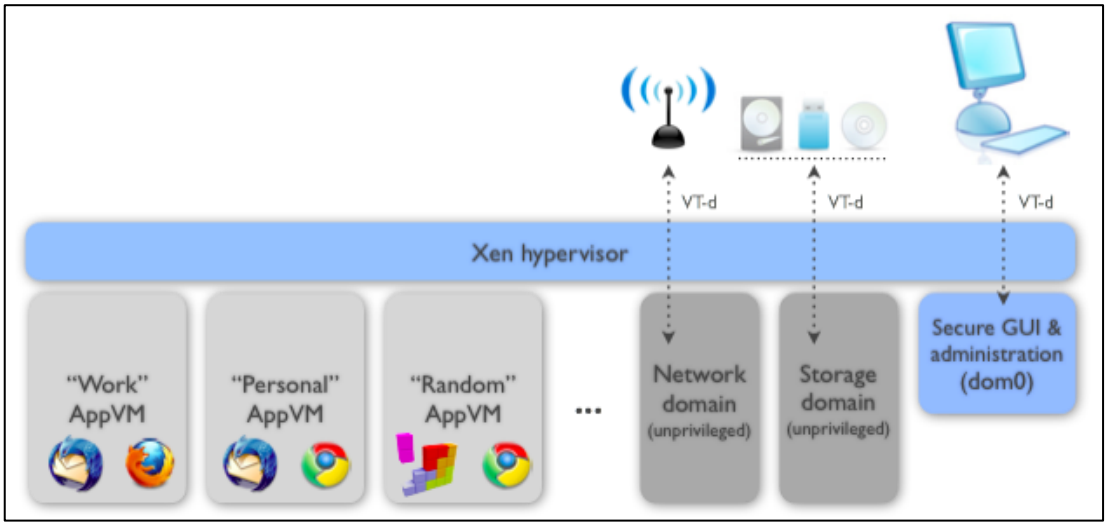

Abbildung 7: Qubes OS – Struktur (The Qubes OS Project, 2018a)

Die Qubes, die vom Nutzer aktiv verwendet werden, werden hier *AppVM* genannt. Diese AppVMs sind für drei verschiedene Zwecke angelegt - *Work*, *Personal* und *Random*. Die eben beschriebenen Zwischendomains können auf der rechten Seite betrachtet werden. Die Zwischendomain *Network domain* ist, wie der Name schon sagt, für die Netzwerkverbindung zuständig, während die *Storage domain* den Zugriff auf die Speichermedien ermöglicht. Ganz rechts sieht man eine Domain namens *Secure domain & administration (dom0).* Diese Domain ist die Administrationsdomain. Hierrüber lassen sich alle Einstellungen und VMs vom Nutzer verwalten (The Qubes OS Project, 2018a).

Angenommen es sollen hoch sensible arbeitsbezogene Informationen auf einer Domain gespeichert und abgerufen werden können. Diese Domain wird *Work* genannt. Dieser Arbeitsbereich soll ermöglichen, dass Excel Listen und Text Dateien gelesen, analysieren und gegebenenfalls bearbeitet werden können. Für solche Aufgaben benötigt die Domain keinen Internetzugang. Außerdem könnten diese Daten sehr sensibel sein und sollten nicht von schädlicher Software

kompromittiert oder von außen gelesen werden können. Über die Dom0 kann der Nutzer nun für diese Domain den Zugang zum Internet oder zum Netzwerk im Allgemeinen sperren lassen. Neben der Work-Domain wird zudem eine Domain gebraucht, die für den täglichen Gebrauch verwenden werden kann. Mit täglichem Gebrauch sind Funktionalitäten wie Onlineshopping, das Abrufen von privaten E-Mails oder Videostreaming gemeint. Für diese Domain wird also Internet benötigt. Angelehnt an das obige Bild könnte diese Domain *Random* heißen. Domains können parallel genutzt werden. Angezeigt werden diese Domains ähnlich wie bei Windows in mehreren Fenstern. Damit ist gemeint, dass der Nutzer einen Anzeigebildschirm hat, den sich die aktiven AppVMs teilen (The Qubes OS Project, 2018a). Angezeigt werden die Domains immer mit der vorher ausgewählten Farbe auf der Titelleiste, sodass die Unterscheidung zwischen den Vertrauensstufen auch Visuell dargestellt wird (vgl. Abbildung 8 und Abbildung 13).
Bevor die AppVMs jedoch nutzbar sind, müssen sie zunächst erstellt bzw. konfiguriert werden. Dies geschieht über die Dom0.

3.2 Administration

Die Abbildung 8 illustriert ein Beispiel für die Dom0. In der ersten Zeile ist deutlich zu erkennen, dass die Dom0 die AdminVM ist. Außerdem sieht man in einigen Zeilen Schlosssymbole mit schwarzer Farbe, die in der Spalte *Template* die Beschriftung *TemplateVM* aufweisen. Diese TemplateVMs werden genutzt um andere VMs zu erstellen. Da diese TemplateVMs bereits im Vorfeld stabil laufen, ist sichergestellt, dass dort alle vorinstallierten Applikationen und Einstellungen problemlos übernommen werden können. Dies kann gemacht werden, indem eine solche TemplateVM geklont wird. Es gibt zudem nicht-stabile TemplateVMs, bei welches es sich um Testumgebung des Nutzers handelt. Außerdem sieht man auf diesem Bild Domains mit grüner, gelber, roter und violetter Farbe. Wie bereits erwähnt werden diese Farben vom Nutzer eingestellt und unterstützen den Vertrauensunterschied zwischen den Domains optisch.

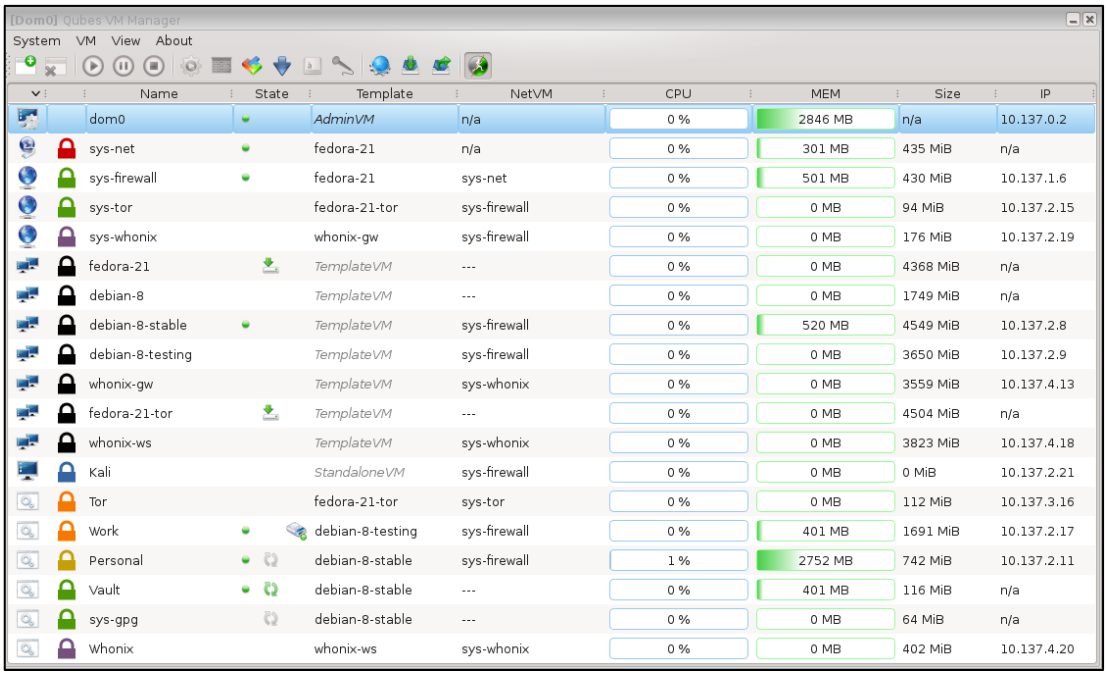

Abbildung 8: Dom0 (Ruether, 2015)

Am Beispiel einer VM, die zum Surfen im Tor-Browser bestimmt ist, wird im Folgenden verdeutlicht, wie eine VM aus einem Template „geklont“ werden kann und wie diese an die eigenen Bedürfnisse angepasst und letzten Endes genutzt werden kann.

Wie auf Abbildung 8 zu sehen ist, ist die *sysnet* Domain rot dargestellt. Sie hat direkten Zugang zur Netzwerkkarte und bildet somit das Bindeglied zum Internet für andere VMs. Damit bei der Nutzung des Internets keinem Außenstehenden die Möglichkeit geboten wird auf das System zugreifen zu können, wird eine weitere Zwischendomain benötigt. Diese weitere Zwischendomain nennt sich in Fall *sys-firewall* und hat eine grüne Farbe, da der Firewall vertraut wird. Außerdem wird noch eine Zwischendomain benötigt, die den Proxy zu Tor darstellt. Hier im Beispiel ist diese Domain grün dargestellt und nennt sich *sys-Tor*. Zu guter Letzt wird die AppVM, die den Tor-Browser tatsächlich öffnet benötigt. Diese Domain nennt sich *Tor* und ist orange dargestellt. Zusätzlich zu den Farben, sieht man in der Abbildung 8 zu jeder der eben genannten Domains (bis auf die sys-net Domain) einen Eintrag in der NetVM spalte. Dies zeigt die Beziehung zwischen den VMs an bzw. den Verbindungsweg (Ruether, 2015). In der folgenden Abbildung 9 wird diese Beziehung anschaulich dargestellt.

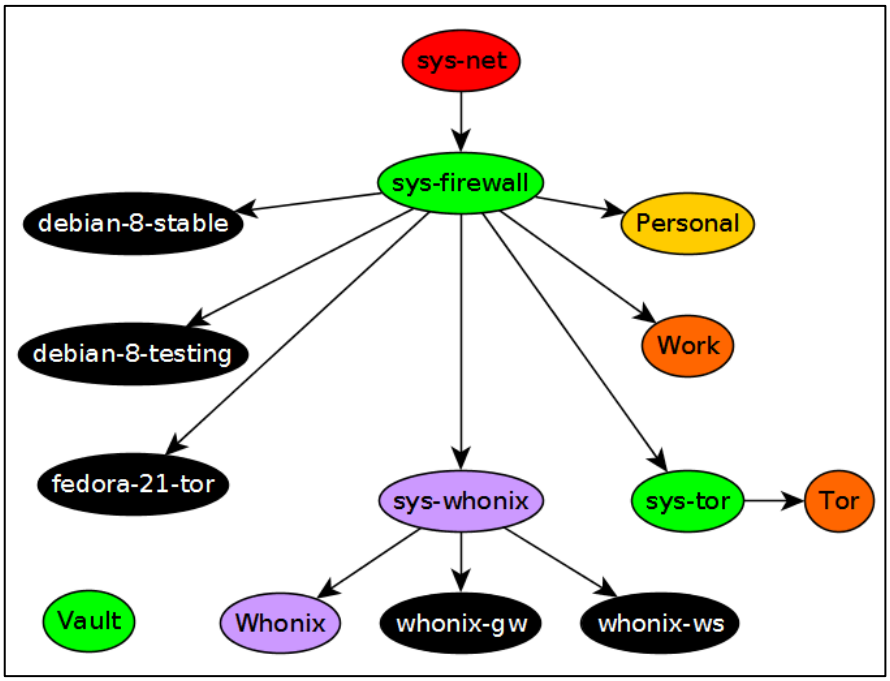

Abbildung 9: VM Verbindungswege (Ruether, 2015)

In dieser Abbildung wird außerdem verdeutlicht, dass die Domain *Vault (*auf Deutsch *Tresor)* mit keiner anderen VM verbunden ist. Diese Vertrauenswürdige (grüne) Domain kann beispielsweise für das Speichern von Passwörtern und sehr privaten Bildern genutzt werden. Die „Tresor“ Domain benötigt also keine Internetverbindung oder ähnliches (Ruether, 2015).
Die nächsten Abschnitte stellen dar, wie die VM-Verkettung zu erstellen ist. Es wird davon ausgegangen, dass die *sys-net* und *sys-firewall* bereits erstellt wurden.

3.3 Einrichtungsbeispiel Tor VM

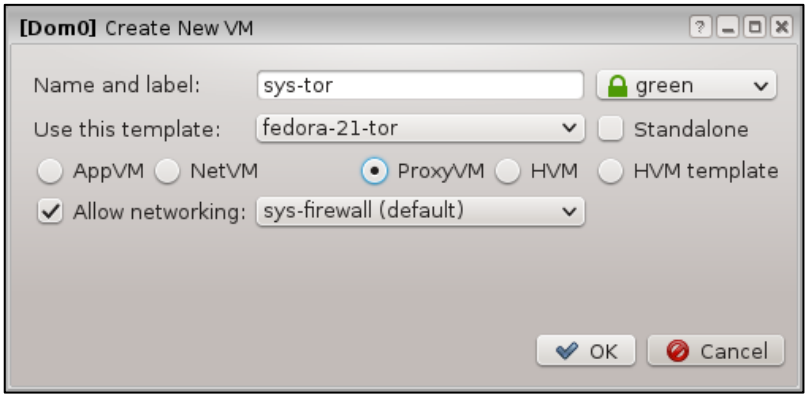

Abbildung 10: Tor ProxyVM (Ruether, 2015)

Hier sieht man das Fenster *Create New VM*. Außerdem kann man am linken oberen Rand sehen, dass dieses Fenster von Dom0 erstellt wurde. In dieser Maske kann

ein Name und eine Farbe für die neue Domain vergeben werden. Des Weiteren kann dem System mitgeteilt werden, welches Template zur Erstellung genutzt werden soll. Man kann anwählen, dass es sich hierbei um eine ProxyVM handelt, die Verbindung zum Netzwerk benötigt. Dafür wird als VerbindungsVM die *sys-firewall* gewählt. Nun wird in etwa dasselbe für die *Tor* VM gemacht. Der Unterschied liegt hierbei zunächst in der Farbe und in der Art der VM. Die Art der VM ist in diesem Fall keine ProxyVM sondern eine AppVM. Die Farbe sollte orange sein, da diese VM über den Tor Browser Zugang ins Internet hat und ihr deshalb nicht gänzlich vertraut werden kann. Nun geht man in die Einstellungen der VM und stellt die NetVM ein (Ruether, 2015).

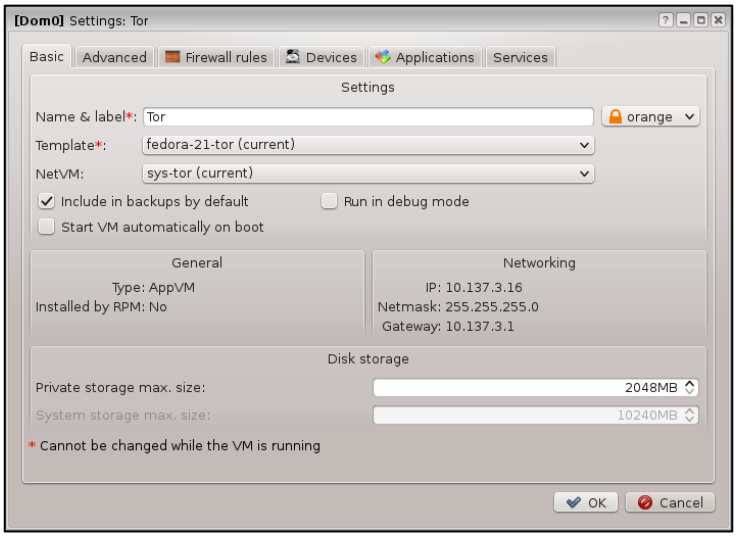

Abbildung 11: Tor VM (Ruether, 2015)

Wie man hier sieht ist als NetVM die eben erstellte VM sys-tor angegeben. Die angegebene Farbe ist orange und das genutzte Template hier ebenfalls „fedora-21-tor“. Außerdem kann in diesem Fenster die maximale Speicherkapazität der VM festlegt werden. Im Reiter *Applications* können nun dieser VM weitere Applikationen und Rechte zur Nutzung von Funktionen zugesprochen werden (Ruether, 2015).
Im nachfolgenden Bild wird beispielhaft für die VM *personal* verdeutlicht, welche Funktionen und Applikationen für VMs freigegeben werden können.

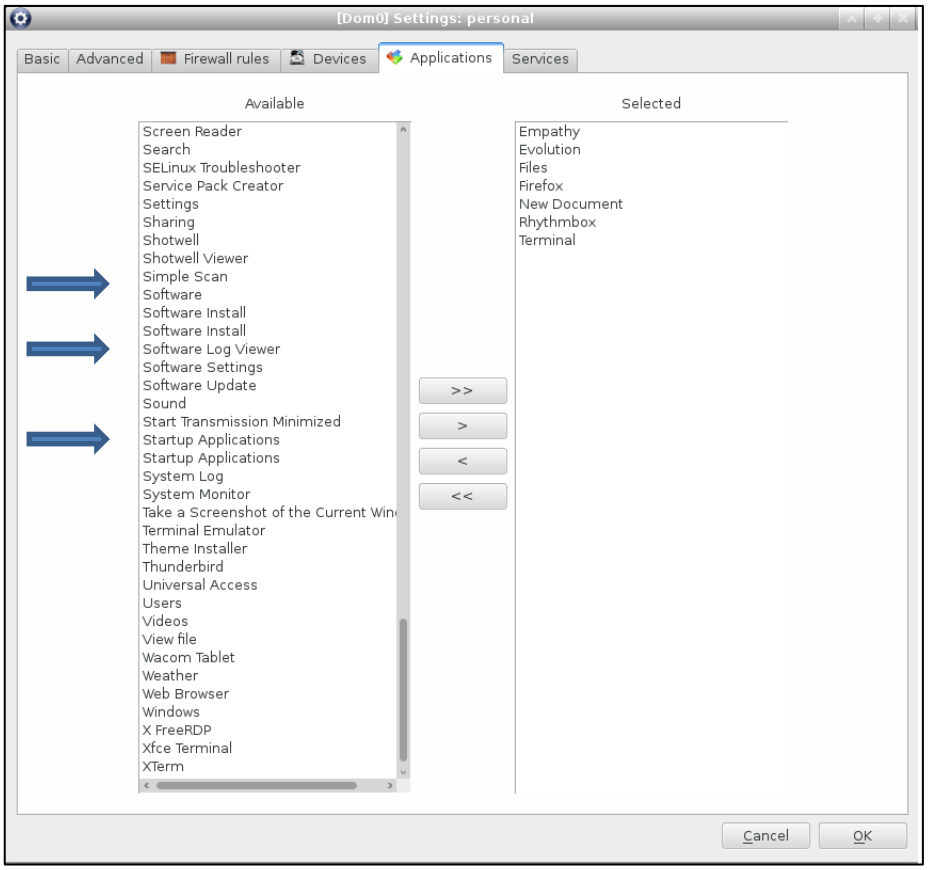

Abbildung 12: Domain Applikations-settings (Thoma, 2014)

Funktionen können unteranderem, wie links zu sehen ist, die Erlaubnis sein, Software zu installieren, Sound von Musik oder Videos wiederzugeben oder einen Screenshot des Bildschirms zu machen. Ebenfalls Funktionen wie über eine Zwischenablage zu verfügen um Texte innerhalb einer Domain zu Kopieren können so eingestellt werden. In diesem Beispiel sieht man außerdem auf der rechten Seite, dass Applikationen wie Firefox und das Dateisystem (hier *files*) für diese Domain freigegeben wurde (Thoma, 2014).

3.4 Nutzeransicht

Die nachfolgenden Bilder stellen dar, wie der Nutzer das Qubes OS wahrnimmt und wie sich die verschiedenen VMs parallel auf einem Bildschirm anzeigen lassen. Die Farbgebung und die Benennung der Domains sind hier anders als in dem Beispiel zuvor.

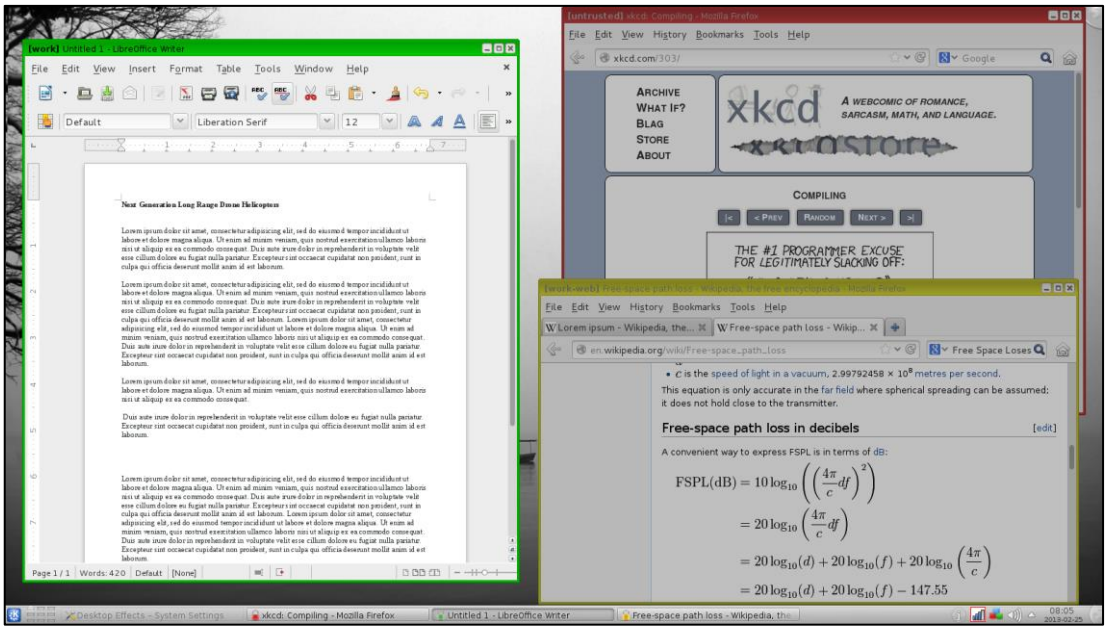

Abbildung 13: Nutzer Ansicht (The Qubes OS Project, 2013)

In diesem Beispiel ist auf der linken Seite die *Work* Domain zu erkennen. Sie wird mit grüner Farbe untermalt, was anzeigt, dass dieser Domain vollkommen vertraut wird. Anders ist es bei der rot untermalten *untrusted* Domain, diese wird für das normale Surfen im Internet, dem Nachrichten lesen oder ähnliches verwendet und ist auf der rechten Seite im oberen Bereich zu finden. Darunter angezeigt wird die gelbe Domain *Work-Web*. Hier wird zwar auch im Internet gesurft, jedoch wird nur auf arbeitsbezogenen Seiten verkehrt. Solche arbeitsbezogenen Seiten, wie beispielsweise Stackoverflow oder Github, stellen meist nur ein geringes Sicherheitsrisiko dar. Apps, die in unterschiedlichen Domains gestartet werden, haben nur Zugriff auf die Domain, in der sie gestartet wurden. Genauso verhält es sich mit dem Speicher. Eine App kann nur auf das Dateisystem der selbigen VM zugreifen, nur von dort aus Dateien öffnen und nur dort abspeichern (The Qubes OS Project, 2013).

Abbildung 14: Domain Benachrichtigung (The Qubes OS Project, 2013)

Es ist für den Nutzer immer ersichtlich aus welcher Domain ein neu geöffnetes Fenster stammt. Hier wird wie ein Fenster dargestellt, welches sich im Vordergrund geöffnet hat. Diese Meldung verlangt eine Passphrase. Obwohl direkt dahinter ein Fenster mit roter Domainüberschrift zu sehen ist, wissen wir, dass das neue Fenster zur grünen, vertrauenswürdigen Work Domain gehört und dort sorglos die Passphrase eingegeben werden kann. Die Überschriftensteuerung wird von der Dom0 verwaltet und keine der VMs bzw. anderen Domains „weiß“, dass sie nur eine VM ist. Sie weiß außerdem nicht welche Farbe ihr zugehörig ist und hat auch keine Befugnis den gesamten angezeigten Bildschirm zu übernehmen. Daher kann man mit Sicherheit sagen, dass keine VM es schafft, ob durch Schadsoftware okkupiert oder nicht, ein Fenster zu erstellen, welches genauso aussieht wie das einer vertrauenswürdigen Domain (The Qubes OS Project, 2013).

Das Kopieren und Einfügen von Dateien oder Text ist auch zwischen Domains möglich. Hierfür sorgt der Hypervisor der über eine spezielle Tastenkombination (Strg-Shift-C bzw. Strg-Shift-V) den Text oder die Datei in den Qubes OS Zwischenspeicher hinterlegt, welche anschließend von einer anderen Domain aus entnommen werden kann. Die Domains müssen für diesen Vorgang berechtigt sein und beim Einfügen wird sicherheitshalber nochmal die explizite Bestätigung des Nutzers benötigt.

Eine weitere Einzigartigkeit des Qubes OS ist das bequeme Erstellen von *DisposableVMs* (vgl Abbildung 15).

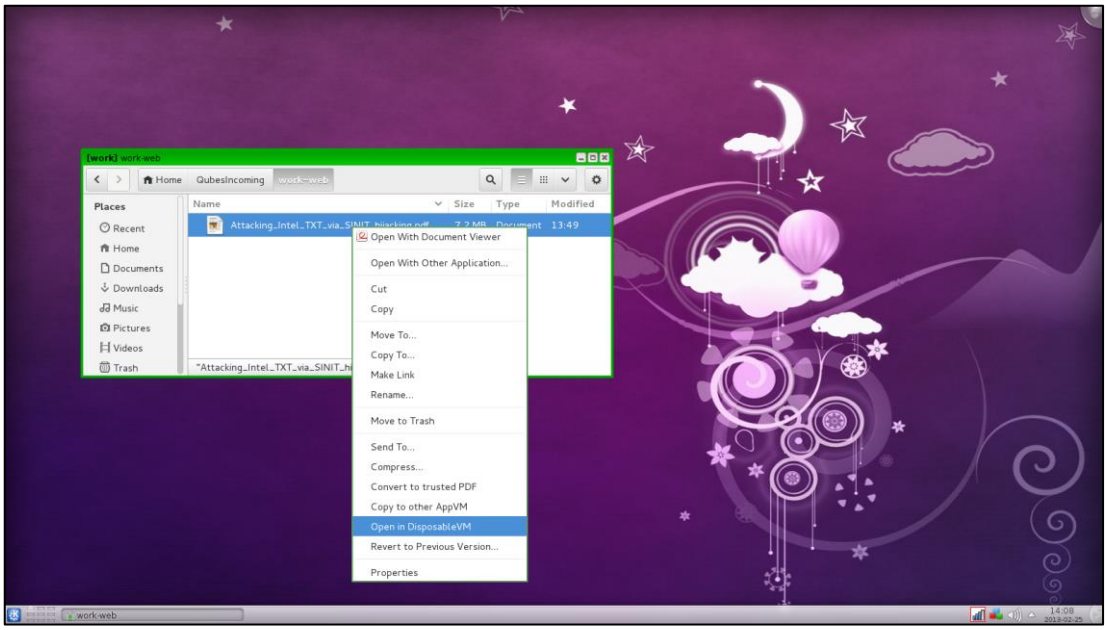

Abbildung 15: Disposable VM (The Qubes OS Project, 2013)

Eine DisposableVM ist eine VM, die für den kurzzeitigen und einmaligen Gebrauch verwendet wird. Diese wird nur für die Zeit der Nutzung generiert und wird hinterher komplett auf den Urzustand zurückgesetzt. Dies ist besonders wichtig, wenn man sich beispielsweise bei einem E-Mail Anhang nicht sicher ist, oder im Internet eine Datei oder eine Software herunter lädt die kompromittiert sein könnte. Diese Datei klickt man mit der rechten Maustaste an und wählt den Punkt *Open in DisposableVM* (vgl. Abbildung 15). Außerdem lassen sich auch bspw. Firefox oder andere Applikationen in so einer DisposableVM starten. Dies ist vor allem dann sehr praktisch, wenn man auf nicht vertrauenswürdige Seiten herumstöbert oder im Internet nach Artikel sucht die nicht gleich von Google oder Amazon als gesponserte Werbung an den Seiten normaler Internetseiten erscheinen sollen. Die DisposableVMs lassen sich in Sekunden generieren, was das nutzen dieser VM-Art sehr angenehm gestaltet (The Qubes OS Project, 2013).

Der einzige „Nachteil“ an dem Betriebssystem Qubes OS ist, dass es nur für einen Benutzer geeignet ist. Es ist nämlich bei der Installation so vorgesehen, dass nur ein Benutzerkonto angelegt wird. Dem ist so, da der einzige Nutzer auch derjenige ist, der die Dom0 verwaltet und damit jegliche Administratorrechte auf dem Rechner

hat. Weitere Nutzer stellen in diesem Kontext ein erhöhtes Risiko für die Sicherheit dar (Thoma, 2014).

3.5 Benötigte Hardware

Viel Leistung wird für die Nutzung des Qubes OS nicht verlangt. Der Rechner sollte lediglich einen Vier-Kern-Prozessor, mindestens 4GB Arbeitsspeicher und eine mindestens 32GB große Festplatte mitbringen. Empfohlen wird eine SSD, vor allem für die Verwendung von Windows. Dort soll nämlich nicht nur das Windows Betriebssystem einwandfrei laufen, sondern auch Applikationen, die auf der WindowsVM gestartet werden sollen flüssig dargestellt werden. Unerlässlich für das Qubes OS ist, eine aktivierbare Virtualisierungsfunktion im BIOS. Für Intel-Chips heißen diese entweder VT-d oder VT-x und für AMD-Chips IOMMU oder AMD-v. Vor allem für die Nutzung von Windows ist diese Einstellung notwendig (Thoma, 2014).

Herunterladen kann man das Betriebssystem auf der offiziellen Website des Herausgebers. Dort findet man neben der aktuellen Version auch die alten Versionen (The Qubes OS Project, 2018c).

4. Fazit

Durch die Art und Weise, wie dieses Betriebssystem funktioniert und eingerichtet wird sieht es zunächst sehr kompliziert aus. Die Sicherheit jedoch, die dadurch gewährleistet wird, ist momentan auf dem Markt ohne Konkurrenz. Da dieses Betriebssystem kostenlos jedem zur Verfügung steht, sollte es meiner Meinung nach häufiger verwendet werden. Die benötigte Eingewöhnungszeit ist meines Erachtens wie das Umstellen von Windows auf Mac OS X. Ich persönlich werde mir auf meinen Heim-PC das Qubes OS installieren und es nutzen. Besonders praktisch ist es für persönliche Bilder, Videos und Passwörter zu nutzen, da man solch einer Domain jegliche Rechte, außer Leserechte, entzieht und somit auf höchstem Sicherheitsniveau für diese sensiblen Daten garantieren kann.

Für besonders „paranoide“ Nutzer stellt das Invisible Things Lab Team eine Installations- und Einrichtungseinleitung zur Verfügung (Invisible Things Lab, 2011).

Selbst Edward Snowden, der wahrscheinlich sehr paranoid ist und dies auch sein darf, sagt über das Qubes OS, dass es kein vergleichbares System gibt. Wörtlich übersetzt schrieb er auf der Social Media Plattform Twitter: „Wenn dir Sicherheit sehr wichtig ist, dann ist das Qubes OS das beste zur Auswahl stehende Betriebssystem. Es ist das Betriebssystem welches ich verwende. Niemand beherrscht VM-Isolation besser.“ (Snowden, 2016)

Literaturverzeichnis

Akritidis, P. (University of Cambridge Computer Laboratory, Hrsg.). (2008). *Xenoservers. Overview*. Verfügbar unter http://www.cl.cam.ac.uk/research/srg/netos/projects/archive/xeno/

Arm Limited (Arm Limited, Hrsg.). (o.J.). *AArch64 virtualization*. Verfügbar unter https://developer.arm.com/products/architecture/a-profile/docs/100942/latest/aarch64-virtualization

AVAST Software (AVAST Software s.r.o., Hrsg.). (o.J.). *Malware*. Verfügbar unter https://www.avast.com/de-de/c-malware

AV-TEST GmbH (AV-TEST GmbH, Hrsg.). (2018). *Malware*. Verfügbar unter https://www.av-test.org/de/statistiken/malware/

Beuth, P. (ZEIT ONLINE GmbH, Hrsg.). (2013). *Alles Wichtige zum NSA-Skandal. Snowden-Enthüllungen*. Verfügbar unter https://www.zeit.de/digital/datenschutz/2013-10/hintergrund-nsa-skandal/komplettansicht

Bitner, B. & Greenlee, S. (IBM Corporation, Hrsg.). (2012). *z/VM – A Brief Review of Its 40 Year History. Virtualization Leadership born from 40 years of Experience*. Verfügbar unter http://www.vm.ibm.com/vm40hist.pdf

Brandt, M. (Statista GmbH, Hrsg.). (2017). *TOR-Nutzung steigt sprunghaft*. Verfügbar unter https://de.statista.com/infografik/11988/tor-netzwerk/#0

Fitzpatrick, J. (Univision Communications Inc., Hrsg.). (2010). *Five Best Virtual Machine Applications*. Verfügbar unter https://lifehacker.com/5714966/five-best-virtual-machine-applications

GNU-Projekt (GNU-Projekt, Hrsg.). (2018a). *Geschichte des GNU-Systems*. Verfügbar unter https://www.gnu.org/gnu/gnu-history.de.html

GNU-Projekt (GNU-Projekt, Hrsg.). (2018b). *Was ist GNU? Mehr über GNU*. Verfügbar unter https://www.gnu.org/home.de.html

Invisible Things Lab (Invisible Things Lab, Hrsg.). (2011). *Partitioning my digital life into security domains*. Verfügbar unter http://theinvisiblethings.blogspot.com/

Invisible Things Lab (Invisible Things Lab, Hrsg.). (2018). *Invisible Things Lab. Services*. Verfügbar unter https://invisiblethingslab.com/

Jaeger, D. T. (Institut für Rechtsfragen der Freien und Open Source Software (ifrOSS), Hrsg.). (o.J.). *Was ist die GPL?* Verfügbar unter http://www.ifross.org/was-gpl

Kohlbrenner, E., Morris, D. & Morris, B. (Hyperlearning Center, Hrsg.). (o.J.). *The History of Virtual Machines*. Verfügbar unter http://denninginstitute.com/itcore/virtualmachine/history.htm

Lipinski, K. (DATACOM Buchverlag GmbH, Hrsg.). (2014). *Hypervisor*. Verfügbar unter https://www.itwissen.info/Hypervisor-hypervisor.html

Microsoft (Microsoft, Hrsg.). (o.J.). *What is a virtual machine?* Verfügbar unter https://azure.microsoft.com/en-us/overview/what-is-a-virtual-machine/

Orgera, S. (lifewire.com, Hrsg.). (2018). *The 6 Best Virtual Machine Software Programs*. Verfügbar unter https://www.lifewire.com/best-virtual-machine-software-4147437

Pratt, I. (LinkedIn Ireland Unlimited Company, Hrsg.). (2018). *Ian Pratt. Co-Founder and President at Bromium*. Verfügbar unter https://uk.linkedin.com/in/ianpratt/de

Rouse, M. (TechTarget Germany GmbH, Hrsg.). (2013). *Virtuelle Maschine. Virtual Machine (VM)*. Verfügbar unter https://www.searchdatacenter.de/definition/Virtuelle-Maschine-Virtual-Machine-VM

Ruether, J. (Ruether, J., Hrsg.). (2015). *Setting Up Qubes*. Verfügbar unter http://jrruethe.github.io/blog/2015/09/12/setting-up-qubes/

Rutkowska, J. (Invisible Things Lab, Hrsg.). (o.J.a). *Invisible Things Blog OS*. Verfügbar unter https://blog.invisiblethings.org/about/

Rutkowska, J. (Invisible Things Lab, Hrsg.). (o.J.b). *Papers*. Verfügbar unter https://blog.invisiblethings.org/papers/

Schrettl, L. (news networld internetservice GmbH, Hrsg.). (2017). *Menschen des Jahres: Was wurde aus … Edward Snowden?* Verfügbar unter https://www.profil.at/shortlist/ausland/menschen-des-jahres-edward-snowden-8554751

Snowden, E. (2016). *Twitter Post*. Verfügbar unter https://twitter.com/Snowden/status/781493632293605376

The Qubes OS Project (The Qubes OS Project, Hrsg.). (2013). *Screenshots*. Verfügbar unter https://www.qubes-os.org/screenshots/

The Qubes OS Project (The Qubes OS Project, Hrsg.). (2018a). *Qubes Architecture*. Verfügbar unter https://www.qubes-os.org/doc/architecture/

The Qubes OS Project (The Qubes OS Project, Hrsg.). (2018b). *Qubes OS - Intro. An introduction to Qubes OS*. Verfügbar unter https://www.qubes-os.org/intro/

The Qubes OS Project (The Qubes OS Project, Hrsg.). (2018c). *The Qubes OS*. Verfügbar unter https://www.qubes-os.org

Thoma, J. (Golem Media GmbH, Hrsg.). (2014). *Abschottung bringt mehr Sicherheit. Qubes OS angeschaut*. Verfügbar unter https://www.golem.de/news/qubes-os-angeschaut-abschottung-bringt-mehr-sicherheit-1410-109975.html

Wikimedia Foundation Inc. (Wikimedia Foundation Inc., Hrsg.). (2018). *GNU-Projekt. Entstehung*. Verfügbar unter https://de.wikipedia.org/wiki/GNU-Projekt

Wolski, D. & Rup, M. (IDG Tech Media GmbH / PC Welt, Hrsg.). (2017). *Crashkurs: Wie funktionieren eigentlich virtuelle PCs?* Verfügbar unter https://www.pcwelt.de/ratgeber/VM-Crashkurs-Wie-funktionieren-virtuelle-PCs-9833574.html

Xen Project (Xen Project, Hrsg.). (2013). *History*. Verfügbar unter https://www.xenproject.org/about/history.html

Xen Project (wiki.Xen Project, Hrsg.). (2017). *Book/HelloXenProject/1-Chapter*. Verfügbar unter https://wiki.xenproject.org/wiki/Book/HelloXenProject/1-Chapter

ZEIT ONLINE GmbH (Hrsg.). (2013). *USA klagen Snowden wegen Spionage an*. Verfügbar unter https://www.zeit.de/politik/ausland/2013-06/prism-usa-snowden-anklage