HOCHSCHULE
MITTWEIDA
University of Applied Sciences

BACHELORARBEIT

Herr
Jonas Schneidewind

Vergleich und Bewertung ausgewählter Sandbox Systeme für die dynamische Malware Analyse

Mittweida, Januar 2023

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Vergleich und Bewertung ausgewählter Sandbox Systeme für die dynamische Malware Analyse

Autor:
Jonas Schneidewind

Studiengang:
Allgemeine und digitale Forensik

Seminargruppe:
Fo19w1-B

Erstprüfer:
Prof. Dr. rer. nat. Dirk Labudde

Zweitprüfer:
Christian Köpp, M.Sc.

Einreichung:
Mittweida, 31.01.2023

Verteidigung/Bewertung:
Mittweida, 2023

Faculty of **Applied Computer Sciences and Biosciences**

BACHELOR THESIS

Comparison and evaluation of selected sandbox systems for the dynamic malware analysis

Author:
Jonas Schneidewind

Course of Study:
General and digital forensics

Seminar Group:
Fo19w1-B

First Examiner:
Prof. Dr. rer. nat. Dirk Labudde

Second Examiner:
Christian Köpp, M.Sc.

Submission:
Mittweida, 31.01.2023

Defense/Evaluation:
Mittweida, 2023

Bibliografische Beschreibung:

Schneidewind, Jonas:
Vergleich und Bewertung ausgewählter Sandbox Systeme für die dynamische Malware Analyse. – 2023. – 50 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2023.

Referat:

Aufgrund der steigenden Anzahl an Angriffen durch neue Malware Varianten ist es wichtig eine effektive Methode zu nutzen, um sich gegen diese Flut zu schützen. Diese von den Angreifern genutzte Malware muss identifiziert und analysiert werden, um die Systeme vor aktuelle und kommenden Angriffen schützen zu können. Für die Informationsextraktion stehen zwei grundlegende Ansätze zur Verfügung. Statische Analyse und die dynamische Analyse. Das Hauptaugenmerk liegt in dieser Arbeit auf der dynamischen Analyse. Diese wird genutzt um das Verhalten einer potentiell bösartigen Datei zu beobachten und anschließend auszuwerten, ob es sich um Malware handelt. Eine dafür häufig eingesetzte Methode ist die Sandbox. Bei dieser handelt es sich um eine isolierte Umgebung, in der eine Malware ausgeführt werden kann, ohne ein Risiko für das eigene System darzustellen. Bei der Nutzung eines Sandbox Systems wird ebenfalls von dem Begriff der automatisierten Malware Analyse gesprochen. Damit ist es möglich auch große Mengen von Malware Samples zu analysieren. Nach der Analyse wird neben der Bösartigkeit einer zu untersuchenden Datei, ebenfalls die gesammelte Daten über diese ausgegeben. Diese Arbeit vergleicht drei verschiedene Sandbox Systeme, um anschließend festhalten zu können, welches dieser Systeme die meisten Vorteile mit sich bringt. Bei diesen ausgewählten Sandbox Systemen handelt es sich um *Cuckoo*, *Any.Run* und *Hybrid Analysis*. Um eine Gegenüberstellung der Sanbox Systeme zu ermöglichen, wurden diverse Metriken verwendet. Zu diesen Metriken zählen unter anderem genutzte Anti-Evasion Techniken und die Möglichkeit einer URL Analyse. Nach umfassender Nutzung aller drei Sandbox Systeme, wurde eine Vergleichmatrix mit den bereits erwähnten Metriken erstellt. Anhand dieser konnten die Vor- und Nachteile der Sandbox Systeme gegeneinander abgewogen werden.

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

API . . . . . Application Programming Interface

APK . . . . . Android Package

C2C . . . . . Command and Control

CFG . . . . . Control Flow Graph

CIA . . . . . Confidentiality Integrity Availability

CPU . . . . . Central Processing Unit

DLL . . . . . Dynamic Link Library

DNS . . . . . Domain Name System

HAS . . . . . Hybrid Analysis Sandbox

HTML . . . . . Hypertext Markup Language

HTTP . . . . . Hypertext Transfer Protokoll

IDS . . . . . Intrusion Detetcion System

IoC . . . . . Indicator of Compromise

IoT . . . . . Internet of Things

IP . . . . . Internet Protokoll

IPS . . . . . Intrusion Prevention System

JSON . . . . . JavaScript Object Notation

PDF . . . . . Portable Document Format

RAM . . . . . Random-Access Memory

RAT . . . . . Remote Access Trojan

REST . . . . . Representational State Transfer

URL . . . . . Uniform Resource Locator

USB . . . . . Universal Serial Bus

VM . . . . . Virtual Machine

1 Einleitung

“Jeder kann Opfer von Internetkriminalität werden, und jedes Computing Gerät kann mit Malware infiziert werden. Niemand ist immun gegen Cyberkriminalität.“ [1]

1.1 Motivation

Durch die ständige Entwicklung neuer Technologien und der immer weiter wachsenden Verbreitung von internetfähigen Geräten, auch als Internet of Things (IoT) bekannt, bietet sich eine wachsende Angriffsfläche. Heute stellen durch ihre technologische Optimierung selbst Lampen oder Gefriertruhen eine potenzielle Bedrohung für den Benutzer dar. Diese Angriffsfläche wird von Angreifern ausgenutzt, dafür wird unter anderem verschiedene Malware genutzt. Bei Malware handelt es sich um Programme, die für ein Gerät schädliche Operationen ausführen oder andere Computerprogramme dafür befähigen. [2, S. 3–4, 237] [3, S. 12] [4, S. 681]

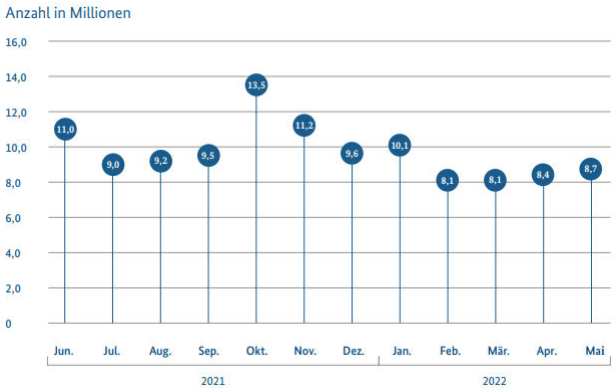

Abbildung 1.1: Neu entstandene Malware Varianten im Zeitraum vom Juni 2021 bis Mai 2022 [3, S. 13]

Die steigende Angriffsfläche führt ebenfalls zu einer steigenden Anzahl an Angriffen. Es kommen täglich neue Malware Bedrohungen in den Umlauf. Um diese enorme Anzahl an Bedrohungen erkennen und bekämpfen zu können, werden viele professionelle Cybersecurity Experten benötigt. Das akute und schon länger bekannte Problem ist dabei die fehlende Anzahl dieser Mitarbeiter. Malware tritt nicht nur in immer größerer Anzahl auf, sie wird immer komplexer und schwerer erkennbar. Angreifer versuchen durch Verwendung neuer Methoden die Erkennbarkeit eines Schadprogrammes zu erschweren. [2, S. 4] [5, S. 7] [3, S. 13]

Aus diesen Gründen muss die Analyse und Detektion dieser Malware stetig verbessert und neue Techniken entwickelt werden. Eine der besten Methoden im Kampf gegen Malware ist der Einsatz von Sandbox Systemen, um Malware zu identifizieren und zukünftige Angriffe durch angepasster Präventionsmaßnahmen zu verhindern. Mithilfe dieser Systeme kann Malware automatisiert analysiert und bestimmte Prozesse beschleunigt werden. Aufgrund des großen Nutzens dieser Methode, gibt es eine große Auswahl an Anbietern, wobei schnell der Überblick verloren geht. Deshalb stellt sich die Frage: Welches der Sandbox Systeme, die aktuell auf dem Markt vertreten sind, bietet die größten Vorteile? [6, S. 1] [7, S. 5] [8]

1.2 Ziel

Die Ziele dieser Arbeit liegen unter anderem darin, grundlegendes Wissen für das Verständnis von Sandbox Systemen zu vermitteln, um anschließend eine Vergleichs-Matrix aufzustellen. Nicht jede Sandbox hat die gleichen Voraussetzungen für die Analyse von Malware. Aus diesem Grund werden ausgewählte Metriken festgelegt, mithilfe welcher ein Vergleich zwischen den unterschiedlichen Sandbox Systemen erfolgen kann. Für die Erstellung dieser Matrix wird eine eingegrenzte Auswahl an Sandbox Systemen getroffen. Die erstellte Vergleichsmatrix kann anschließend eingesetzt werden, um die Vor- und Nachteile der jeweiligen Sandbox Systeme gegeneinander abzuwiegen. Durch diesen Vorgang soll erkennbar gemacht werden, welche Sandbox für welche individuellen Wünsche und Einsatzfelder am besten geeignet ist.

1.3 Abgrenzung

Gupta et al. [8] vergleichen vier Sandbox Systeme miteinander, um die Features reichste Sandbox zu ermitteln. Dabei kommen die Sandbox Systeme Norman Sandbox, GFI Sandbox, Anubis und Cuckoo zum Einsatz. Der Großteil der Vergleichsmetriken bezieht sich auf analysierbare Dateitypen. Pernet [9] stellt einen kurzen Vergleich zwischen den kostenlosen Versionen der online Sandboxen Any.Run und Joe Sandbox auf. Barker [10] liefert eine Übersicht und Erklärung über einzelnen Funktionalitäten von Sandbox Systemen. Dabei geht er auf die Sandbox System von Any.Run, Hybrid Analysis und anschließend Cuckoo ein. Sowohl Monnappa [11, S. 79] als auch Kaspersky [12] haben den Ablauf einer Sandbox dargestellt. Der von Kaspersky erläuterte Ablauf ist auf die von ihnen entwickelte Sandbox angepasst. Jedoch fehlen einige nicht dargelegte Details für diese Arbeit. Auch bei dem von Monnappa erläutertem Ablauf fehlten für diese Arbeit einige wichtige Informationen. Es gibt also keinen vollständig detaillierten Ablauf eines Sandbox-Runs.

Diese Arbeit beschränkt sich für den Vergleich auf drei verschiedene Sandbox Systeme. Die Auswahl dieser Systeme erfolgt anhand verschiedener Kriterien. Weiterhin werden verschiedene aussagekräftige Metriken bestimmt, um einen Vergleich zwischen diesen Sandboxen zu ermöglichen. Zum jetzigen Zeitpunkt sind keine Arbeiten publiziert, welche sich auf den Vergleich dieser explizit ausgewählten Sandboxen beziehen. Weiterhin ist keine Arbeit bekannt, welche eine Vergleichsmetrik im selben Umfang wie diese Arbeit bietet. Zuletzt wurde in dieser Arbeit ein vollständiger Ablauf einer Sandbox dargelegt. Die bereits veröffentlichten Arbeiten legen meist nur einen Teilablauf dar, beziehungsweise ist der Ablauf nur für die eigen entwickelte Sandbox zutreffend.

1.4 Aufbau

Die Kapitel 2 bis 4 erläutern die Grundlagen, welche für den Vergleich und die Bewertung der Sandbox Systeme benötigt werden. Dabei geht es in Kapitel 2 um die Grundlagen der Malware, wobei auf einige Arten eingegangen wird, welche für die diese Arbeit für wichtig erachtet werden. Weiterhin werden Formen der Verbreitung beschrieben und Ziele hinter Malware Angriffen beleuchtet. In Kapitel 3 geht es um die Malware Analyse, dabei wird besonders auf die Ziele der Analyse eingegangen, sowie die Methoden die dafür eingesetzt werden. In Kapitel 4

spielt das Thema der Sandbox Systeme eine besondere Rolle. Hierbei wird näher beleuchtet, wie eine Sandbox eine Malware Analyse durchführt. Weiterhin beschäftigt sich dieses Kapitel mit den Schwächen von Sandbox Systemen und dabei werden insbesondere Ausweichtechniken von Schadprogrammen hervorgehoben. Folgend wird auf die ausgewählten Sandbox Systeme eingegangen und anschließend die Vergleichsmatrix vorgestellt. Die entstandenen Ergebnisse werden in Kapitel 6 vorgestellt. Zuletzt erfolgt eine Zusammenfassung und ein Ausblick für zukünftige Arbeiten.

2 Malware

Häufig wird der Begriff Malware fälschlicherweise mit dem Begriff Virus gleichgesetzt. Allerdings ist ein Virus lediglich eine Untergruppe der Malware und deckt somit nur einen kleinen Teil dieser ab. Der Unterschied wird im folgenden Kapitel ersichtlich. [5, S. 10]

Bei Malware, oftmals auch als Schadsoftware bezeichnet, handelt es sich um Code, welcher bösartige Handlungen ausführt. Diese sind darauf angelegt, einem Computer, Netzwerk oder Benutzer Schaden zuzufügen. Zu solch schädlichen Handlungen zählen zum Beispiel das Stehlen wichtiger Informationen, die Kontrollübernahme eines Gerätes oder das Ausspionieren von Geräten und Netzwerken. Der Nutzen einer Malware weicht je nach Art ab. Der Code der Malware kann in Form eines Skriptes, Codes, einer ausführbaren Datei oder einer Software vorkommen. [13, S. 14] [11, S. 6] [14]

Neben der Vielfältigkeit verschiedener Malware Arten gibt es diverse Motivationen, welche ein Angreifer verfolgen könnte, sowie zahlreiche Methoden und Vorgehensweisen um diese zu erreichen. Im Folgenden wird eine ausgewählte Anzahl an Malware sowie deren Motivation erläutert, um einen ersten Einblick in das Thema zu liefern.

2.1 Motivationen des Angreifers

Warum führt ein Angreifer überhaupt einen Angriff aus? Und welche Absichten und Motive stecken hinter einer solchen Tat? Oft verfolgen Angreifer egoistische Interessen zum eigenen Vorteil. Es gibt jedoch auch Menschen, die dieses Wissen selbstlos als Mittel einsetzten, um gesellschaftliche oder politische Veränderungen herbeizuführen. Die Motivationen des Angreifers sind ebenso vielseitig wie die Methoden, welche eingesetzt werden, um diese Ziele zu erreichen. [15, S. 45]

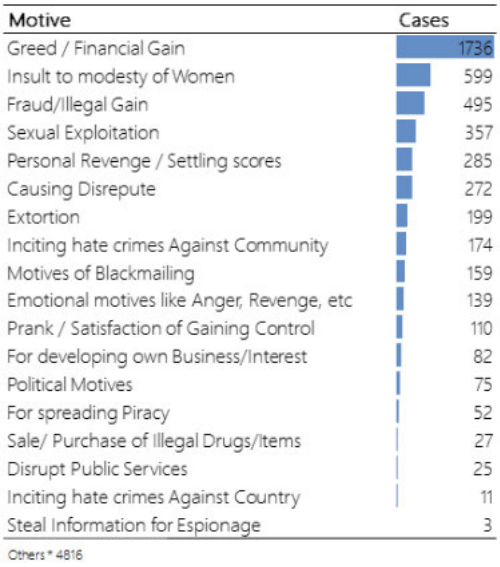

Abbildung 2.1: Motivationen für Cyberkriminalität in Indien im Jahre 2014 [16]

Rache/Vergeltung
Eines der am weitesten verbreiteten Motive für Angriffe ist Rache. Laut Eric Amberg und Daniel Schmid sind sowohl Arbeitgeber, Firmen und auch ehemalige Lebensgefährten betroffen. Der Grund für einen Angriff auf einen Arbeitgeber können Entlassungen sein. Der Auslöser für einen Angriff auf Firmen oder ehemaligen Partnern können aufgetretene Probleme sein. Das Ziel des Angreifers besteht darin, so viel Schaden wie möglich zu verursachen. [15, S. 45]

Geld
Ein ebenfalls sehr verbreitetes Motiv ist das Erzielen von Profit. Dieser Profit kann durch den Einsatz verschiedener Methoden erlangt werden. Besonders gängig sind hier Erpressung und Datendiebstahl. Die gestohlenen Daten können weiterverkauft werden. Abnehmer solcher Daten nutzen diese häufig zur Verbreitung von Spam E-Mails, um so sensible Daten des Opfers abzugreifen. Dies wird im folgenden Kapitel 2.2.3 näher erläutert. Nebst Erpressung und Datendiebstahl gibt es viele weitere Möglichkeiten für Hacker, sich mithilfe von Angriffen Profit zu sichern. [15, S. 45] [2, S. 8]

Interesse
Ein weiterer und eher simpler Grund für Hackerangriffe ist schlicht und einfach der Spaß an der Sache selbst. Für einige Menschen ist Hacking ein Hobby, welches ihnen einen kleinen Rausch aus Nervenkitzel und Triumph beschert. [15, S. 45]

Veränderung
Einige Hacker gehören zu den sogenannten Hacktivisten, welche durch ihr Handeln Veränderungen auf der Welt erzielen wollen. Der Begriff setzt sich aus den Worten „Aktivismus “ und „Hacken “ zusammen. Es handelt sich also um Hacker, welche politisch aktiv sind. Dafür Nutzen sie ihr Wissen um Angriffe auf Länder, Bewegungen, Firmen oder politische Parteien auszuführen und sich so Gehör zu verschaffen. [15, S. 45] [17, S. 30]

Damit wurden nur auf einige der wichtigsten Motivationen näher eingegangen. Neben diesen hier aufgeführten, gibt es eine Vielzahl weiterer Motivationen. Mehr über diese in “A Review of Motivations of Illegal Cyber Activities“. [18]

2.2 Verbreitung

Häufig kommen Methoden aus dem Social Engineering zum Einsatz, um Malware erfolgreich zu Verbreiten. Social Engineering nutzt menschliche Eigenschaften aus, um einen oder mehrere Menschen zu manipulieren. Zu diese Eigenschaften zählen unter anderem Angst, Vertrauen, Hilfsbereitschaft oder Respekt. Das Ziel ist es, vertrauliche Informationen zu erlangen, und Menschen dazu zu bringen, Schadsoftware auf dem eigenem Gerät zu installieren. Ein Social Engineer nutzt Methoden wie Phishing, drive-by-downloads oder auch die Neugier und Naivität des Menschen. Die Verbreitung von Malware kann in vier Kategorien unterteilt werden, die jedoch auch in Kombination auftreten können, um eine effizientere Verbreitung zu gewährleisten. [2, S. 171, 183] [15, S. 756, 766, 768] [19]

2.2.1 Physische Verbreitung

Für die physische Verbreitung wird meist ein Universal Serial Bus (USB)-Stick oder eine Festplatte verwendet. Dabei wird auf die Neugier des Menschen gesetzt. Sobald ein Mensch einen USB-Stick findet, ist es sehr wahrscheinlich, dass dieser sich den Inhalt des USB-Sticks anschaut, wie in Abbildung 2.2 zu erkennen. Die infizierte Datei, welche sich als Inhalt auf dem USB-Stick befindet, muss lediglich ausgeführt werden. Nachdem das Opfer die infizierte Datei ausgeführt hat, kann sich die Malware auf dem Computer ausbreiten. Eine weitere Form eines solchen Angriffs Angriffs kann mittels BadUSB-Sticks ausgeführt werden. Diese agieren als Tastatur und sind in der Lage Befehle auszuführen die dem System oder den Daten schaden können oder Daten des Opfers auslesen. [2, S. 171, 183] [20] [21]

Category	Drives Opened		p
Drive Type			
Confidential	29/58	(50%)	0.72
Exams	30/60	(50%)	0.71
Keys	32/60	(53%)	0.47
Return Label	17/59	(29%)	0.10
None	27/60	(45%)	–
Location Type			
Academic Room	25/58	(43%)	0.35
Common Room	26/60	(43%)	0.36
Hallway	24/59	(41%)	0.23
Outside	28/60	(47%)	0.58
Parking Lot	32/60	(53%)	–
Location Geography			
North	49/100	(49%)	0.26
South	46/97	(47%)	0.36
Main	40/100	(40%)	–
Time of Day			
Morning	71/149	(48%)	0.52
Afternoon	64/148	(43%)	–
Day of Week			
Tuesday	58/147	(39%)	0.05
Tuesday (no Return Label)	41/88	(47%)	0.57
Monday	77/150	(51%)	–

Abbildung 2.2: Wahrscheinlichkeitsstatistik für die Verwendung gefundener USB-Sticks [22, S. 5]

2.2.2 Verteilung mithilfe von Websites

Malware kann ebenso über Websites verbreitet werden. Die Infizierung des Systems der Opfer erfolgt, sobald diese die bösartige Website besuchen. Die Links zu solchen bösartigen Websites werden entweder per E-Mail verschickt, mithilfe von Werbung oder mittels kompromittierter legitimer Websites verbreitet. Malware-Verbreitung mithilfe von Werbung wird auch Malwaretising genannt. Diese Werbung beinhaltet Links, welche zu den kompromittierten Websites und Dateien führen. [2, S. 20, 171]

Websites werden als bösartig kategorisiert, sobald diese das Opfer bei einem Besuche mit Malware infizieren. Dabei kann es vorkommen, dass das Opfer keinerlei Kenntnis davon hat, da keine Interaktion seitens Opfer oder Angreifer notwendig ist. Für diese interkationslose Infizierung werden drive-by-downloads genutzt. Diese sorgen dafür, dass Malware automatisiert und ohne Zustimmung des Opfers auf sein System heruntergeladen und installiert wird sobald eine Website besucht wird. Für den Download sowie die Installation in Unwissenheit des Nutzers, werden Exploit Kits benutzt. Drive-by-downloads kommen ebenfalls oftmals bei dem Download von kostenloser Software zum Einsatz. Sobald der Benutzer diese herunterlädt und installiert, kommt es zu einer parallelen Installation der Malware im Hintergrund. Dafür wird Beispielsweise Spyware und Adware genutzt, welche in Kapitel 2.3 näher erläutert werden. [2, S. 21, 171, 184]

2.2.3 Verbreitung mittels E-Mail Verkehr

Bei dieser Art der Verbreitung werden die Links bösartiger Websites sowie Dateianhänge mittels E-Mails verbreitet, um die Verbreitungswahrscheinlichkeit zu erhöhen. Diese Methode wird Phishing genannt. Dabei verfolgt der Angreifer das Ziel, dem Opfer mittels Täuschung sensible Daten zu entlocken. Bei den verbreiteten bösartigen Dateianhängen kann es sich beispielsweise um Portable Document Formats (PDFs), Microsoft Office Dokumente und Skripte handeln. Diese dienen dem Zweck andere Malware zu Downloaden, weshalb sie auch als Downloader bezeichnet werden. Auf diese Malwareart wird in Kapitel 2.3 näher eingegangen. [2, S. 171, 185] [23] [15, S. 766]

In den meisten Fällen werden solche Links oder Anhänge in sehr großer Anzahl verschickt, um möglichst viele Opfer zu erreichen. Die Methode, die genutzt wird, um Malware massenhaft mithilfe von E-Mails zu verschicken, wird Spam genannt. Es ist allerdings wichtig zu erwähnen, dass Spam nicht bösartig sein muss, sondern auch harmlos sein kann. Die daraus resultierende erwünschte Folge ist die Ausführung der Datei oder das Öffnen des Linkes durch das Opfer, um die Malware zu starten und zu verbreiten. Bei Phishing handelt es sich um eine Form von Spam, welche jedoch definitiv bösartig ist. [2, S. 21, 181–182] [23]

2.2.4 Instant Messenger

Dieser Weg ermöglicht es infizierte Dateien zu übertragen und bösartige Links an Opfer zu senden. Das Opfer muss nur noch den Link anklicken oder die Datei ausführen, damit sein System mit Malware infiziert wird. Es ist dementsprechend sehr ähnlich zur Verbreitung durch E-Mails, allerdings handelt es sich um einen anderen Verbreitungskanal. [15, S. 457]

2.3 Arten/Typen

Malware wird in verschiedenste Arten eingeteilt. Jede dieser Arten teilt sich wiederum in eine Vielzahl von Unterarten auf. Beispielsweise ist ein Trojaner eine Art der Malware, welche weiterhin in verschiedensten Formen vorkommt. Zum Beispiel gibt es Backdoor, Ransomeware, Keylogger und viele weitere. Es kann ebenfalls zu einer Kombination unterschiedlicher Arten kommen.[15, S. 454] [5, S. 11]

In diesem Kapitel werden einige der bekanntesten Malware Arten näher beleuchtet. Das Ziel dieser Arbeit ist es einige der bekanntesten Arten zu erläutern um ein besseres Verständnis zu vermitteln. Bei einigen der vorgestellten Malware handelt es sich um Unterarten von anderen Malware Arten. Ebenfalls ist es wichtig zu benennen, dass Schadsoftware zu mehreren Arten gehören kann. [5, S. 11]

Virus
Der Virus ist eine Form der Malware, welche die Eigenschaft hat, sich selbst reproduzieren zu können. Bei einem Virus handelt es sich um ein Programm, welches fort besteht, indem es eine nicht infizierte Datei auf dem System infiziert und sich selbst in diese einfügt. Wenn diese infizierten Dateien oder Programme ausgeführt werden, werden alle normalen Funktionen des Programms ausgeführt. Jedoch wird gleichzeitig auch der Virus im Hintergrund ausgeführt, was zur Folge hat, dass der Prozess der Infizierung von vorne beginnt. Um die Funktion eines Virus zu starten, wird eine Interaktion des Nutzers benötigt, bevor dieser sich selbst reproduzieren kann. Eine solche Interaktion kann beispielsweise die Installation von bösartiger Software oder das öffnen eines E-Mail Anhanges sein. [15, S. 455] [2, S. 5] [24]

Wurm
Ein Wurm verteilt sich mittels physischen Medien wie USB-Sticks oder Netzwerken und infiziert so weitere Computer. Für die Verbreitung dieser Malware Art wird keine Nutzerinteraktion benötigt, wie es bei dem Virus der Fall ist. Der Wurm ist ebenso wie der Virus dazu fähig, sich selbst zu reproduzieren. [2, S. 5] [11, S. 7]

Trojaner
Diese Malware ist auch unter dem Begriff des Trojanisches Pferdes bekannt. Es gibt sich als harmloses Programm aus, welches allerdings Schadcode beinhaltet, mit dem der Computer infiziert wird. Dieses Programm wird mit Einverständnis des Nutzers installiert, dabei ist ihm der bösartige Zweck des enthaltenem Schadcodes unbekannt. Der Schadcode wird im Hintergrund ausgeführt. Diese Malware ist in der Lage, Daten zu stehlen, Webcams auszuspionieren und alle gesammelten Daten an den Angreifer zu übermitteln. Oftmals handelt es sich bei dem verstecktem Schadcode beispielsweise um Keylogger, Backdoor Trojaner oder Ransomware. [15, S. 454] [2, S. 5] [11, S. 7]

Backdoor/Remote Access Trojan (RAT)
Bei dieser Malware handelt es sich um einen Trojaner. Diese Variante des Trojaners ermöglicht es dem Angreifer Fernzugriff auf ein System zu erhalten und auf diesem Befehle ausführen zu können. [11, S. 7]

Adware
Dieser Typ Malware befindet sich oftmals in Software, die von Drittanbieter-Websites kostenlos heruntergeladen werden kann. In der Unwissenheit des Nutzers wird neben der gewünschten Software ebenfalls die Adware installiert. Adware führt dazu, dass ungewollte Werbung auf dem Gerät des Opfers angezeigt wird, indem Browsereinstellungen geändert werden. [2, S. 6] [11, S. 7] [25]

Ransomware
Der Angreifer nimmt die Dateien, Daten und andere Systemressourcen des Opfers als 'Geisel'. Die 'Geiselnahme' erfolgt indem die Ressourcen entweder verschlüsselt werden oder das Opfer aus seinem eigenem System ausgeschlossen wird. In den meisten Fällen verlangt der Angreifer ein Lösegeld für die Freigabe der Ressourcen. Diese Art der Malware ist sehr einfach für den Angreifer zu erstellen und sehr schwer für das Opfer zu beheben. Das liegt daran, dass die Verschlüsselung der Daten dafür sorgt, dass die Wiederherstellung kaum möglich ist und die Daten in den meisten Fällen verloren sind. [2, S. 6–7] [11, S. 7]

Rootkit
Diese Malware wird mit anderem Schadcode kombiniert. Das bedeutet, dass ein Rootkit für die Tarnung einer anderen Malware genutzt wird, sodass diese unbemerkt ihren Nutzen erfüllen kann. Dafür manipuliert es Prozesse oder Log Dateien und kann sogar das Anti-Viren Programm deaktivieren, um Spuren der transportierten Malware zu verwischen. [15, S. 483] [2, S. 6]

Downloader
Der einzige Zweck eines Downloaders besteht wie der Name bereits sagt, darin, den Download weiterer Malware oder Malware-Komponenten auszuführen. Diese werden meist installiert, nachdem der Angreifer sich das erste mal Zugriff auf ein Gerät verschafft hat. [11, S. 8] [13, S. 24]

Spyware
Dies Malware späht sensible Daten aus, stiehlt diese vom System des Opfers und sendet sie anschließend an den Angreifer. Unter diese sensible Daten fallen beispielsweise Benutzernamen, Passwörter, Dokumente, Bilder, aber auch Video- und Audioaufnahmen. [2, S. 5] [15, S. 456] [26, S. 8]

Keylogger
Keylogger gehören zu der Gruppe der Spyware und werden genutzt, um die Tastatureingaben des Opfers aufzuzeichnen und diese Aufzeichnung anschließend an den Angreifer zu senden. [2, S. 5]

Spammer
Der Einsatz des Spammers erfolgt für die Verbreitung von Spam E-Mails. Dabei nutzt ein Spammer die E-Mail Adresse des Opfers, um sich weiter zu verbreiten. Die E-Mails können schädliche Anhänge oder Links zu bösartigen Websites enthalten. Um möglichst viele Menschen damit zu erreichen liest die Malware die E-Mail Kontakte des Opfers und sendet die E-Mail an diese. [2, S. 7–8]

3 Malware Analyse

Bei der Malware Analyse spricht man von dem Prozess des Zerlegens von Schadprogrammen. Dieser Prozess dient dazu, das Verständnis für die Identifikation und Funktionsweise, sowie die Eliminierung der Schadsoftware zu erweitern. Die Analyse von Malware ist essentiell, um deren Haupteigenschaften und die verfolgten Ziele verstehen zu können, und so besseren Schutz vor kommenden Angriffen zu gewährleisten. Malware Analyse ist ein bedeutender Baustein für die Erkennung von bisher unbekannten Bedrohungen. [13, S. 14] [27, S. 5] [28, S. 1662]

Um Malware zu analysieren werden zwei grundlegende Ansätze verfolgt. Bei diesen handelt es sich um die statische Malware Analyse, zum anderen um die dynamische Malware Analyse. Bei der dynamischen Malware Analyse wird die Malware in einer gesicherten Umgebung ausgeführt. Im Gegensatz dazu wird bei der statischen Malware Analyse keine Ausführung der Malware erlaubt. Hierbei werden lediglich Rückschlüsse aus dem Assembler Code gezogen. Dieser Code und die darin enthaltenen Anweisungen werden aus dem Maschinen Code gewonnen und anschließend inspiziert. Laut Michael Sikorski und Andrew Honig, können diese beiden Ansätze weiterhin in 'Grundlegend' oder 'Erweitert' unterteilt werden, wie es in "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software" nachgelesen werden kann.[13] Die Unterschiede der beiden Ansätze werden in Kapitel 3.2 erläutert. Die beiden grundlegenden Ansätze können ebenfalls zur hybriden Malware Analyse kombiniert werden, worauf ebenfalls folgend eingegangen wird. [13, S. 22–23] [2, S. 526] [28, S. 1664, 1667]

3.1 Ziele

Die Malware Analyse verfolgt verschiedene Ziele. Zum einen wird die Analyse genutzt, um die Schadhaftigkeit einer Malware zu beurteilen. Ein weiterer Grund für die Analyse ist das Sammeln nützlicher Informationen über die vorliegende Malware. Bei den gesammelten Informationen handelt es sich um Bedrohungsinformationen, welche auch unter dem Begriff 'Threat Intelligence' bekannt sind. Bei den Informationen, die in der Threat Intelligence enthalten sind, handelt es sich um Indicator of Compromises (IoCs). Ein IoC ist ein auf dem System zurückgelassenes Artefakt, welches auf eine Kompromittierung hinweist. [2, S. 22] Dazu zählen beispielsweise Hash Werte und Netzwerkartefakte, die für die Identifizierung einer Malware benutzt werden können. Netzwerkartefakte können unter anderem Internet Protokoll (IP) Adressen, Uniform Resource Locators (URLs), Domains und einige weitere Informationen sein. [26, S. 4–5] [11, S. 8] [5, S. 13]

Das Sammeln von Informationen über die Malware hat einen weitreichenden Nutzen. Sie werden genutzt, um zukünftige Attacken zu identifizieren, zu verhindern und sind außerdem wertvoll für Forschungen in diesem Gebiet. Außerdem können im Fall eines Angriffes entsprechende Reaktionen auf diese Malware ausgeführt werden. Beispielsweise müssen die infizierten Systeme und Netzwerke eingedämmt werden, um eine Verbreitung zu unterbinden. [11, S. 8] [26, S. 4–5]

Eine sehr wichtige Informationsquelle für die von der Malware genutzten Techniken und Taktiken ist das MITRE ATT&CK Framework. Mithilfe dieses Frameworks kann ein besseres Verständnis von den genutzten Taktiken und die dafür eingesetzten Techniken, sowie deren Nutzen erlangt werden. Außerdem erlaubt MITRE ATT&CK eine bessere Klassifizierung, Kategorisierung und Berichterstattung von bestimmte Malware Samples, weshalb es auch von diversen Sandbox Systemen für die Berichterstattung genutzt wird. Die Bedeutung dieses Frameworks für die Malware Analyse wird in Kapitel 5.1 näher erläutert. Bei der Untersuchung der Malware ist es wichtig, dass die Analysten herausfinden, wie diese vorgeht, wie sie detektiert werde kann und wie man den Schaden eindämmt. Weiterhin müssen infizierte Maschinen und Dateien gefunden werden, um die Bedrohung anschließend auf weiteren Systemen entfernen zu können und Neuinfektionen zu verhindern. [26, S. 4] [11, S. 8–9] [10, S. 233–234]

Neben den Systemen und Netzwerken, welche gesichert werden müssen, ist es ebenso wichtig die darin vorhandenen und übertragenen Daten zu schützen. Dafür müssen die drei Schutzziele, welche auch mit Confidentiality Integrity Availability (CIA) abgekürzt werden, eingehalten werden. Zu Deutsch sind diese Ziele Vertraulichkeit, Integrität und Verfügbarkeit. Vertraulichkeit bedeutet, dass vertrauliche Informationen, wie beispielsweise personenbezogene Daten eines Kunden vor unbefugten Zugriffen geschützt sind. Dabei werden die Daten in Unterschiedliche Schutzklassen, top secret, geheim, sensibel und öffentlich eingeteilt. Je höher die Schutzklasse, umso höher sind die Schutzmechanismen, da es Angreifer meist auf die am höchsten geschützten Daten abgesehen haben. Integrität bedeutet, dass Daten weder während der Übertragung noch der Zwischenspeicherung verändert werden dürfen. Es muss also die Echtheit der Daten garantiert werden, bis diese bei dem Empfänger ankommen. Bei

dem Schutzziel der Verfügbarkeit muss sichergestellt werden, dass ein System im Internet und Netzwerk ansprechbar bleibt. Überlastungen, welche diese Ansprechbarkeit verhindern oder Einschränken sind zu vermeiden. [15, S. 51–55]

Sollte ein System von einer Malware kompromittiert wurden sein, kann nicht sichergestellt werden, dass die Schutzziele eingehalten werden können. Ein Angreifer, welcher Zugriff auf das System hat, kann ebenfalls die darin gespeicherten Daten verändern und anschauen. Somit ist sowohl das Ziel der Integrität, als auch der Vertraulichkeit verletzt. [15, S. 51]

Anhand aller gesammelten IoC können Signaturen und Verhaltensmuster für Anti-Viren Programme, Intrusion Detetcion Systems (IDSs) und Intrusion Prevention Systems (IPSs) erstellt werden. Ein IDS wird genutzt, um eine Infizierung zu erkennen und einen Alarm auszulösen. Ein IPS hingegen erkennt und alarmiert bei einer Infektion, wie es bei dem IDS der Fall ist. Jedoch versucht das IPS zusätzlich die Infizierung zu verhindern. Bei einer Signatur handelt es sich beispielsweise um einen einzigartigen Hashwert, um eine Schadsoftware Identifizieren zu können. [2, S. 19, 822] [26, S. 4] [11, S. 9]

3.2 Analyse Methoden

3.2.1 Statische Analyse

Die statische Analyse wird genutzt, um ausführbare Dateien potentieller Malware zu untersuchen und anschließend Informationen zu sammeln ohne diese Dabei auszuführen. Es wird sich vor allem auf die Daten und den Maschinencode der zu untersuchenden Bedrohung konzentriert, um zu verstehen wie diese operiert. [13, S. 22, 113] [5, S. 13]

Der Vorteil dieser Methode gegenüber der dynamischen Analyse liegt dabei in der Geschwindigkeit und den Kosten. Des weiteren ist diese Methode wesentlich sicherer, da die Malware nicht ausgeführt wird und somit keinen Schaden verursachen kann. Der Nachteil dieser Methode liegt darin, dass es nicht garantiert ist, dass eine Malware analysiert werden kann, da Ausweichtechniken (Evasion Techniken), wie Obfuskation oder das Verpacken von Malware (Packing) zum Einsatz kommen können. Obfuskation wird genutzt, um ein Programm durch ein anderes Programm zu ersetzten, welches die exakt gleiche Funktionalität bietet. Der Unterschied der Programme besteht dabei im Aufbau des Codes. Bei der Methode des Packing wird aus einer ausführbaren Datei eine neue ausführbare Datei erzeugt. Die verpackte ausführbare Datei ist komprimiert oder verschlüsselt. Auf diese Methoden wird in Kapitel 4.3 näher eingegangen. [29, S. 208] [13, S. 528] [28, S. 1662, 1664, 1667]

Bei der statischen Malware Analyse wird sich verschiedensten Methoden bedient, um Daten zu extrahieren. Ein Dissembler kann für die Dekompilierung von Malware eingesetzt werden. So wird der Maschinencode der ausführbaren Datei in Assembler-Code umgewandelt. Der Maschinen Code ist für einen Menschen zu schwer verständlich. Er beinhaltet bestimmte Anweisung, je nachdem welche Funktion das Programm erfüllen soll. Allerdings besteht dieser Code aus einer Byte Sequenz, was es für einen Menschen unmöglich macht diesen zu interpretieren. Aus diesem Grund wird Maschinen Code in Assembler Code umgewandelt. Dieser besteht aus für den Menschen lesbaren Abkürzungen. Aus diesem Code können die Anweisungen, wertvolle Informationen über die Malware und Muster für Signaturen, um die Erkennung des Schadprogrammes zu ermöglichen, ausgelesen werden. Die Umwandlung von Maschinen Code in Assembler Code wird als Reverse-Engineering bezeichnet. [13, S. 112–113] [2, S. 526–527] [11, S. 105] [28, S. 1164]

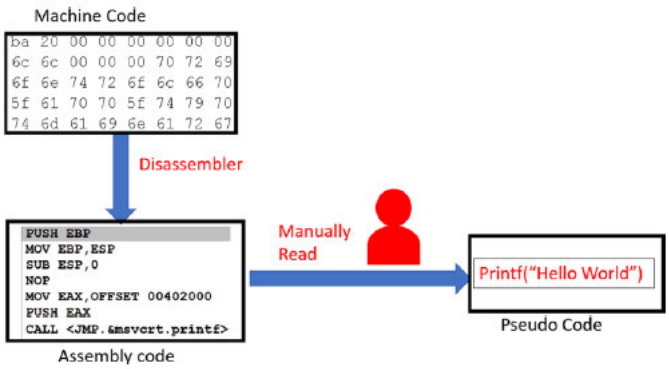

Abbildung 3.1: Umwandlung von Maschinencode in für Menschen lesbaren Assembler Code unter Verwendung von Reverse Engineering [2, S. 527]

Weiterhin können Windows Application Programming Interface (API) Anfragen genutzt werden, um eine Kommunikation mit dem Betriebssystem zu ermöglichen. Diese Kommunikation kann Aufschluss über das Verhalten des Programmes liefern. Zudem kann ein gerichteter Graph, auch als Control Flow Graph (CFG) bekannt, eingesetzt werden, um die Programmstruktur zu extrahieren und das Verhalten einer Datei festzuhalten. Auch Anti-Viren Programme werden bei der statischen Analyse verwendet, um die Bösartigkeit einer bereits bekannten Malware zu identifizieren. Dafür kann die Sandbox beispielsweise den Hash Wert der Malware ermitteln und übermittelt diesen anschließend an ein Anti-Viren Programm. Diese Methoden sind nur ein kleiner Einblick in die unzähligen Möglichkeiten der statischen Analyse. Es gibt eine Vielzahl weiterer Methoden wie String Analyse oder die Operation Code Analyse. [13, S. 28] [28, S. 1164]

3.2.1.1 Grundlegende statische Analysis

Die grundlegende statische Analyse wird genutzt, um eine ausführbare Datei zu untersuchen ohne sie auszuführen und ohne einen Dissembler zu nutzen. Anhand dieser Methode kann ohne hohen Aufwand und in einer kurzen Zeit bestätigt werden, ob es sich um eine bösartige Datei handelt. Außerdem können Informationen über die Funktionalität erhoben werden, um anschließend Netzwerk Signaturen für die jeweilige Malware zu erstellen. Dabei kommen beispielsweise Methoden wie die bereits erwähnte String Extraktion, Anti-Virus Abfrage und viele weitere zum Einsatz. Die Vorteile dieser Methode liegen in der Geschwindigkeit der Analyse, allerdings werden viele wichtige Informationen übersehen. Die Resultate dieser Methode sind grob und nicht immer für eine Detektion geeignet. Bei dieser Form der Analyse darf die zu untersuchende Datei niemals ausgeführt werden. Dies wird erst bei der Dynamischen Malware Analyse vorgenommen, wobei zu diesem Zeitpunkt genügend Schutzmaßnahmen für den eigenen Computer getroffen wurden. [13, S. 22, 28, 32, 110] [8]

3.2.1.2 Erweiterte statische Analysis

Die erweiterte statische Analyse nutzt einen Dissembler um Reverse-Engineering durchführen zu können. Damit ist es möglich die Programm Anweisungen zu analysieren und anhand der gesammelten Informationen zu bestimmen, was das Programm genau tut. “Grundlegende statische Techniken sind so, als würde man sich bei einer Autopsie das Äußere einer Leiche ansehen.“ [13, S. 110] Mithilfe von grundlegender statischer Analyse können Informationen über Importierte Funktionen erhalten werden, nicht aber wie diese eingesetzt werden. Für diese tiefgreifendere Analyse kommt die erweiterte statische Analyse zum Einsatz. Für diese Methode wird allerdings spezielles Wissen benötigt, um alle erhaltenen Informationen verstehen zu können. Zu den benötigten Kenntnissen zählen Windows Betriebssystem Konzepte, Code Konstrukte und Wissen über die Umwandlung von Maschinen Code in Assembler Code. [13, S. 22–23, 110]

3.2.2 Dynamische Analyse

Bei dieser Methode muss die potentiell bösartige Datei nicht zerlegt werden; sie wird lediglich in einer isolierten Umgebung ausgeführt, beobachtet und anschließend ausgewertet. Dabei ist es mithilfe der dynamischen Analyse möglich sowohl bekannte, als auch unbekannte

Schadsoftware zu erkennen. Durch diese Analyse kann aufgedeckt werden, welche Aktionen tatsächlich von einem Schadprogramm ausgeführt werden. Es können ebenfalls weitere nützliche Informationen erhoben werden, welche mithilfe der statischen Analyse nicht ermittelt werden können. Ein weiterer positiver Aspekt ist, dass bei der dynamischen Analyse keine Limitierung durch Obfuskation besteht, wie es bei der statischen Analyse der Fall ist. Der Code spielt für diese Analyse keine Rolle, da wie bereits erwähnt, das Verhalten der Malware beobachtet wird. Ein Nachteil der dynamischen Analyse hingegen ist, dass sie sehr zeitintensiv ist und eine große Anzahl an Ressourcen benötigt. Sollte das System nicht entsprechend abgesichert sein, kann die zu untersuchende Datei eine Gefahr für das bestehende System darstellen, da die Möglichkeit der Infizierung besteht. Die am meisten verwendete Technologien für die dynamische Malware Analyse ist der Einsatz einer Sandbox oder eines Debuggers. [13, S. 22, 75] [28, S. 1665]

3.2.2.1 Grundlegende dynamische Analyse

Bei der grundlegenden dynamischen Analysis kommt es zu einer Ausführung der Malware, um das Verhalten, sowie alle Aktionen ausgehend von der Malware beobachten zu können. Dabei können beispielsweise Änderungen, die an dem Dateisystem, der Internet Verbindung oder den Registrierungsschlüsseln, auch als Registry Keys bekannt, vorgenommen wurden, beobachtet werden. Es wird ebenfalls geprüft, ob neue Dateien erstellt und bestehenden Dateien verändert wurden. Anhand dieser und vieler weiterer Ergebnisse können Signaturen erstellt und infizierte Systeme wiederhergestellt werden. Es sollte darauf geachtet werden, entsprechende Umgebungen aufzubauen bevor es zu einer Malware Ausführung kommt, um seinen Rechner oder Netzwerk keinerlei Sicherheitsrisiken auszusetzen. Bei dieser Methode besteht die Möglichkeit, dass der schadhafte Code der Malware nicht ausgeführt wird und sie nur harmloses Verhalten aufweist. Grund dafür ist, dass Malware durch verschiedene Techniken die isolierte Umgebung erkennen kann. Auf einige dieser Techniken wird in Kapitel 4.3 eingegangen. Sobald eine isolierte Umgebung wahrgenommen wird, ändert sich das Verhalten der Malware oder der Schadcode wird nicht ausgeführt. Somit kann keine Analyse durchgeführt werden, da entweder harmloses oder kein Verhalten untersucht werden kann. [13, S. 22] [30, S. 315–316] [15, S. 507] [31, S. 2–3]

3.2.2.2 Erweiterte dynamische Analysis

Bei dieser Methode wird ein Debugger genutzt, um einen Einblick in den internen Zustand eines bösartigen Programmes während seiner Laufzeit zu erhalten. Debugger erlauben es ebenfalls Änderungen an Werten, wie beispielsweise Variablen, während der Ausführung vorzunehmen. Sie werden genutzt, um aus einer ausführbaren Datei detailliertere Informationen zu extrahieren. Dabei handelt es sich vor allem um Informationen, welche mit anderen Techniken schwer zu sammeln sind. [13, S. 23, 242]

3.2.3 Hybride Analysis

Diese Methode wird genutzt um Daten sowohl mittels statischer, als auch dynamischer Analyse zu sammeln. Durch die Kombination beider Methoden können Vorteile beider Methoden genutzt werden und Malware folglich präziser erkannt werden. Durch die Verwendung beider

Methoden ist es sowohl möglich Malware Evasion Methoden wie Obfuskation zu überwinden, indem die Malware ausgeführt wird, als auch intelligentere Berichte zu erhalten. Einige Malware Evasion Techniken, wie unter anderem Obfuskation, werden in Kapitel 4.3 näher beleuchtet. [6, S. 6] [28, S. 1667]

4 Sandbox

Die steigende Anzahl neuer Malware Varianten führt zu einer erschwerten Erkennung mittels Anti-Viren Programmen. Anti-Viren Programme erkennen Malware basierend auf ihrer Signatur und zählen unter die statische Analyse. Die Erkennung ist nur möglich, wenn bereits ein einzigartiges Muster im Quellcode oder ein Hash von dieser Malware bekannt ist. Ein Hash ist eine Zeichenkette, welche aus den Daten einer Datei erzeugt wird. Anhand dieses Wertes kann festgestellt werden, ob Änderungen an der Datei vorgenommen wurden, da bei der kleinsten Änderung ein neuer Wert herauskommt. Jedoch kommt es nicht bei jeder Hash Funktion zu einer vollständigen Veränderung des Wertes. Es gibt ebenfalls Hash Funktionen, bei welchen der Wert weiterhin für den Abgleich genutzt werden kann. Ein Beispiel für ein solches Verfahren ist ssdeep. Die Anti-Viren Programme vergleichen die vorhanden Daten mit den Daten der geprüften verdächtigen Datei. Damit kann bestimmt werden, ob es ein bereits bekanntes bösartiges Programm mit denselben einzigartigen Mustern oder Hashes gibt. Das Problem daran ist, dass Anti-Viren Programme bei einigen Hashes nicht mehr in der Lage sind, eine Malware an dem bekanntem Hash zu erkennen. [7, S. 5] [13, S. 28] [32] [3, S. 13] [11, S. 55]

Wie bereits erwähnt kann es bei der kleinsten Änderung des Inhaltes einer Datei zu einem komplett neuem Hash kommen. Das bedeutet, sobald der Angreifer den Inhalt seiner Malware verändert hat, kann der von einigen Funktionen erstellte Hash nicht mehr von Anti-Viren Programmen erkannt werden. Es entsteht eine neue Malware Variante. Aus diesem Grund muss ein anderes Vorgehen genutzt werden, um in der Lage zu sein, unbekannte Bedrohungen identifizieren zu können. Dafür kommt die dynamische Malware Analyse und speziell Sandbox Systeme zum Einsatz. Die Bedrohlichkeit einer potenziellen Schadsoftware kann anhand ihres Verhaltens identifiziert werden. Dafür wird die Malware bei der dynamischen Analyse in einem isolierten Bereich ausgeführt. Weiterhin können mit dieser Methode weitere nützliche Informationen gesammelt werden, wie Funktionalität, Herkunft, Auswirkungen auf infizierte Systeme und viele weitere. Anhand dieser Informationen können potenziell Auswirkungen, Ziele des Angreifers und der Angreifer selbst ermittelt werden. [15, S. 503] [30, S. 315] [33, S. 2] [34, S. 1] [3, S. 13]

Ebenfalls ist es nicht möglich alle neuen Malware Bedrohungen per Hand zu analysieren, was dazu führt, dass der Einsatz von Sandboxen umso wichtiger wird. Diese können neben der Erkennung unbekannter Bedrohungen ebenfalls für die automatisierte Malware Analyse genutzt werden. Die Sandbox führt automatisch die Analyse durch, indem es das Verhalten der verdächtigen Datei observiert und anschließend die gesammelten Informationen in einem Bericht zusammenfasst. Somit kann viel Zeit gespart werden, die für die händische Analyse benötigt worden wäre. [5, S. 7] [2, S. 21] [13, S. 76] [28, S. 1662]

Eine Sandbox wird also in vielen Situation verwendet und ist sehr effizient. Genauer beschrieben handelt es sich bei einer Sandbox, um einen speziell isolierten Bereich, der sich innerhalb eines Systems befindet. Alle Vorkommnisse die in diesem Bereich geschehen, haben keinerlei

Auswirkung auf die Umgebung außerhalb. Genutzt werden diese speziell isolierten Bereiche zum Testen von Anwendungen oder potentiellen Schadprogrammen, ohne das System der Gefahr einer Infizierung auszusetzen. [33, S. 2] [15, S. 536]

In der Sandbox kann das Verhalten einer Datei oder Programmes beobachtet und analysiert werden. Dieses Verhalten wird in dem Bericht in verschieden Kategorien unterteilt. [15, S. 536] [35] Einige der wichtigsten sind dabei folgend aufgeführt:

Prozesse
Malware ist in der Lage Prozesse zu erstellen. Diese Prozesse können folgend dafür genutzt werden, den eigentlichen Schadcode der Malware ausführen. In einigen Fällen, wird Schadcode in einer sogenannten Dynamic Link Library (DLL)-Datei gespeichert. Diese Datei kann anschließend an einen Prozess gehangen werden. All diese von der Malware erstellten bösartigen Prozesse müssen identifiziert werden, um die Infizierung anschließend eliminieren zu können. [10, S. 43] [13, S. 14, 84, 216–218] [26, S. 4]

Windows Registry
In Windows Registry werden Informationen über die Konfigurationen des Betriebssystem gespeichert. Das Registry wird von der Malware genutzt, um sich persitent auf dem System verfügbar zu machen. Dafür wird von der Malware ein Eintrag in Windows Registry eingefügt. Danach wird die Malware automatisch beim Start des Systemes ausgeführt. Die Überwachung der Registry Aktionen ermöglicht es dem Analysten nachzuvollziehen, wie sich eine Malware im Windows Registry verankert hat. [13, S. 84, 208]

Dienste
Dienste können von der Malware genutzt werden, um weiteren Schadcode auszuführen. Zusätzlich kann die Malware mithilfe eines Dienstes automatisch beim Start des Systems ausgeführt werden und kann somit das System langfristig infizieren. Für den Analysten ist wichtig einen solchen Dienst identifizieren zu können, um das infizierte System wiederherzustellen. [13, S. 224–225] [26, S. 4]

Datei System
Malware ist in der Lage, existierende Dateien und Dateinamen zu verändern, sowie Dateien zu erstellen. Durch die Beobachtung von Aktivitäten bezüglich des Datei Systems, kann festgestellt werden, um welche Art Malware es sich handelt und welche Dateien für eine weitere Untersuchung von Bedeutung sind. [13, S. 84, 204]

Netzwerk Informationen
Durch die Kontrolle des ausgehenden Netzwerkverkehrs können wertvolle Daten erlangt werden. Darunter zählen IP-Adressen und Domain Name System (DNS)-Namen. Die IP-Adressen werden gebraucht, um die Command and Control (C2C) Server identifizieren zu können, von denen die Malware Befehle entgegen nimmt. Die dabei gewonnen IP-Adressen sind Indikatoren für einen Angriff. Sie können genutzt werden, um Systeme vor Netzwerkverkehr zu schützen, welcher von diesen ermittelten IP-Adressen ausgeht. Die ermittelten DNS-Namen werden genutzt, um die zu den bösartigen IPs gehörenden Domains zu ermitteln. Ebenfalls kann herausgefunden werden, auf welchen Ports möglicherweise eine Malware lauscht. [10, S. 63, 65] [13, S. 84] [26, S. 4–5]

Anhand dieser Daten kann festgestellt werden, ob es sich um Malware handelt und was deren Charakteristika sind. [4, S. 683]

Auch wenn die Nutzung einer Sandbox unter die dynamische Malware Analyse fällt, ist es wichtig zu erwähnen, dass einige Sandboxen ebenfalls Techniken der statischen Malware Analyse nutzten, um mehr Informationen zu sammeln. In den folgenden Kapitel wird auf den Ablauf von Sandbox Systemen, sowie die Schwächen dieser gesprochen. Insbesondere wird dabei auf die Schwäche der Evasion Techniken eingegangen. [27, S. 8] [36]

4.1 Ablauf eines Sandbox-Runs

Es existiert keinen genauer Ablaufplan, welcher auf jede Sandbox zutreffend ist. Aus diesem Grund wurde für diese Arbeit der Ablauf einer Sandbox aus diversen Quellen herausgearbeitet.

Eine Sandbox besteht aus zwei Komponenten. Die erste Komponente ist die Host Maschine, über welche die Komponenten für den Analyse Prozess und die Auswertung ausgeführt werden. Die zweite Komponente ist die Gast Maschine wobei diese aus ein oder mehreren Maschinen bestehen kann. Bei diesen handelt es sich meist um Virtual Machines (VMs), es können aber auch physische Maschinen sein, die für die Analyse benutzt werden. VMs sind Computerprogramme, welche Anwendungen und Betriebssysteme ausführen. [28, S. 1666] Auf diesen Gast Maschinen können die Malware Proben, auch als Samples bekannt, in einer isolierten Umgebung ausgeführt werden, ohne das ein Risiko für das System des Benutzers besteht. Bei den VMs für die Gast Maschine handelt es sich meist um Windows Maschinen, da dieses Betriebssystem statistisch von der Mehrheit der Computer Nutzer benutzt werden und somit die meiste Malware für dieses erstellt wird. Mehr Informationen darüber, werden in Kapitel 5.1 dargestellt. Ein weiterer Vorteil von VMs für Malware Analyse ist die Funktion des Snapshots. Diese Funktion sorgt dafür, dass der aktuelle Stand einer VM gespeichert werden kann. Dieser Snapshot kann zu einem späteren Zeitpunkt wiederhergestellt werden, sollten Änderungen an der VM vorgenommen werden.[13, S. 62, 69] [37, S. 32] [38, S. 46] [26, S. 7]

Bei jeder Sandbox gibt es gewisse Unterschiede in Hinsicht auf den Ablauf des Analyse Prozesses. Der Grundablauf bleibt jedoch gleich, lediglich einzelne extra Eigenschaften, die jede Sandbox individuell machen, kommen an bestimmten Stellen zum Einsatz. Dabei ist es wichtig vor jeder Untersuchung die einzelnen Maschinen, die für die Analyse genutzt werden, auf ihren Urpsrungszustand zurückzusetzten. Dafür werden die bereits erwähnten Snapshots verwendet. Gründe dafür sind, dass die Malware nach der Ausführung Änderungen an dem System vornehmen kann, das vorherige Sample noch nicht terminiert wurde oder noch Artefakte von dieser vorhanden sind. Alle diese Dinge könnten dazu führen, dass eine Malware, die in einer solchen Umgebung ausgeführt wird, verfälschtes Verhalten beziehungsweise kein bösartiges Verhalten aufweist. [2, S. 404] [31, S. 3]

Darauf folgt die Ausführung der Überwachungs- und Analyse-Werkzeuge über die Host Maschine, um das Verhalten der Malware beobachten zu können. Nachdem der Urpsungszustand wiederhergestellt wurde und alle wichtigen Bereiche beobachtet wurden, kann das Malware Sample an die Gast Maschine übertragen werden. Dabei können je nach Sandbox sowohl unterschiedliche Dateien, als auch URLs von verdächtigen Websites untersucht werden. Sobald das Sample an die Sandbox übertragen wurde, kommt es zur Ausführung der Malware in der Gast Maschine. Das zu analysierende Objekt wird über einen vordefinierten Zeitraum ausgeführt. Währenddessen werden Artefakte gesammelt, die das Verhalten der Malware betreffen. Dabei kann es sich beispielsweise um Registry Änderungen oder über Informationen von Netzwerkverbindungen handeln. [11, S. 79] [12] [37, S. 32] [38, S. 46]

Sobald diese Beobachtung gestoppt wurde, analysiert die Sandbox die gesammelten Resultate. Anhand dieser Informationen entscheidet die Sandbox über die Bösartigkeit des überwachten Objektes. Sandboxen sind in der Lage sowohl bekannte, als auch unbekannte Bedrohungen zu identifizieren. Anschließend wird die getroffene Entscheidung, sowie die gesammelten Artefakte als Bericht an das Anfrage System übertragen, um das Ergebnis besser nachvollziehen zu können. Dieser Bericht beinhaltet unter anderem detaillierte Beschreibungen über verdächtige Aktivitäten, welche von der Malware ausgehen. Auf die wichtigsten dieser Aktivitäten wurde bereits in Kapitel 4 eingegangen. [11, S. 79] [12] [28, S. 1665]

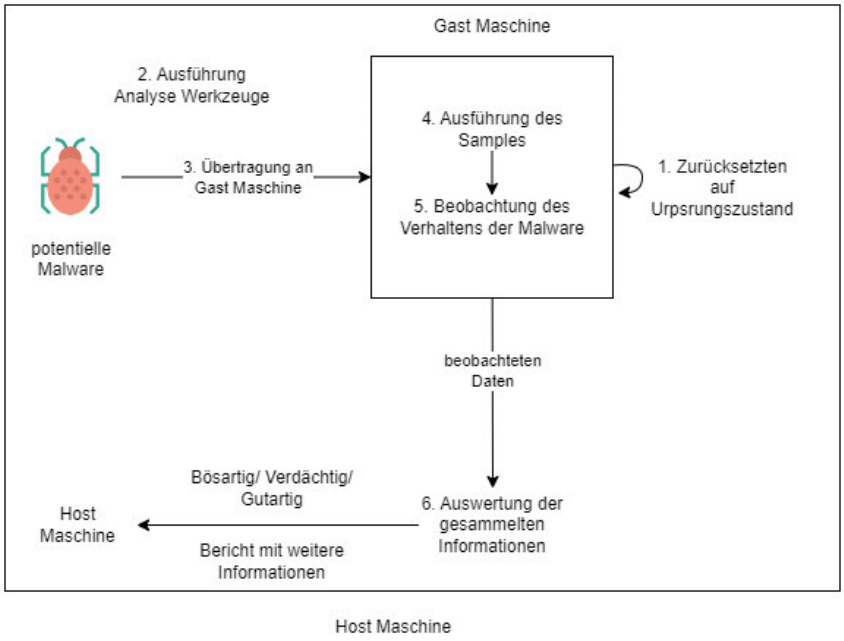

Abbildung 4.1: Prozess des Ablaufes einer Sandbox

Neben dem hauptsächlichem Ablauf gibt es noch einige Besonderheiten die von Sandbox zu Sandbox unterschiedlich sind. Bei den meisten Sandboxen müssen vor der Anfrage bestimmte Einstellungen getroffen werden. Diese Einstellungen betreffen die Umgebung in der die Malware ausgeführt wird. Dabei geht es beispielsweise um das gewünschte Betriebssystem, Konfigurationen und vorinstallierte Programme. [12]

Eine öffentliche Sandbox, auch Online Sandbox genannt, kann über eine Website im Internet erreicht werden. Diese Form der Sandbox hat gegenüber einer lokal erstellten Sandbox bestimmte Vor- und Nachteile. Ein großer Vorteil bei der Nutzung von öffentlichen Sandboxen ist der Wegfall der Notwendigkeit von leistungsfähiger Hardware, welche bei der Erstellung einer lokalen Sandbox mithilfe von VMs benötigt worden wäre. Ebenfalls sind diese Sandbox Systeme immer auf dem neusten Stand der Technik, um sich gegen die Konkurrenz behaupten zu können. Der Nutzer hat dabei weder Aufwand für die Implementierung der Sandbox, noch für die Wartung. [13, S. 640] [33, S. 2]

Der Nachteil einiger Online Sandbox Systeme hingegen liegt darin, das Samples die zur Analyse an die Sandbox übermittelt werden, von den Sandbox Betreibern gespeichert werden. Da diese Samples vertrauliche Informationen über eine Firma beinhalten können, ist es sicherer diese Informationen nicht der Öffentlichkeit zugänglich zu machen. Meist werden die analysierten Dateien nur in kostenlosen Versionen der Sandbox Systeme gespeichert. Wie bereits erwähnt, stellen manche Online Sandbox Anbieter kostenlose Versionen ihrer Sandbox zur Verfügung, bei denen jedoch oftmals bestimmte Features fehlen. Sollten diese fehlenden Features essentiell für den gewünschten Einsatz sein, können die Versionen der Sandbox aufgewertet werden, was dazu führt das diese dann kostenpflichtig sind. [13, S. 76] [39]

4.2 Nachteile einer Sandbox

Ein bedeutender Nachteil beim Einsatz von Sandbox Systemen ist die Analysedauer, da verschiedene Malware Arten unterschiedlich lange ausgeführt werden müssen, bevor diese ihr bösartiges Verhalten aufweisen. Grund dafür ist die Implementierung von Funktionen, welche die Ausführung absichtlich herauszögern. Auf diese wird in Kapitel 4.3 eingegangen. Somit kann es bis zu fünf Minuten dauern, bis das Verhalten einer einzelnen Malware überhaupt beobachtet werden kann und anschließend einen Bericht erstellt werden kann. [40] [7, S. 10]

Nachdem der Bericht erstellt wurde, muss dieser von einem Menschen ausgewertet werden. Dabei ist die Länge des Berichtes, sowie die Menge an Details von Sandbox zu Sandbox unterschiedlich. Wenn viele Informationen in einem Bericht enthalten sind, kann es dazu führen, dass die Auswertung dieses Berichtes zeitintensiver wird. In dem Bereich des Incident Response ist es wichtig schnell auf einen Vorfall zu reagieren. Sollte ein Bericht viele Informationen beinhalten, von denen einige irrelevant sind, raubt die Auswertung des Berichtes dem Analysten Zeit. Es ist sehr wichtig für einen Analysten schnell auf einen Vorfall zu reagieren. Somit ist verschenkte Zeit umso problematischer. [40] [41]

Ein weiterer Nachteil der Sandboxen betrifft die Erkennung von Malware. Angreifer entwickeln die Funktionalität ihrer Malware immer weiter. Dabei implementieren sie in ihrer Malware Funktionen, um sich vor Sandboxen verbergen zu können. Ein Beispiel für eine solche Funktion ist das Hinauszögern der Ausführung, welche bereits im ersten Absatz erwähnt wurde. Eine weitere Funktion, ist die Überprüfung der Installation bestimmter Software, bevor die Malware ausgeführt wird. [4, S. 683] [6, S. 4, 5]

Die Methoden, die Angreifer nutzen, um ihre Malware vor Sandbox Systemen zu verschleiern, sind umfangreich und es werden immer neue Methoden entwickelt. Diese müssen von Sandboxen umgangen werden, um sicherzugehen, dass eine Malware ihr wahres schädliches Verhalten aufweist. Eine Malware kann bei Erkennung einer Sandbox verändertes Verhalten oder sogar überhaupt kein schädliches Verhalten aufweisen, was die Ergebnisse der Verhaltensanalyse der Sandbox verfälscht. Das folgende Kapitel beschäftigt sich mit diesen Evasion Techniken und wird einige näher beleuchten. [7, S. 6, 7, 8–10]

4.3 Evasion Techniken

Der Sinn dieser Ausweich-Techniken ist es zu überprüfen, ob die Malware möglicherweise in einer isolierten Umgebung ausgeführt wird. Sollte dies der Fall sein, wird das Verhalten der Malware verändert oder komplett unterbunden, um so die Analyse zu verhindern. [7, S. 7]

Angreifer haben diese Maßnahmen getroffen, um ihre Malware vor der Analyse zu verbergen. Sie testen, ob es sich tatsächlich um das System eines Nutzer handelt oder ob stattdessen ein Testsystem, wie eine Sandbox, für die Ausführung genutzt wird. Diese Malware wird als kontext-bewusste Malware oder Sandbox Evasive Malware bezeichnet. Der Angreifer geht sicher, dass es sich bei dem Ziel um ein System handelt, auf welchem wertvolle Daten oder Dienste vorhanden sind. Um die Malware trotzdem analysieren zu können, nutzt die Sandbox sogenannte Anti-Evasion Techniken. So kann sie die Evasion Techniken umgehen, damit die Malware analysiert werden kann. [42, S. 21, 22]

Es gibt bestimmte Evasion Techniken, auch als anti-statische Analyse Funktionalitäten bezeichnet, welche gezielt die statische Analyse verhindern. Ebenso gibt es auch Evasion Techniken, welche für die Verhinderung der dynamischen Analyse eingesetzt werden. Diese sind analog unter dem Namen Anti-dynamische Analyse Techniken bekannt. Zu den Methoden der Anti-statischen Analyse zählen beispielsweise Code-Obfuskation und Packing des Codes. Für die Anti-dynamische Analyse werden Methoden wie Hinauszögerung, System Umgebungserkennung und viele weitere Methoden genutzt, welche im folgenden näher erklärt werden. [7, S. 10, 12] [6, S. 1, 4]

Obfuskation
Obfuskation wird genutzt, um die Malware in ein Programm mit gleicher Funktionsweise umzuwandeln, jedoch hat die neue Variante einen deutlich schwieriger analysierbaren Code. Dabei werden Techniken eingesetzt, um den Code der Malware zu verändern. Das Ziel dieser Veränderung ist es die Anordnung oder den Ablauf des Codes zu ändern, indem beispielsweise die Anweisungen dieses Programmes reorganisiert wird, auch als Code Transposition bekannt. Es können ebenfalls sinnlose Operationen in dem Code eingefügt werden, wie es bei Dead-Code Insertion der Fall ist, um Signatur-basierte Erkennung zu umgehen. Der Code der Malware wurde verändert, nicht jedoch die Funktionsweise. Mit der Veränderung des Codes, ändert sich auch der Hashwert der Datei, was dazu führt, dass mithilfe einer einfachen Überprüfung diesen Wertes durch ein Anti-Viren Programm kein bösartiges Programm erkennt werden kann. Dieses Thema wurde bereits in Kapitel 4 erwähnt. Es gibt eine große Anzahl verschiedener Obfuskation Techniken, die von Angreifern genutzt werden, jedoch sind nur Ausgewählte Gegenstand der in dieser Arbeit vorgestellten Untersuchung, da der fachliche Schwerpunkt hier auf der dynamischen Analyse liegt. [29, S. 208–209] [7, S. 6] [13, S. 30]

Packing
Diese Methode versteckt den echten Code, indem eine oder mehrere Schichten Verschlüsselung oder Kompression auf dieses Programm angewendet werden. Um eine solche Malware analysieren zu können, muss die Malware zuerst entpackt werden. [7, S. 6] [13, S. 528]

Aufgrund der Techniken zur Verhinderung der statischen Analyse musste eine Neuerung für die Malware Analyse entwickelt werden. Um diese Hürde zu überwinden, wurde die dynamische Analyse entwickelt und eingesetzt. Unter anderem kommen Sandbox Systeme zum Einsatz, um die dynamische Analyse automatisiert durchzuführen. Aufgrund der neuen Lösung für die Malware Analyse haben die Angreifer ebenfalls neue Methoden entwickelt. Bei diesen Methoden und Funktionen handelt es sich um so genannt Anti-dynamische oder auch Anti-Sandbox-Malware-Analyse Methoden. Mithilfe dieser Techniken ist es dem Angreifer möglich, auch die dynamische Analyse zu erschweren oder sogar ganz zu verhindern. Von diesen Techniken werden einige im folgenden kurz vorgestellt. [6, S. 1, 4, 5]

Hinauszögern
Diese Methode wird genutzt, um die Malware in einen Ruhezustand zu versetzten und so einer Entdeckung durch die Sandbox zu entkommen. Aufgrund der begrenzten Analysezeit, kann es somit sein, das die Malware erst ausgeführt wird, nachdem die Sandbox ihren Analyseprozess bereits abgeschlossen hat.[7, S. 10]

Menschliche Interaktion
Einige Sandbox Systeme simulieren keinerlei menschliche Interaktion. Diese fehlende Interaktion kann die Malware als Indikator nutzten, um festzustellen, dass es sich um eine Sandbox handelt. Menschliche Interaktion bedeutet in diesem Fall Mausbewegungen, die Anzahl der ausgeführten rechts und links Klicks, die Bewegungsgeschwindigkeit der Maus und sogar das Scrollverhalten des Benutzers. Diese Faktoren werden gemessen und mit den Messwerten eines menschlichen Benutzers verglichen. Wenn die gemessenen Daten nicht mit den Daten eines solchen Benutzers übereinstimmen, kann es dazu kommen, dass der Schadcode nicht ausgeführt wird. Weiterhin greifen einige Malware Varianten, dazu erst ihr bösartiges Verhalten zu zeigen, sobald der Benutzer einen Neustart des Systems ausführt. Der Grund dafür ist, das ein Neustart die Zurücksetzung auf den Ursprungszustand bedeutet. Somit ist alles bereits aufgezeichnete Verhalten der Malware verloren und die Auswertung findet nicht statt. Eine Beurteilung der Bösartigkeit der Malware ist damit nicht möglich, weshalb Gegenmaßnahmen getroffen werden sollten. [6, S. 5] [7, S. 10]

Registry
Das Windows Registry enthält wichtige Informationen über ein System, wie zum Beispiel installierte Programme, System Informationen oder vorhandene Dienste. Diese Informationen können ein System als Debugger oder VM entlarven. Dafür muss die Malware nur nach Registry Keys suchen, welche nur in einer virtuellen Umgebung vorhanden sind. Ein Registry Key von 'Virtual Box' ist beispielsweise "Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\VirtualBox Guest Additions". [7, S. 10, 11]

System Umgebung
Das Prüfen von systemrelevanten Informationen, wie der Anzahl an Central Processing Unit (CPU) Kernen, Random-Access Memory (RAM) Speicher oder Festplattenspeicher, kann Aufschluss darüber geben, ob es sich um eine isolierte Umgebung oder ein Nutzer System handelt. Die Nutzung dieser Ressourcen ist in einer isolierten Umgebung oftmals wesentlich niedriger als es in einem Nutzer System der Fall ist. Eine Überprüfung dieser Ressourcen kann der Malware Aufschluss über die Umgebung, in welcher sie ausgeführt wird, geben. Ei-

ne Malware kann ebenfalls nach vorinstallierten Programmen, wie E-mail Clients oder Instant Messengern, suchen. Diese sind in isolierten Umgebungen manchmal nicht installiert und somit sind diese daran erkennbar. [7, S. 12] [31, S. 4] [43] [44]

Dateisystem
Indem die Malware das Dateisystem nach bekannten Ordnern durchsucht können Hinweise auf eine VM gefunden werden. Beispielsweise könnte der Ordner gefunden werden, welcher von dem Programm 'Virtual Box' genutzt wird, um VMs auszuführen. Der Pfad zu diesem Ordner lautet "C:/ProgramFiles/Oracle/VirtualBox Guest Additions/". Sollte die Malware diesen Ordner finden, kann es zu einer Verhaltensänderung führen, welche die Analyseergebnisse verfälscht. [7, S. 12] [31, S. 3]

Prozesse
Bestimmte Prozesse weisen auf die Ausführung einer isolierten Umgebung hin. Die Überprüfung der Prozesse durch die Malware kann sicherstellen, ob sie möglicherweise in einer Sandbox, beziehungsweise isolierten Umgebung ausgeführt wird. Ein solcher Prozess wäre zum Beispiel "VBox.exe". [7, S. 13]

Bei diesen Evasion Techniken handelt es sich um eine Auswahl der bekanntesten Methoden, welche von Angreifern verwendet werden. Es kommen immer wieder neue Methoden hinzu, um bereits entwickelte Anti-Evasion Techniken der Sandbox Systeme zu umgehen und damit die Analyse der Malware zu erschweren. [6, S. 5]

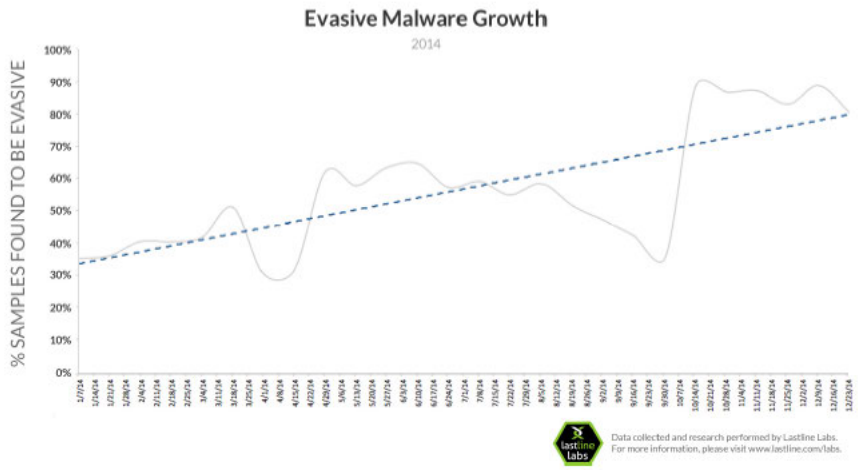

Abbildung 4.2: Anstieg von Evasive Malware über das Jahr 2014 [6, S. 3]

5 Auswahl der Sandbox Systeme

Es stand eine große Anzahl an Sandbox Systemen zur Auswahl, welche unterschiedlichste Features zur Verfügung stellen. Durch eigenständige Recherche wurde eine Liste verschiedener Sandbox Systeme erstellt. Anschließend wurden aus dieser Liste, anhand der folgenden erläuterten Kriterien, die für diese Arbeit genutzten Sandboxen ausgewählt. Dabei war es auffällig, dass hier vor allem Online Sandbox Systeme vertreten waren. Diese sind meist kostenpflichtig erwerbbar, beziehungsweise es muss für bestimmte Features gezahlt werden. Dagegen gab es eine kleine Anzahl an Sandbox Systemen, die selbst installiert werden müssen. Unter diesen gab es eine noch kleinere Anzahl, welche ähnlich viele Features mit sich bringen, wie es bei öffentlichen Sandbox Systemen der Fall ist. Für diese Arbeit soll ein Vergleich aufgestellt werden, der unterschiedliche Preis- und Aufwandssegmente abdeckt. Dafür wird zwischen einer kostenlosen Variante, einer kostenfreien Sandbox mit einhergehendem Implementierungs- und Konfigurierungsaufwand, und einer kommerziell erwerblichen Sandbox unterschieden. Das Ziel dieser Gegenüberstellung ist es, ein Verständnis zu erlangen, welche Features in den jeweiligen Preisklassen der ausgewählten Sandboxen vorhanden sind. Anhand dieser Informationen soll sich eine Entscheidung treffen lassen, ob es sich lohnt, den Aufwand der Implementierung auf sich zu nehmen oder die kommerziellen Versionen zu erwerben. Je nach Preis und Aufwand kann entschieden werden, welche Sandbox für den angestrebten Nutzen die beste Wahl ist. Die Entscheidung zwischen Kosten, Aufwand und Nutzen soll vereinfacht werden, indem bestimmte Metriken zur Messung der Sandbox als Matrix veranschaulicht werden. Die dafür ausgewählten Mess-Metriken werden in Kapitel 5.1 dargestellt und erläutert.

Es wurden die folgende Sandbox Systeme für den Vergleich ausgewählt: Cuckoo wurde als Repräsentant für die kostenlose Sandbox, welche selbst implementiert und konfiguriert werden muss, ausgewählt. Weiterhin wurde Any.Run als Vergleichspartner für die Kategorie der kommerziellen Sandboxen gewählt, wobei hier die Version mit der höchst möglichen Feature Anzahl genutzt wurde. Zuletzt fiehl die Wahl auf die Sandbox von Hybrid Analysis, welche keine Implementierung benötigt und kostenlos ist, womit sie jedermann zur Verfügung steht. Jede dieser Sandboxen verfügt über einige besondere Eigenschaften, welche die anderen Vergleichspartner nicht mit sich bringen, weshalb diese ausgewählt wurden. Im Folgendem wird auf Vorteile, sowie Nachteil der einzelnen Produkte eingegangen, welche die einzelnen Charakteristiken der jeweiligen Sandbox spezifizieren. [42, S. 20] [9] [45]

	Cuckoo	Any.Run	Hybrid Analysis
Kosten	Kostenlos	Kostenpflichtig	Kostenlos
Open Source	✓	x	x
Online Sandbox	x	✓	✓
Betriebssysteme			
Windows XP	✓	x	x
Windows 7	✓	✓	✓
Windows 10	x	✓	✓
Windows 11	x	✓	x
Linux	✓	x	✓
MacOS	✓	x	x
Android	✓	x	✓
MITRE ATT&CK	x	✓	✓
Sample Veröffentlichung	x	x	✓
Implementierungs- und Konfigurierungsaufwand	✓	x	x
Kommentar	Quellcode anpassbar Verwendung Python 2	Ebenfalls in Kostenloser Version verfügbar	In kommerzieller Version verfügbar

Abbildung 5.1: Charakteristika der ausgewählten Sandbox Systeme

5.0.1 Cuckoo

Die erste Sandbox ist unter dem Namen Cuckoo bekannt. In dieser Arbeit repräsentiert diese Sandbox den kostenlosen Vergleichspartner, welcher jedoch mit dem Implementierungs- und Konfigurationsaufwand einhergeht. Cuckoo ist eine Open Source Sandbox, welche in verschiedenen Versionen existiert und in Python programmiert wurde. Open Source bedeutet, dass der Code öffentlich verfügbar ist. [46] [33, S. 2, 3] [42, S. 20]

Durch ihren Feature Umfang und die Anpassbarkeit ist Cuckoo eine der bekanntesten Sandboxen für die dynamische Malware Analyse. Deshalb wird diese Sandbox in vielen Vergleichen aufgegriffen. Die im Moment am weitesten entwickelte Version ist Cuckoo 2 und die neuste, jedoch sehr wenig dokumentierte Version ist Cuckoo 3. In Cuckoo 3 gab es einige Veränderungen, zum jetzigen Zeitpunkt ist die Implementierung überaus aufwendig, da sie bisher nicht offiziell von Cuckoo veröffentlicht wurde. Was dazu führt das für diese Bachelorarbeit Cuckoo 2 verwendet wurde. [47] [27, S. 7]

Cuckoo bietet einige Vorteil gegenüber ihren Konkurrenten. Der wahrscheinlich größte Vorteile davon ist der Fakt das Cuckoo Open Source basiert ist. Damit ist sie kostenlos verfügbar, muss jedoch selbst implementiert und konfiguriert werden. Es ist eine umfangreiche Dokumentation für die Erstellung einer Cuckoo Sandbox vorhanden. Neben dem nicht existierenden Kostenfaktor ist positiv zu erwähnen, dass der Code von Cuckoo öffentlich verfügbar und somit auch veränderbar ist. Das bedeutet, der Code kann so bearbeitet werden, dass er den Bedürfnissen des Nutzers entspricht und es können sogar neue Features implementiert werden. [42, S. 20] [46] [37, S. 31] [27, S. 116–124] [8]

Für diese Arbeit kam eine unmodifizierte Cuckoo Sandbox zum Einsatz. Ein weiterer Vorteil ist, dass diese Sandbox in der Lage ist, unterschiedlichste Betriebssysteme zu nutzten, um Malware zu analysieren, wenn diese von dem Benutzer konfiguriert wurden. Dazu gehören neben Windows weiterhin Linux, MacOS und Android. [8] [48]

Das Problem an dieser Sandbox ist der Aufwand, diese zu implementieren und anschließend zu konfigurieren. Die Implementation und Konfiguration ist sehr umfangreich und zeitaufwändig. Ein weiteres Problem ist die verwendete Python Version die benutzt wird. Es handelt sich um Python Version 2.7, welche nicht mehr aktuell ist, da mittlerweile Version 3 genutzt wird. Python 2 ist technisch veraltet, was dazu führt, dass diese Sprache keinen Support mehr erhält, da die letzte Version 2010 veröffentlicht wurde und seit 2020 nicht weiter entwickelt wird. Deshalb unterstützen die meisten Python Bibliotheken nur noch die neuere Python Version und nicht Python Version 2, wodurch diese nicht mehr effektiv verwendbar ist. Bei Python Bibliotheken handelt es sich um eine Gruppe aus verwandten Code-Modulen. Diese können in den eigenen Programmen genutzt werden, um Programmierung schneller und einfacher zu gestalten. [42, S. 20] [49] [50] [51]

5.0.2 Any.Run

Bei Any.Run handelt es sich um eine öffentliche Sandbox. Diese kann über das Internet erreicht werden. Die Any.Run Sandbox ist in verschiedenen Preisklassen erwerblich. Darunter zählen neben der kostenlosen Variante drei weitere Varianten mit unterschiedlichen Features. Eine öffentliche Sandbox wird durch die Beiträge der Kunden immer auf dem neusten Stand gehalten. Das bedeutet, dass diese Sandboxen mit den aktuellsten Analyse Techniken arbeiten und keinerlei Aufwand bei der Implementierung für den Benutzer entsteht. Wie bereits erwähnt, standen eine sehr große Anzahl an Auswahlmöglichkeiten zur Verfügung. Das machte es sehr anspruchsvoll eine Entscheidung zu treffen. Die Wahl fiel schließlich auf Any.Run, da diese Sandbox eine große Anzahl vielversprechender Features aufwieß, darunter auch einige Mechanismen, um ausweichende Malware zu umgehen und eine benutzerfreundliche Darstellung der gesammelten Informationen, um einen einfachen Einstieg zu gewährleisten. [52] [53] [54] [13, S. 640] [39]

Any.Run hat sehr nützliche Features, um beispielsweise die Analyse von Malware zu vereinfachen oder Informationsextraktion sogar erst zu ermöglichen. Darunter zählt zum Beispiel die Funktion der Live Interaktion, mit welcher der Nutzer in das Geschehen der Analyse eingreifen kann. Dieses Feature ist überaus nützlich wenn es darum geht Malware zu analysieren, welche auf bestimmte Aktionen wartet. Bei diesen Aktionen handelt es sich um von dem Benutzer ausgeübt Mausbewegungen oder Klick-Verhalten, bevor die Malware in ihrem vollen Funktionsumfang ausgeführt wird. Diese Evasion Techniken wurden bereits in Kapitel 4.3 näher beschrieben. Weiterhin verfügt die Sandbox der Marke Any.Run über vorinstallierte Programme, um ausweichende Malware zu täuschen, sollte diese auf die System Umgebung achten, wie in Kapitel 4.3 beschrieben. [52] [9] [54]

Außerdem sind die von Any.Run bereitgestellten Informationen über eine Malware sehr umfangreich und gut verständlich. In Kapitel 4.1 wurde bereits erwähnt, dass einige öffentliche Sandbox Systeme die Ergebnisse einer Analyse abspeichern. Im Fall von Any.Run werden

die Ergebnisse nur abgespeichert und öffentlich zugänglich gemacht, wenn die kostenlose Version verwendet wird. Andernfalls können andere Nutzer die Ergebnisse nicht einsehen. Weiterhin ist positiv zu erwähnen, dass bei Any.Run MITRE ATT&CK zum Einsatz kommt. MITRE ATT&CK ist eine Wissensdatenbank, welche Informationen über Angriffstechniken und Taktiken enthält. Durch das Wissen über die eingesetzten Taktiken und Techniken ist es einfacher Malware Samples miteinander zu vergleichen. Das Wissen hilft zu verstehen, welche Techniken eingesetzt wurden und wie der Angriff statt fand. Auf das MITRE ATT&CK Framework wird in Kapitel 5.1 weiter eingegangen. [53] [55] [9] [52] [39]

Ein Nachteil hingegen ist, dass diese Sandbox lediglich Windows Betriebssysteme für die Beobachtung und Analyse der Sandbox verwendet. Dabei kann auf Windows 7, Windows 8, Windows 10 und Windows 11 zurückgegriffen werden. Malware kann also nicht auf anderen Betriebssystemen wie Android, Linux oder MacOS untersucht werden. Ein weiterer Nachteil in der Gegenüberstellung zu den anderen Sandboxen in diesem Vergleich ist, dass diese wie bereits erwähnt kommerziell erworben werden muss. [39] [9]

5.0.3 Hybrid Analysis

Bei der dritten und letzten Vergleichssandbox kam Hybrid Analysis zum Einsatz. Die Entscheidung für diese Sandbox fiel aufgrund der Möglichkeit eine kostenlose Version nutzten zu können. Dabei kam es zu Einschränkungen einiger Funktionalitäten, was für die Gegenüberstellung zu den kostenpflichtigen oder Implementierungsintensiven Konkurrenten wichtig ist. Somit können die Funktionsunterschiede der einzelnen Preisklassen möglicherweise besser hervorgehoben werden. Was können aktuelle Sandboxen mit vollem Funktionsumfang bieten, wozu kostenlose Sandbox Versionen nicht in der Lage sind. Damit kann besser über den Nutzen gegenüber den Kosten oder dem Aufwand entschieden werden. [9]

Die Sandbox von Hybrid Analysis ist, wie es bei Any.Run der Fall ist, eine öffentliche Sandbox. Wie bereits erwähnt, besteht bei öffentlichen Sandbox Systemen das Problem der Veröffentlichung von durchgeführten Analysen. Dies ist meist nicht zutreffend, wenn die Vollversion einer Sandbox genutzt wird. In diesem Fall handelt es sich jedoch um eine kostenlose Plattform, was bedeutet, dass die gesammelten Daten gespeichert werden. Es werden also auch vertrauliche Daten gespeichert und öffentlich zugänglich gemacht. In diesem Fall ist positiv zu erwähnen, dass diese Sandbox, obwohl sie ein kostenloses Produkt ist, über eine Vielzahl an Betriebssystemen zur Beobachtung und Analyse von Malware verfügt. Neben Windows 7 und Windows 10 ist ebenfalls Linux verfügbar. Auch Android wird unterstützt, jedoch ist hier bisher nur eine statische Analyse verfügbar. Wie es bereits bei Any.Run der Fall war, bindet auch die Hybrid Analysis Sandbox MITRE ATT&CK in ihre Analyse mit ein, um mehr Informationen über die Bedrohung preiszugeben. [9] [45] [10, S. 70–72, 74] [35]

5.1 Vergleichsmetriken

Es gibt keine Sandbox, die jede Malware erkennen kann. Jedoch ist es wichtig, die Sandbox mit dem höchst möglichem Schutz vor Bedrohungen auszuwählen. Abgesehen von dem Schutz sind noch andere Faktoren wichtig, um sich für ein Produkt zu entscheiden. Dabei ist die Wichtigkeit einer jeden Metrik, je nach gewünschtem Einsatz, unterschiedlich. Die Auswahl der Metriken erfolgt anhand der Erfahrungen anderer Experten und mithilfe der eigenen Kenntnisse, welche während des Prozesses der Sandbox Test erlangt wurden. Auf diese ausgearbeiteten Metriken soll folgend eingegangen werden und erklärt werden, warum es zur Auswahl dieser entsprechenden Metrik kam, beziehungsweise worin ihr Nutzen liegt.

Ergebnisse
Es ist sehr wichtig detaillierte Berichte von der Sandbox zu erhalten, damit sichergestellt werden kann, dass keine wichtigen Informationen übersehen wurden. Dabei ist wichtig, dass Informationen über die getestet Datei, beobachtetes Verhalten, Registry Änderungen, Prozesse und vor allem die Bösartigkeit dargestellt werden. Wie in Kapitel 3.1 beschrieben, werden die gesammelten Sandboxdaten anschließend für die Erzeugung der Signaturen benutzt, welche die Erkennung von Malware ermöglichen. Fehlende Informationen könnten dazu führen, dass Bedrohungen später nicht erkannt oder Programme von der Sandbox fälschlicherweise als schädlich eingestuft werden. Es ist wichtig im Incident Response schnell auf eine Bedrohung reagieren zu können. Wenn nun der Bericht zu viele überflüssige Informationen enthält können die wichtigen Signale nicht schnell genug erkannt werden. [41] [13, S. 84] [26, S. 4] [11, S. 9]

Weiterhin sollten verschiedene Formate angeboten werden, in denen der Bericht heruntergeladen werden kann, um unterschiedlichsten Geräten und Plattformen die Arbeit mit diesen zu ermöglichen. JavaScript Object Notation (JSON) Dateien werden beispielsweise von Anwendungen zum Austausch von Daten genutzt. Diese werden von den meisten Betriebssystemen, Programmiersprachen und Browsern unterstützt, was bedeutet, dass sie effizient weiter verarbeitet werden können. Ebenfalls ist es sinnvoll die Berichte als PDF bereitzustellen, da dieses Format das am meist genutzte ist. [56] [13, S. 98] [8]

Open Source
In Kapitel 5.0.1 wurde bereits der Begriff 'Open Source' erklärt. Es ist praktisch eine Sandbox zu nutzen, welche auf die individuellen Ansprüche einer Firma oder eines Benutzers angepasst werden kann. Damit ist es möglich die individuellen Wünsche, welche möglicherweise von keiner Sandbox erfüllt werden, selbst zu implementieren. Die fehlenden Funktionalitäten können hinzugefügt werden, indem der Quellcode der Open Source Sandbox verändert und angepasst wird. Außerdem kann genau nachvollzogen werden, was bei dem Ablauf der Sandbox passiert und wohin Daten fließen. [46] [8, S. 20] [37, S. 31]

Evasion Resistenz
Evasion Techniken sind sehr weit verbreitet und in großer Anzahl zu finden. Damit die Sandbox eine Evasive Malware überhaupt als solche erkennt, muss sie über bestimmte Mechanismen verfügen, die es ihr ermöglichen, diese Techniken zu umgehen beziehungsweise

dagegen zu wirken. Wenn diese Mechanismen nicht vorhanden sind, wird der Schadcode der Malware möglicherweise nicht ausgeführt oder es tritt ein verändertes Verhalten auf. Somit kann es dazu kommen, dass wichtige Informationen verloren gehen und die Malware fälschlicherweise als harmlos klassifiziert wird. Der Grund für diese Falschklassifizierung liegt daran, dass die Malware kein bösartiges Verhalten ausgeführt. Beim Vergleich der Sandbox Systeme wurden einige der am meist vertretensten Evasion Wiederstands-Techniken genutzt. [8] [7, S. 7, 9–10, 12–13] [31, S. 3] [57, S. 1] [6, S. 1]

Skalierbarkeit
Es kann sinnvoll sein, je nach Umfang täglicher durchgeführter Malware Analysen, eine Sandbox auszuwählen, mit der mehrere Malware Samples parallel analysiert werden können. Anhand der parallelen Analyse kann Zeit eingespart werden, welche für andere Dinge genutzt werden kann. Dadurch können beispielsweise die Berichte bereits ausgewerteter Malware Samples verarbeiten werden, während die Untersuchung weiterer Malware ausgeführt wird. [8]

Betriebssystem Unterstützung
Es ist wichtig die Möglichkeit zu haben, Malware auf mehreren Betriebssystemen unterschiedlicher Firmen zu testen. Angriffe werden auf jedem Betriebssystem ausgeführt, dabei ist es egal ob es sich um Windows, Linux, Android oder MacOS handelt. Windows ist neben Android das am meisten verwendete Betriebssystem. Auch wenn das Betriebssystem von Android öfter genutzt wird, wie in Abbildung 5.2 sichtbar ist, kommen Angriffe auf diese Betriebssystem laut einer Statistik der Firma Devcon sehr selten vor. Dabei wird sich speziell auf bösartige Werbung als Methode eines Angriffes bezogen. Etwas mehr als zwei Prozent der bösartigen Werbungskampagnen sind gegen Android gerichtet. Bei Windows hingegen handelt es sich um das am meisten betroffene System, da 61 Prozent der Angriff gegen dieses Betriebssystem gerichtet sind. [58] [26, S. 7]

Das ist der Grund, warum für diesen Vergleich verschiedene Windows Betriebssysteme genutzt wurden, nicht jedoch Linux, Android oder MacOS. Durch verschiedene Versionen der Windows Systeme können unterschiedliche Informationen sichtbar werden. Nur Linux Systeme sind noch seltener das Ziel der bösartigen Werbung, als es Android ist. Das bedeutet jedoch nicht, dass der Fall niemals eintreten wird. Es ist ebenso wichtig neben Windows auch die anderen drei am meisten verbreiteten Betriebssysteme nicht zu vernachlässigen. Sobald eine Sandbox eine Malware, welche beispielsweise für Android entwickelt wurde, untersuchen muss und kein Android Betriebssystem unterstützt, kann die Analyse nicht ausgeführt werden. Es ist nicht möglich eine Datei mit der Endung 'Android Package (APK)' auszuführen, da diese nur auf Android ohne weitere Umstände ausgeführt werden kann. [26, S. 7] [8] [58] [59]

URL Analyse
Kapitel 2.2.2 zeigt das Malware oft über Internetseiten verbreitet wird. Es ist wichtig diese Internetseiten vorzeitig erkennen zu können, um so eine Infizierung durch Malware zu verhindern. Dafür können die URLs der Websites mithilfe einer Sandbox überprüft werden. Anschließen wird erkenntlich, ob es sich dabei um bösartige Website handelt oder ob dies nicht der Fall ist. Deshalb ist neben der Untersuchung von Dateien auch die Analyse von URLs hilfreich. [8]

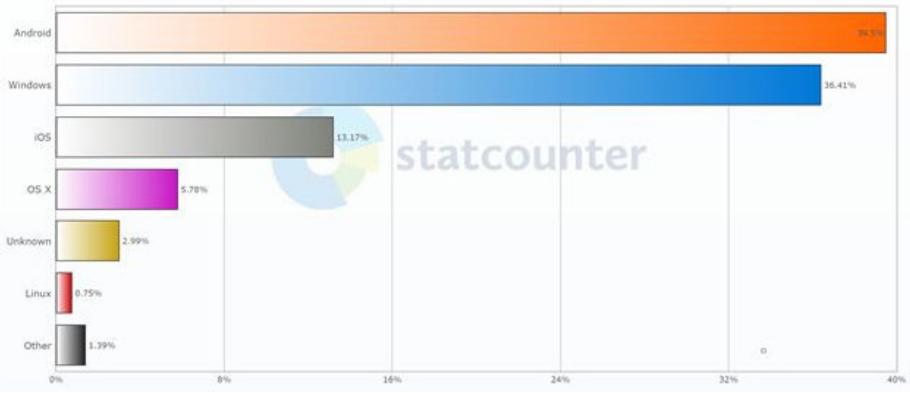

Abbildung 5.2: Nutzung verschiedener Betriebssysteme im Jahr 2018 [60]

Automatisierung

Immer mehr Firmen versuchen ihre Abläufe zu automatisieren, damit zeitraubende Aufgaben effizienter erledigt werden können und Personal mehr Zeit für anspruchsvollere Aufgaben hat. Dies ist ebenfalls für die Malware Analyse gewünscht, um so die zu untersuchenden Dateien automatisiert an die Sandbox zu senden und anschließend den Bericht zu erhalten. Somit kann sich der Mitarbeiter auf die Recherche konzentrieren. Ein weiterer Grund für die Automatisierung von Malware Analyse ist, dass die immer weiter ansteigende Zahl von Malware Angriffen, nicht per Hand analysiert werden kann. Dafür können beispielsweise Representational State Transfer (REST) APIs eingesetzt werden. Bei einem REST API handelt es sich um einen Architekturstil für eine Programmschnittstelle auch als API bekannt. Dabei werden Hypertext Transfer Protokoll (HTTP)-Anfragen genutzt, um Daten zu verwenden und darauf zugreifen zu können. [52] [61] [62] [3, S. 13]

Implementierung und Wartung

In manchen Fällen soll die Sandbox direkt implementiert sein und keinerlei Wartungsarbeit für den Benutzer anfallen, wie es bei online Sandboxen der Fall ist. Auf diese wurde bereits in Kapitel 4.1 eingegangen. Aus eigenen gesammelten Erfahrungen wird eine gewisse Zeit benötigt, um eine Sandbox zu implementieren, wie in Kapitel A nachvollzogen werden kann. Außerdem kommt es oft zu auftretenden Fehlern. Um diese zu beheben, müssen gewisse Kenntnisse über die zugrunde liegende Technik vorhanden sein. Sollten diese Grundvoraussetzungen nicht erfüllt sein, ist es sinnvoll, zu einer online Sandbox zu greifen. Diese Form der Sandbox ist zeitsparend und es wird kein spezifisches Wissen für die Erstellung und Konfiguration benötigt. Um eine Sandbox implementieren zu können, wird ebenfalls leistungsfähig Hardware benötigt, worüber nicht jeder Anwender verfügt. [33, S. 2] [48] [42, S. 20]

Statische Analyse

Die statische Analyse ist in der Lage Informationen zu liefern, welche die dynamische Analyse nicht liefern kann. Diese können eine gute Erweiterung zu den durch die dynamische Analyse gesammelten Informationen sein und neues wichtiges Wissen liefern. Dabei geht es vor allem um Informationen, die aus dem Maschinencode gewonnen werden können, was mithilfe der dynamische Analyse nicht möglich ist. Die anhand der dynamischen Analyse gesammelten Informationen begrenzen sich auf die Beobachtung des Verhaltens der Malware. Bei dem Einsatz von Hybrider Analyse kann von den Vorteile beider Techniken profitiert werden. [13, S. 22, 28, 110] [6, S. 6] [28, S. 1667]

Internet Verbindung
Diverse Malware benötigt eine Internetverbindung, um mit dem Angreifer zu kommunizieren. Ein Beispiel dafür ist der C2C Server, welcher von dem Angreifer genutzt werden kann, um die Malware zu befehligen und kontrollieren. Damit können verschiedene schädliche Aktivitäten ausgeführt werden. Sollte die Malware nicht in der Lage sein mit dem Angreifer in Kontakt zu treten, könnte das dazu führen, dass die Malware kein bösartiges Verhalten aufweist. Ohne eine Internetverbindung kann bösartige Netzwerkaktivität nicht analysiert werden und somit würde die Analyse einiger Malware Samples falsche Ergebnisse liefern. [2, S. 237–238] [13, S. 65] [31, S. 3] [10, S. 37] [63]

FakeNet
Bei einigen Malware Implementierungen kann es vorkommen dass diese, sobald sie keine Antwort auf gesendete Anfrage erhalten, verändertes Verhalten aufweisen. Um das zu verhindern kann die Funktion von FakeNet hinzugezogen werden. Diese wird genutzt, um automatische Antworten auf diese Anfragen zu erstellen. Solche automatischen Antworten werden erstellt, indem echte Netzwerkdienste simuliert werden und Netzwerkverkehr umgeleitet wird. Sollte eine Malware also ein Anfrage an eine Seite stellen, wird diese mit dem HTTP Statuscode 404 beantwortet. Das bedeutet, dass die angefragte Seite nicht gefunden werden konnte. Entweder wurde die Seite auf eine andere URL verlegt oder entfernt. Sobald die Malware diesen Statuscode als Antwort erhält, wird sie eine Anfrage an die nächste Website senden. Mithilfe der durch diese Option gesammelten Daten, können weitere mögliche bösartige Websites erkannt werden. [64] [63]

MITRE ATT&CK
Wie bereits aus Kapitel 5.0.2 bekannt, nutzen einige Sandbox Systeme das MITRE ATT&CK Framework, um ein besseres Verständnis über die vollzogenen Schritte und deren Ziele zu erhalten. Außerdem hilft es bei der Klassifizierung und Kategorisierung der Malware. Dafür stellt MITRE die von der Malware genutzten Taktiken und die dafür verwendeten Techniken dar. Zu den Taktiken gehören die einzelnen Stufen des Angriffes und die Ziele der jeweiligen Stufe. Es ist wichtig genügend Informationen über eine Malware zu haben. Ein Incident Response Team beispielsweise benötigt die gewonnen Informationen, um eine Eindämmung der Infektion durchzuführen und anschließend die involvierten Artefakte zu bestimmen. Außerdem werden mittels der gesammelten Daten Signaturen erstellt, um weitere Infizierung zu erkennen. [11, S. 85] [2, S. 18] [53] [10, S. 233–234]

6 Ergebnisse

Die für den Vergleich ausgewählten Metriken, sind ausschlaggebend für die Entscheidung einer Sandbox. Jedoch ist es essentiell, dass die Sandbox über die wichtigsten Informationen verfügt, um eine Malware identifizieren und eliminieren zu können. Bei diesen Informationen handelt es sich unter anderem, um das Verhalten der Prozesse, der Windows Registry, des Datei System, des Netzwerkverkehrs und der Dienste. Die Wichtigkeit dieser Informationen wurde in Kapitel 4 bereits erläutert. [13, S. 14, 84, 224]

Die ausgewählten Sandboxen wurden unter Verwendung unterschiedlicher Malware getestet, um schlussendlich einen aussagekräftigen Vergleichsdatensatz zu erhalten.

Bei den Tests wurden 25 verschiedenen Malware Samples in jeder der drei Sandbox Systeme ausgeführt und analysiert. Für die Analyse wurde bei jeder Sandbox das Betriebssystem Windows 7, sowie falls möglich Windows 10 genutzt, um feststellen zu können in welchem Betriebssystem die Ausführung der Malware die meisten Informationen liefert. Es wurde lediglich Windows benutzt, da es das einzige von allen Sandboxen unterstützt Betriebssystem war. Dabei war ersichtlich, dass die Verwendung verschiedener Windows Versionen, ebenso verschiedene Ergebnisse hervorbrachte. Das bedeutet, dass die Analyse auf älteren Systemen wie Windows 7 von Nutzen sein kann, um so eine größere Palette an Informationen zu generieren. Außerdem wurden fünf bösartige URLs ausgewählt, welche anschließend für den Vergleich in jeder Sandbox analysiert wurden. Damit konnte festgestellt werden, wie jede Sandbox in dem Aspekt der URL Analyse abschneidet.

Bei der ausgewählten Malware wurde eine Mischung aus aktuell verwendeten und den meist verwendeten Samples gewählt. Es handelte sich beispielsweise um Malware wie 'WannaCry', 'Agent Tesla' und 'Formbook'. Dabei war es von Bedeutung neben der Aktualität und Nutzung der ausgewählten Malware, ebenfalls Malware auszuwählen, welche auf eine Vielzahl an Evasion Techniken zurückgreift. Dies war von Bedeutung, um feststellen zu können, ob die getesteten Sandbox Systeme in der Lage sind ausweichende Malware zu erkennen. Ein Beispiel für eine solche Malware stellt 'Formbook' dar. Jede der analysierten Malware Samples nutzte Evasion Techniken, jedoch unterschied sich das Maß der Nutzung zwischen den einzelnen Malware Varianten. Zuletzt sollte erwähnt werden, dass keine unbekannte Malware, sondern lediglich bereits von Sandbox Systemen erkannte Malware analysiert wurde. Somit ist unklar, wie effektiv die ausgewählten Sandbox Systeme in der Erkennung von unbekannter Malware sind. Weitere Ergebnisse der Untersuchungen werden im Folgenden näher ausgeführt. [65] [66]

Angesichts der zahlreichen Techniken zur Umgehung ausweichender Malware, wird an dieser Stelle darauf hingewiesen, dass in dieser Arbeit lediglich vereinzelte Techniken vorgestellt wurden. Bei den ausgewählten Anti-Evasion Techniken wurde sich auf einige der bekanntesten und effektivsten Vertreter beschränkt. Zu diesen Techniken zählen die Live Interaktion, die Möglichkeit des Neustarts, der Einsatz automatisierter Mausbewegungen und vorinstallierte Software. [6, S. 5] [41, S. 2–7] [9]

	Any.Run	Cuckoo	Hybrid Analysis
Berichte			
Details	✓	✓	✓
Export Formate	JSON, PDF, HTML	JSON, HTML, PDF	HTML, PDF
Open Source	x	✓	x
Evasion Resistenz			
Neustart	✓	x	x
Live Interaktion	✓	✓	x
Automatisierte Mausbewegungen	x	✓	✓
Vorinstallierte Software	✓	✓	✓
Unterstütze Betriebssysteme			
Windows 7	✓	✓	✓
Windows 10	✓	x	✓
Windows 11	✓	x	x
weitere Windows Betriebssysteme	✓	✓	x
Linux	x	✓	✓
Android	x	✓	x
Mac OS	x	✓	x
URL Analyse	✓	✓	✓
Implementierungs- und Konfigurierungsaufwand	x	✓	x
Statische Analyse	✓	✓	✓
Skalierbarkeit	✓	x	x
Internet Verbindung	✓	✓	✓
FakeNet	✓	✓	✓
Automatisierung	✓	✓	✓
MITRE ATT&CK	✓	x	✓

Abbildung 6.1: Matrix für den Vergleich der vorgestellten Sandbox Systeme

6.1 Cuckoo

Cuckoo stellt ausgewertete Ergebnisse einer getesteten Malware sowohl online als auch als Download zur Verfügung. Dabei kann zwischen dem JSON-Format und dem HTML-Format ausgewählt werden. Es ist zu beachten, dass die HTML Datei ebenfalls in eine PDF Datei umgewandelt werden kann. Bei Bedarf können Informationen aus dem herunterzuladenden Bericht ausgelassen werden, um ausschließlich die wichtigsten Fakten einzubringen. Neben den Berichten können weitere gesammelte Daten für die Auswertung herangezogen werden. Dazu zählen beispielsweise Mitschnitte des Netzwerkverkehrs, erstellte Log Dateien und Informationen der statischen Analyse. Außerdem beinhaltet ein Bericht Informationen über beobachtetes Verhalten von beispielsweise Diensten, Prozessen und Dateien. Wichtig zu erwähnen ist, dass der Aufbau des online Berichtes wesentlich übersichtlicher ist, als es bei dem heruntergeladenen Dokument der Fall ist. Das liegt daran, dass dieser über einen großen Umfang an Seiten verfügen kann. Der online Bericht hingegen ist in einzelne Abschnitte geteilt, welche die Informationen separieren. [10, S. 40–41, 48–49] [67, S. 19]

Bei den durch die statischen Analyse gewonnen Daten handelt es sich unter anderem um Informationen aus Anti-Viren Programmen, Informationen über die ausführbare Datei und von der Malware genutzte Imports. Bei Imports handelt es sich, um die bereits in Kapitel

5.0.1 erklärten Bibliotheken, welche von der Malware eingebunden wurden. Ein Beispiel für eine solch importierte Funktion ist 'CreateProcessA', welche genutzt wird um einen neuen Prozess zu erstellen. Diese Imports geben einen Hinweise darauf, welche Aktivitäten von der Malware ausgeführt werden könnten. [13, S. 39, 41]

Um Informationen über den Netzwerkverkehr zu sammeln, besteht bei Cuckoo die Option, eine echte Internetverbindung zu nutzen. Damit kann die Malware das Verhalten ausführen, welches vom Angreifer beabsichtigt ist. Folglich ist eine Beobachtung des authentischen und realen Verhaltens der Malware möglich. Dadurch kann diese beispielsweise mithilfe eines C2C Servers Kontakt mit dem Angreifer aufnehmen. Diese Information können wichtig sein, um den Angreifer identifizieren und entsprechende Gegenmaßnahmen treffen zu können. Neben der echten Internetverbindung kann ebenfalls eine simulierte Internetverbindung mit vordefinierten Antworten auf Anfragen der Malware genutzt werden. Somit ist es möglich, alle Verbindungen, die von der Malware aufgebaut werden, nachzuvollziehen. Aufgrund der vorgefertigten Antworten stellt die Malware an alle bösartigen Adressen HTTP-Anfragen, bis alle angefragten Seiten einen 404 Statuscode zurück schicken, wie es bereits in Kapitel 5.1 erwähnt wurde. [10, S. 16, 48] [2, S. 27] [13, S. 68] [27, S. 44] [68]

Cuckoo bindet MITRE ATT&CK nicht in ihre Berichte mit ein, wie es bei den beiden Konkurrenten der Fall ist. Das bedeutet, dass in den Berichten keine Informationen bezüglich der während des Angriffes verwendeten Taktiken und Techniken vorhanden sind. Diese Informationen können jedoch ein besseres Verständnis des Angriffes ermöglichen. Um diese Informationen zu erlangen, muss der Analyst selbst die Zuordnung der einzelnen Stufen des Angriffes zu den Taktiken vornehmen. Diese Zuordnung raubt dem Analysten jedoch wertvolle Zeit. Zuletzt ist anzumerken, dass Cuckoo keine Zusammenfassung über die gesammelten IoCs, wie IPs, Domänen und URLs abbildet. Eine solche Zusammenfassung wird bei Any.Run genutzt, um einen schnellen Überblick über die gesammelten Artefakte, die auf einen Angriff hindeuten, zu erhalten. Bereits in Kapitel 5.0.1 wurde erwähnt, dass einer der größten Vorteile von Cuckoo die Open Source Basierung ist. Damit können mithilfe der benötigten Kenntnisse unterschiedlichste Funktionen hinzugefügt und angepasst werden. Dies ist unter der Verwendung von online Sandbox Systemen nicht möglich. Sollte bei einer solchen Sandbox ein Feature fehlen, muss dies in Kauf genommen oder eine andere Sandbox gewählt werden. [10, S. 233–234] [53] [46] [33, S. 2]

Cuckoo ist in der Lage einige Anti-Evasion Techniken anzuwenden, um die Ausführung der Malware zu ermöglichen. Dazu zählen Live Interaktion, der Einsatz automatisierter Mausbewegungen und vorinstallierter Software. Der Einsatz von Live Interaktion mit der Malware kann gegen viele Evasion Techniken sehr effektiv sein. So kann ausweichende Malware ausgeführt werden, welche sich vor allem auf die Überprüfung menschlicher Interaktionen beschränkt. Cuckoo ist nicht in der Lage, einen Neustart während der Analyse auszuführen, was bedeutet, dass der Schadcode bestimmter Malware Samples nicht ausgeführt werden kann. Bei Bedarf ist es möglich, die Option für automatisierte Mausbewegung zu aktivieren oder zu deaktivieren. Die Deaktivierung ist empfehlenswert, sobald die Live Interaktion genutzt wird. Automatisierte Mausbewegung ermöglicht es, das Mausbewegungsverhalten eines Benutzers zu imitieren, worauf bestimmte Evasive Malware ihren Fokus richtet. [9] [6, S. 5] [7, S. 10] [69]

Eine weitere von Cuckoo getroffene Maßnahme ist die Installation von Software auf der Gast Maschine, um ein Benutzersystem nachzuahmen. Betrachtet man die anderen beiden Sandboxen, dann liegt der Unterschied der vorinstallierten Software bei Cuckoo darin, dass diese eigenständig vom Nutzer installiert werden muss. Dadurch kann der Analyst selbst über die zu installierende Software entscheiden, wie in Kapitel A erwähnt. Somit kann die Software angepasst werden, je nachdem welche Programme momentan auf der Mehrzahl von Systemen vorhanden sind. Wie aus Kapitel 4.3 bekannt, gibt es Malware Varianten, welche nach spezieller Software Ausschau halten, bevor sie ihren Schadcode ausführen. Deshalb ist es sinnvoll, Programm wie PDF Reader oder Microsoft Office zu installieren. Der Nachteil dieser eigenen Auswahl an Software ist, dass bestimmte Programm wie Microsoft Word oder Excel kostenpflichtig sind. Der eigentliche Nachteil, den diese Sandbox mit sich bringt, ist der Aufwand der Implementierung und Konfiguration. Die Implementierung und Fehlerbehebung ist sehr zeitintensiv und muss von dem Anwender vorgenommen werden. Beim Prozess der Implementierung können die unterschiedlichsten Fehler auftreten. Diese sind schwierig zu finden und stellen selbst für Experten manchmal ein Problem dar. In einigen Fällen muss der Prozess sogar komplett neu begonnen werden. Es ist keine simpler Vorgang eine solche Sandbox selbst bereitzustellen. Dieser Nachteil wurde bei eigener Anwendung festgestellt, wie in Kapitel A zu lesen, und ist außerdem Erfahrungswert anderer Analysten. [10, S. 93] [42, S. 22] [43] [44] [42, S. 20] [49]

Neben der Implementation wird leistungsfähige Hardware für eine Sandbox benötigt. Erst dann ist die Sandbox überhaupt in der Lage, eine Analyse durchzuführen. Je nach Einsatz benötigt sie ausreichend Speicherplatz, RAM und auch CPU Kerne. Diese Eigenschaften sind dabei nicht nur für die Leistungsfähigkeit wichtig, sondern auch, um Evasion Techniken zu umgehen. Wurden einer Sandbox zu niedrige Ressourcen zugeordnet, kann dies bedeuten, dass sie als eine solche entlarvt wird, wie es bereits in Kapitel 4.3 erläutert wurde. [33, S. 2] [7, S. 12] [10, S. 4]

Bei den unterstützten Betriebssystemen ist Cuckoo seinen Konkurrenten überlegen. Diese Sandbox ist in der Lage Analysen sowohl auf Windows, Linux, MacOS und sogar Android Betriebssystemen durchzuführen. Damit deckt sie mehr Betriebssysteme als beide anderen Konkurrenten ab. Bei dem Einsatz von Windows muss sich jedoch auf Windows 7 und Windows XP verlassen werden. Es existiert zu diesem Zeitpunkt auf der Cuckoo Website keinerlei Dokumentation, um die Einrichtung einer Windows 10 oder sogar Windows 11 Gast Maschine für die Analyse zu ermöglichen. Da es sich bei diesen Betriebssystemen um die aktuellsten und meist genutzten Versionen handelt, wird diverse Malware explizit für diese Plattform entwickelt. Damit kann die best mögliche Analyse diverser Malware nicht gewährleistet werden. Da Windows 10 und 11 nicht genutzt werden können, kann es zu fehlenden Informationen oder anderem Verhalten kommen. Wichtig zu erwähnen ist, dass bei der großen Auswahl an Betriebssystemen jede einzelne Gast Maschine mit einem neuem Betriebssystem selbst installiert, konfiguriert und anschließend in den Cuckoo Konfigurationsdateien eingebunden werden muss. [70] [68] [48]

Die Analyse und Auswertung von URLs ist mithilfe von Cuckoo problemlos möglich. Dabei war die Sandbox in der Lage, die Schadhaftigkeit jeder getesteten URL korrekt zu klassifizieren. Es können mehrere zu untersuchende Dateien an Cuckoo übergeben werden, jedoch wird Cuckoo diese Dateien nacheinander und nicht parallel abarbeiten und auswerten. Somit ist

diese Sandbox nicht skalierbar. Bei Cuckoo sind bestimmte Aufgaben durch den Einsatz von REST APIs automatisierbar. Zu diesen Aufgaben zählen beispielsweise das Hinzufügen von Dateien in die Analysewarteschlange oder das Abfragen spezifischer Informationen, die den Analysevorgang betreffen. Es gibt noch dutzende weitere Aufgaben, welche mithilfe von REST APIs automatisierbar sind. [71]

6.2 Any.Run

Die Berichte, welche Any.Run als PDF oder Hypertext Markup Language (HTML) ausgibt, können sehr lang sein. Die Länge der Berichte ist abhängig von der Anzahl der aufgerufenen Prozesse oder den aufgebauten Internet Verbindungen. Dabei können mehr als 60 Seiten erstellt werden, was eine Bearbeitung für den Analysten wesentlich aufwendiger macht. Dafür sind diese Berichte im Vergleich zu Cuckoo jedoch viel detaillierter und übersichtlicher. Ebenfalls können Berichte von Any.Run als JSON-Dokument exportiert werden, was bedeutet, dass diese Berichte einfacher weiterverarbeitet werden können. Das liegt daran, dass dieses Format von den meisten Betriebssystemen, Programmiersprachen und Browsern unterstützt wird, wie bereits in Kapitel 6 erwähnt. Bei Hybrid Analysis hingegen können Berichte nicht in diesem Format heruntergeladen werden. Der online Bericht und der Bericht in Form eines Dokumentes beinhalten die selben Informationen. Jedoch sind diese Informationen online übersichtlicher und leichter verständlich dargestellt. Es können detailliertere Informationen über bestimmte HTTP-Verbindungen und Prozesse in Erfahrung gebracht werden, sobald man den spezifischen Prozess oder die Verbindung anklickt. Hinzu kommt, dass diese Ansicht eine Zusammenfassung aller gesammelten IoCs bietet, um dem Analysten eine schnelle Übersicht zu präsentieren. [52] [56] [53] [39]

Weiterhin war Any.Run bei einem Großteil bekannter Malware Sampels in der Lage, den Namen zu identifizieren und zu diesem eine kurze Definition in den Bericht zu ergänzen. Zu diesen Informationen zählen beispielsweise die Art der Malware und deren Ziel. Auch ein Prozess Graph, Prozess Informationen und Netzwerk Aktivitäten sind in dem Bericht enthalten. Ein Prozess Graph veranschaulicht alle von der Malware erzeugten Prozesse. Informationen über die Netzwerk Aktivität können auch bei dieser Sandbox mithilfe einer echten Internetverbindung gesammelt werden, es kann jedoch auch eine simulierte Internetverbindung genutzt werden. Mithilfe der simulierten Internetverbindung können weitere Informationen gesammelt werden. Diese Funktion ist wie bereits in Kapitel 5.1 erwähnt, bei dieser Sandbox unter dem Namen Fake Net bekannt. [53] [39] [63]

Wie es bereits bei Cuckoo der Fall war, sammelt auch Any.Run weitere nützliche Informationen mithilfe der statischen Analyse. Dabei ist der Umfang der in den Berichten enthaltenen Informationen kleiner als bei Cuckoo. Oftmals war Any.Run beispielsweise nicht in der Lage die von der Malware genutzten Strings zu identifizieren, wozu Cuckoo jedoch in der Lage war. Bei diesen Strings handelt es sich um Zeichenketten, welche von dem Programm verwendet werden, wenn dieses beispielsweise eine Datei erstellt. Die Malware speichert den Dateinamen als String. Diese Informationen können Hinweise auf die Funktionalität einer Malware liefern. Weiterhin gibt Any.Run nicht alle Anti-Viren Programm an, die diese Malware untersucht haben. Es wird lediglich angegeben, dass die Malware von einem Anti-Viren

Programm erkannt wurde. Cuckoo hingegen gibt eine Liste aller getesteten Anti-Viren Programm, samt Ergebnis aus, wobei keine weiteren Informationen erworben werden. Ebenso wie in Cuckoo, sind Informationen über die ausführbare Datei und Imports in den Any.Run Berichten enthalten. [11, S. 34] [27, S. 67]

Wie bereits in der Vorstellung der Sandbox Systeme erwähnt, nutzt diese Sandbox MITRE ATT&CK, um zusätzliche Informationen, in Hinsicht auf den Ablauf des Angriffes zu sammeln. Diese Informationen können jedoch nur auf der Website eingesehen werden, nicht jedoch in den abgespeicherten Berichten. Diese Informationen bieten einen tieferen Einblick in den Ablauf des Angriffs worauf bereits in Kapitel 5.1 eingegangen wurde. Weiterhin ist anzumerken, dass in dem Bericht neben den sehr komplexen Informationen über Prozesse, Registrys und den Indikatoren für die bösartiges und verdächtiges Verhalten ebenfalls eine Übersicht des aufgetreten Verhaltens während der Analyse vorhanden ist. Dabei wurde beispielsweise das Verhalten und dessen Auswirkung von Dateien, Prozessen und Diensten während der Analyse beobachtet und ausgewertet. Während der Analyse kam es beispielsweise zu einem CPU Überlauf. Die Malware hat andere Dateien heruntergeladen oder sie hat eine Verbindung zum Internet aufgebaut. [53]

Bei Any.Run handelt es sich um keine Open Source basierte Sandbox, sondern um eine online bereitgestellte Sandbox. Damit können keine benutzerdefinierten Anpassungen für gewünschte Funktionen an der Sandbox vorgenommen werden. Auch Any.Run verfügt über eine Vielzahl an Methoden, um Evasion Techniken zu umgehen. Diese Sandbox ist in der Lage einen Reboot auszuführen, falls es bei der Analyse einer Malware nötig werden sollte. Bestimmte Malware erfordert den Neustart einer Maschine um sicherzugehen, dass es sich nicht um ein Analyse System handelt. Weiterhin verfügt Any.Run über die Funktion der Live Interaktion. Das bedeutet, dass der Analyst während der Malware Analyse beispielsweise Programme, den Internet Browser oder Dokumente öffnen kann. Damit wird ein Benutzerverhalten simuliert, auf das einige Malware Samples achten. Die menschliche Interaktionen wurden bereits in Kapitel 4.3 näher erläutert. Simuliertes Mausverhalten ist unter Any.Run nicht verfügbar. Diese Funktion wäre jedoch überflüssig, da echtes Nutzerverhalten ausgeführt wird und die Evasion Technik somit bereits umgangen werden kann. Es wurde verglichen, ob bei dieser Sandbox vorinstallierte Software vorhanden ist. Any.Run biete mehrere Optionen an, in denen in der Gast Maschine verschiedene Software in unterschiedlichem Umfang installiert ist. Darunter zählen beispielsweise der Internet Explorer, Mozialla Firefox, Microsoft Office Excel und viele weitere Programme. [39] [6, S. 5] [9]

Bei der Betriebssysteme Auswahl schneidet die Sandbox von Any.Run am schlechtesten ab. Diese bietet zwar eine Vielzahl an Windows Systemen, darunter sogar Windows 10 und Windows 11, jedoch kein Betriebssystem von Linux, MacOS oder Android. Größtenteils ist Windows als Betriebssystem für die Analyse ausreichend, da die meisten Malware Samples für dieses entwickelt wurde. In manchen Fällen gibt es jedoch auch Malware, welche auf anderen Systemen untersucht werden muss und in diesem Fall muss eine andere Sandbox herangezogen werden. Dahingegen bieten die anderen beiden Sandbox Systeme eine ausgeglichenere Auswahl verschiedener Betriebssystemen an, nicht nur Windows als einzige Plattform. [39] [58]

Any.Run konnten ebenso wie Cuckoo die getesteten URLs problemlos analysieren und die Schädlichkeit anschließend korrekt klassifizieren. Bei der Implementierung und Bereitstellung der Sandbox entstand keinerlei Aufwand für den Benutzer, da es sich bei dieser um eine online Sandboxen handelt, welche einfach über das Internet aufgerufen und verwendet werden kann. Diese Sandbox ist in der Lage, die Analyse verschiedener Malware Samples parallel ablaufen zu lassen, was ein Vorteil gegenüber Cuckoo darstellt. Dafür wird für jede Malware eine einzelne Maschine gestartet und die Ergebnisse können nach der Beendigung der Beobachtung abgelesen werden. Damit ist diese Sandbox skalierbar und die Zeit zwischen den Analyse kann effektiver genutzt werden, indem bereits weitere Malware Samples ausgewertet werden. Wie es bereits bei Cuckoo der Fall war, können auch bei dieser Sandbox bestimmte Aufgaben automatisiert werden, indem REST APIs eingesetzt werden. [33, S. 2] [52]

6.3 Hybrid Analysis

Ein Nachteil gegenüber den andern beiden Sandboxen, ist das bei der Hybrid Analysis Sandbox (HAS) keine Berichte im JSON-Format heruntergeladen werden können. Das bedeutet, dass die darin enthalten Informationen nicht direkt weiterverarbeitet werden können, wie es bei den anderen Sandbox Systeme möglich war. Der Grund für diesen Nachteil ist die Verwendung der kostenlosen Version, jedoch kann der Bericht online aufgerufen werden. Ein weiterer bereits erwähnter Nachteil, ist das der Bericht auf der Website nicht nur für den Analysten, sondern für jeden Nutzer von Hybrid Analysis zugänglich ist. Auch das liegt an der hier verwendeten Version. [56]

Die auf der Website verfügbaren Informationen sind nach einer kurzen Eingewöhnungszeit sehr übersichtlich dargestellt. Die Menge sowie die Qualität der dort aufgeführten Informationen ist nicht minderwertig im Vergleich zu den beiden Konkurrenten. Im Gegenteil, es sind mehr Informationen vorhanden, als bei Cuckoo und eine ähnliche Menge an Daten wie bei Any.Run. Jedoch sind die Daten nicht so gut strukturiert und verständlich dargestellt wie bei der Any.Run Sandbox. Hybrid Analysis ist in der Lage der untersuchten Malware den zugehörigen Namen zu versehen, leider werden keine weiteren Informationen über die mit diesem Namen zusammenhängende Malware ausgegeben. Es werden am Anfang kurze Informationen dargestellt, um bestimmte Eigenschaften der Malware zu nennen, bevor die technischen Details der Malware ausgewertet werden. Zu diesen Informationen gehören beispielsweise ausgeführte Evasion Techniken oder eingesetzte Methoden, um eine persistente Verbindung herzustellen. Der Inhalt des Berichtes dieser Sandbox ist sehr ähnlich zu den Informationen, welche in den Berichten der anderen beiden Sandbox vorhanden sind. Weiterhin liefert die Hybrid Analysis Sandbox und auch die Sandbox von Any.Run bei diversen Malware Samples mehr Informationen, über die Bedrohung, als es die Cuckoo Sandbox tut. Diese vermehrten Informationen sind vor allem bei dem aufgezeichneten Netzwerkverkehr auffällig. Bei Cuckoo werden weniger Anfragen ausgehend von der Malware abgebildet, als bei HAS. Der Bericht der HAS enthält zum Beispiel Informationen über ausgeführte Prozesse, Hash Werte die in Beziehung zu der Malware stehen und etablierte Netzwerkverbindungen. [35]

Darüber hinaus setzt auch diese Sandbox auf den Einsatz von statischer Analyse, um den Informationsumfang zu erweitern. Dabei kommen aus den anderen Sandbox Systeme bereits bekannten Methoden der String Extraktion, Anti-Viren Programm Überprüfung und Imports zum Einsatz. Für die Sammlung relevanter Netzwerkdaten kann auch bei dieser Sandbox, sowohl eine echte Internetverbindung, als auch eine simulierte Verbindung genutzt werden. [72]

Die echte Verbindung kann genutzt werden, um den tatsächlichen Datenaustausch zwischen dem C2C Server auszuwerten. Die Option der simulierten Verbindung liefert Informationen über alle Ziele, mit denen eine Kommunikation während der Analyse statt fand. Zuletzt nutzt Hybrid Analysis MITRE ATT&CK, um Daten bezüglich der von der Malware verwendeten Angriffstaktiken und genutzte Techniken zu erhalten. Diese Daten helfen ein besseres Verständnis über den Ablauf zu erlangen, wie bereits in Kapitel 5.1 erläutert. [63] [72] [10, S. 46, 233–234]

Diese Sandbox ist nicht Open Source basiert, womit sie Cuckoo gegenüber einige damit einhergehende Vorteile verliert. Auch bei den hier verglichenen Anti-Evasion Techniken liegt Hybrid Analysis hinter seinen Konkurrenten. Diese Sandbox nutzt die Methoden der automatisierten Mausbewegung, um einen Benutzer nachzuahmen. Aus den Screenshots, welchen in den Berichten enthalten sind, geht hervor, dass vorinstallierte Software zum Einsatz kommt. Diese wird genutzt, um ein Benutzersystem so gut wie möglich zu imitieren. Bei Analyse Systemen wird keine dieser Software benötigt, was bedeutet, dass eine Malware ein System deshalb als Sandbox enttarnen könnte. Damit können einige der bekanntesten Malware Evasion Techniken umgangen werden. Hybrid Analysis verzichtet auf das Feature der Live Interaktion, welches ein großer Vorteil im Kampf gegen die Malware ist. Das liegt daran, dass diese Funktion genutzt wird, um die Evasion Technik, welche menschliches Verhalten überprüft, zu umgehen. Weiterhin gibt es keine Neustart Option, welche wichtig sein kann, sollte die untersuchte Malware einen Neustart erfordern. Einige Malware Varianten sind in der Lage ihren Schadcode so lange zu verstecken, bis der Computer neu gestartet wird. Damit stehen von den vier verglichenen Techniken lediglich zwei zur Verfügung. [73] [31, S. 4] [52]

Ein Vorteil von Hybrid Analysis ist, dass verschiedene Betriebssysteme unterstützt werden. Darunter zählen Windows, Linux und sogar Android. Jedoch ist zu beachten, dass die Analyse eines Android Betriebssystem auf die Verwendung von statischer Analyse begrenzt ist. Es kann also keine dynamische Analyse und damit keine Verhaltensanalyse durchgeführt werden. Zu den verwendbaren Windows Betriebssystemen zählen Windows 7 und Windows 10. Hybrid Analysis ist in der Lage URLs zu analysieren, jedoch war die Durchführung bei Any.Run und Cuckoo einfacher als bei Hybrid Analysis. Außerdem war es dieser Sandbox nicht möglich jede URL zu analysieren. Der Grund dafür ist, dass die Sandbox in einigen Fällen nicht in der Lage war, die Domain zu erkennen. Es kam nur eine Fehlermeldung, dass die Domain nicht existiert und anschließend wurde der Vorgang abgebrochen. Die Konkurrenten hingegen waren in der Lage dieselben URLs problemlos zu analysieren und anschließend zu klassifizieren, wobei es sich also um einen Nachteil dieser Sandbox handelt. [45]

Wie es bereits bei Any.Run der Fall war, muss auch bei der Hybrid Analysis Sandbox keinerlei Aufwand betrieben werden, um die Sandbox nutzen zu können. Es muss lediglich die Website besucht werden und schon kann kann der Vorgang der Malwareanalyse starten. Hybrid

Analysis bietet die Möglichkeit mehrere Samples gleichzeitig an die Sandbox zu senden. Es können allerdings keinerlei Optionen für die Auswahl der Umgebung, in der die Malware ausgeführt wird, getroffen werden. Bei den getesteten Samples wurden unter Verwendung dieser Option kaum Malware richtig klassifiziert und es gab nahezu keine Informationen über diese Samples. Es gibt weiterhin die im Normalfall genutzte Möglichkeit einzelne Samples nacheinander an die Sandbox zu übergeben. Dabei kommt es jedoch dazu, dass die Samples in eine Warteschlange gesteckt werden und dann nacheinander abgearbeitet werden. Das bedeutet für diese Sandbox, dass sie nicht skalierbar ist. Auch Hybrid Analysis bieten die Nutzung von REST APIs an, um damit Aufgaben wie die Übermittlung, die Abfrage von Informationen oder das anfragen von Berichten zu automatisieren. [74]

7 Zusammenfassung und Ausblick

Der Vergleich der Sandbox Systeme erfolgte unter anderem, indem verschiedene Malware Samples ausgewertet wurden. Dabei handelte es sich sowohl um aktuelle Varianten, als auch um ältere. Somit wurden einige Vor- und Nachteile der Sandbox Systeme erkennbar. Aus den Ergebnissen dieser Auswertungen waren weitere Fakten ersichtlich.

Es wurde bereits eine Vielzahl verschiedener Malware Arten benannt und erläutert. Außerdem wurde kurz erwähnt, dass es oftmals zu einer Kombination von verschiedenen Arten kommt. Diese Kombination war während der Analyse von Malware zu beobachten. Dabei kam es beispielsweise zu Kombinationen von Torjanern mit Downloadern. Weiterhin war auffällig, dass es sich bei der Mischung der Malware Samples, um einen großen Anteil an Trojanern handelte. Zuletzt wurde in Kapitel 3.2, neben der dynamischen und statischen Analyse, ebenfalls die hybride Analyse vorgestellt. Bei diesem Ansatz wird die dynamische Analyse durchgeführt und weiterhin werden Informationen mithilfe der statischen Analyse erlangt. Bei jeder der drei getesteten Sandboxen kam die statische Analyse neben der dynamischen Analyse zu Einsatz. Das bedeutet, dass alle der drei Sandbox Systeme den Ansatz der hybriden Analyse verfolgen. [5, S. 11] [28, S. 1667]

Letztendlich lag das Ziel dieser Arbeit darin, eine Entscheidungshilfe für die Auswahl einer Sandbox zu liefern. Dabei stellt sich die Frage: Welches der Sandbox Systeme die aktuell auf dem Markt vertreten sind, bietet die größten Vorteile? Hier muss gesagt werden, dass keine eindeutige Antwort auf diese Frage geliefert werden kann, da nicht alle Faktoren in die Entscheidung fließen können. Denn jedes Unternehmen besitzt andere Ansprüche und finanzielle Mittel. Im Folgenden werden auf Grundlage der Metriken und eigenen Annahmen, einige Empfehlungen für unterschiedliche Ansprüche ausgesprochen. Jedoch ist zu erwähnen, dass einzelne Fakten, wie die erwähnten finanziellen Mittel oder Anzahl der Mitarbeiter, zu einer anderen Sandboxauswahl leiten könnten. Jede der vorgestellten Sandbox Systeme war in der Lage bekannte Malware zu identifizieren und klassifizieren. Allerdings bot jede bestimmte Vorteile, dafür ebenfalls einige Nachteile, welche für bestimmte Anwendungsbereiche nicht annehmbar sind.

Jede der Sandboxen hat unterschiedlich viele Informationen, in unterschiedlicher Darstellung über die Bedrohung geliefert. Das liegt unter anderem daran, dass die von den Sandbox Systemen verwendeten Anti-Evasion Techniken auch gegen neuere Malware funktioniert haben. Diese neuere Malware verwendet ausgereiftere Techniken, als es ältere Malware Samples tun. Jede einzelne getestete Malware nutzte Evasion Techniken, jedoch variiert der Umfang dieser von Malware zu Malware. Diverse ausweichende Malware nutzte eine Vielzahl dieser Methoden, um den Schadcode vor der Sandbox zu verbergen. Andere Malware Exemplare nutzten weniger komplexe und in geringer Anzahl auftretende Evasion Techniken. Neben der Malware bot jede Sandbox einen unterschiedlichen Umfang an Anti-Evasion Techniken an, um diese Malware trotzdem erfolgreich analysieren zu können. [6, S. 5]

Die benötigten Funktionen einer Sandbox variieren, je nach individuellen Umständen und Wünschen. Dabei ist es ebenfalls individuell, wie viel Aufwand investiert werden soll. In diesem Abschnitt soll auf drei verschiedene Umstände eingegangen werden, um zu verdeutlichen, welche Sandbox für welchen Einsatz am besten geeignet ist. Zu diesen Umständen gehört zum einen die Privatperson, welche sich in der Freizeit mehr mit der dynamische Malware Analyse auseinandersetzen möchte. Jedoch verfügen Privatpersonen oftmals nicht über die finanziellen Mittel, wie Unternehmen dies tun. Weiterhin wird eine Empfehlung für kleinere Unternehmen ausgesprochen. Diese müssen vielleicht bekannte und unbekannte Malware erkennen, um die Firma vor finanziellem Ruin oder einem Imageschaden zu schützen und dabei muss der Aufwand gegen das zu zahlende Geld abgewogen werden. Zuletzt gibt es noch das Großunternehmen, welches jegliche Bedrohung erkennen muss und darüber hinaus über die nötigen finanzielle Mittel verfügt. Die Sandbox Auswahl für die jeweiligen Umstände sind, wie schon erwähnt, in dieser Arbeit subjektiv. [28, S. 1665] [15, S. 46–47]

Zuerst wird auf die Privatperson eingegangen, welche sich mehr mit der dynamischen Malware Analyse beschäftigen möchte. Hier entsteht kein Schaden, sobald eine Malware nicht richtig klassifiziert wurde, da kein Produktionssystem in Gefahr ist. Weiterhin ist es wahrscheinlich, dass eine solche Person nicht bereit ist, Geld dafür zu investieren oder wenn nur kleinere Summen, als es ein Unternehmen könnte. Damit wird die Auswahl bereits auf zwei Sandbox Systeme eingegrenzt, Cuckoo und Hybrid Analysis, da sich diese beiden Sandboxen kostenlos nutzen lassen. Hybrid Analysis liefert etwas mehr Informationen als es die Cuckoo Sandbox macht und beinhaltet sogar Informationen von MITRE ATT&CK über die verwendete Taktiken. Bei den Betriebssystemen ist Hybrid Analysis in der Lage neuere Windows Systeme, wie Windows 10 zu nutzten, jedoch kann keine dynamische Android Analyse durchgeführt werden, wozu Cuckoo hingegen in der Lage ist. Weiterhin ist abzuwägen, ob die Zeit investiert werden soll, die Sandbox eigenhändig aufzusetzen. Dafür werden bestimmte Kenntnisse benötigt, sollten diese nicht vorhanden sein, sollte abgewägt werden, ob Cuckoo in diesem Falle nicht doch die falsche Wahl ist. Sollte jemand jedoch die nötigen Kenntnisse besitzen und der Wille existiert die benötigte Zeit zu investieren, verfügt Cuckoo über einige erheblichen Vorteil. Dazu zählt, dass diese Sandbox Open Source basiert ist, somit kann der Nutzer anschließend Erweiterungen implementieren und fehlende Features hinzufügen. Sollten URL Analysen eine große Rolle für die Entscheidung spielen, ist Cuckoo zu bevorzugen, da Hybrid Analysis in der URL Analyse wesentlich schlechtere Ergebnisse abgeliefert hat. Hybrid Analysis war in einigen Fällen nicht in der Lage die URLs zu analysieren. Meist spielt die Skalierbarkeit keine Rolle, da eine Privatperson nur eine begrenzte Anzahl an Malware analysiert und nicht mehrere Tausend Samples pro Woche, wie es bei Firmen der Fall sein kann. Es sollte zuletzt beachtet werden, dass die analysierten Malware Samples von Hybrid Analysis veröffentlicht werden. Dadurch könnten private Daten veröffentlicht werden, dies sollte jedoch ein nicht so wichtigen Fakt darstellen, da meist bereits bekannte Malware untersucht wird. [10, S. 233–234] [35] [42, S. 20] [27, S. 116–124] [8]

Bei einem kleineren Unternehmen können andere Entscheidungen getroffen werden. Cuckoo muss implementiert und gewartet werden, was möglicherweise zeitlich für einige Firmen nicht realisierbar ist. Weiterhin müssen Experten vorhanden sein, um die Sandbox erweitern zu können. Auf der anderen Seite ist Hybrid Analysis eine kostenlose Variante einer online Sandbox, das bedeutet, dass die analysierte Malware für alle Nutzer zugänglich gemacht wird. Sollten dabei firmeneigene Informationen enthalten sein, ist zu vermeiden, dass diese veröf-

fentlicht werden. Es kann sich also zwischen Cuckoo, welche einen relativ großen Aufwand erfordert, und Any.Run entschieden werden, welche bezahlt werden muss. Any.Run bietet mehr Informationen, welche ebenfalls wesentlich übersichtlicher dargestellt sind, als es bei der Cuckoo Sandbox der Fall ist. Die Informationsdarstellung kann für ein Unternehmen eine große Relevanz darstellen, um beispielsweise bestehende Bedrohungen eines Kunden oder eines eignen Systems schneller beseitigen zu können. Es sollte weiterhin beachtet werden, wie viele Malware Analysen über einen bestimmten Zeitraum ausgeführt werden. Sollte es sich um eine große Anzahl handel, spielt Skalierbarkeit für die Entscheidung eine große Rolle. Diese bringt nur Any.Run mit sich, nicht jedoch Cuckoo. [42, S. 20] [45] [13, S. 76] [8]

Bei großen Unternehmen kann der Fokus auf ganz andere Faktoren gerichtet werden, jedoch kann nicht generalisiert werden welche Faktoren von großer Bedeutung sind. Jedes Unternehmen setzt ihren Fokus auf andere Eigenschaften. Ein größeres Budget ist bei einem Großenunternehmen mit einer hohen Wahrscheinlichkeit vorhanden. Sobald der Ablauf von Prozessen bekannt ist, kann damit begonnen werden die einzelnen Prozesse zu automatisieren. Die Möglichkeit der Automatisierung bot in diesem Vergleich jede Sandbox, weshalb anhand diesen Punktes kein Ausschluss erfolgen kann. Wie es bereits bei einem kleinerem Unternehmen der Fall war, hat nicht jede Firma die Zeit und Ressourcen eine Sandbox selbst zu implementieren auch wenn entsprechende Experten dafür vielleicht vorhanden sind. Somit sollte beachtet werde, ob eine Cuckoo Sandbox wegen des Zeitaufwands überhaupt eine Option ist. Aus Kapitel 5.0.3 ist bereits bekannt, dass analysierte Samples öffentlich gemacht werden, weshalb es von Firmen zu einer Vermeidung von kostenlosen Sandbox Systemen wie Hybrid Analysis kommt. Weiterhin ist es wichtig einen Überblick über die zu analysierenden Samples pro Tag zu haben. Sollte es sich um eine erhebliche Menge handeln, sollte eine skalierbare Sandbox gewählt werden. Die Skalierbarkeit ermöglicht es dem Nutzer, verschiedene Malware Samples parallel analysieren zu können, um eine gewisse Zeitersparnis zu erzielen. Diese Möglichkeit bietet lediglich Any.Run an. Zuletzt sind die verwendbaren Betriebssysteme zu erwähnen. Any.Run bietet neben der Menge und Darstellung der Informationen, unter anderem mithilfe von MITRE ATT&CK, viele weitere Vorteile. Jedoch sollte festgehalten werden, dass diese Sandbox nur Betriebssysteme von Windows für die Analyse benutzten kann. Sollte eine Malware auf Linux oder Android Systemen festgestellt werden, kann diese nicht analysiert werden. Cuckoo hingegen kann auf diesen Plattformen Malware analysieren, was einen weiteren Vorteil gegenüber Any.Run darstellt. Für den Einsatz von URL Analysen, kann sowohl Cuckoo, als auch Any.Run genutzt werden, da diese während der Tests in der Lage waren die bösartigen URLs problemlos zu analysieren. [75] [42, S. 20] [13, S. 76] [8] [39]

7.1 Ausblick

Trotz der bisherigen Möglichkeiten, die die dynamische Malware Analyse aufweist, wird diese Methode immer weiter verbessert. Diese Entwicklung ist zwingend erforderlich, um mit der immer weiter anwachsenden Bedrohung der Malware mithalten zu können. Abgesehen von den kommenden Entwicklungen gibt es bereits einige Aspekte, die nicht in dieser Arbeit realisiert wurden, welche jedoch einen Einfluss auf die Ergebnisse nehmen können. Das bedeutet, dass neue Informationen gewonnen werden könnten, was wiederum die Klassifizierung von Malware, sowie anschließend getroffene Gegenmaßnahmen beeinflussen kann. [6, S. 5] [13, S. 14]

Hierbei bietet Cuckoo die größten Verbesserungsmöglichkeiten an. Wie bereits in Kapitel 5.0.1 erwähnt, wurde für diese Arbeit Cuckoo 2, anstatt der neusten Version Cuckoo 3, eingesetzt. Cuckoo auf Basis von Python 3 ist bereits verfügbar, jedoch gibt es keine offizielle Dokumentation für die Installation von Cuckoo 3. Die bereits vorhanden Anleitungen werden lediglich von Drittanbietern angeboten. Wie bereits erwähnt, handelt es sich bei Python 2 um eine veraltete Version. Diese wird seit 2020 nicht weiter entwickelt und viele Python Bibliotheken unterstützen diese Version nicht mehr. Das bedeutet, dass eine Verwendung von Cuckoo auf Basis von Python 3 einen positiven Einfluss auf die Ergebnisse haben könnte. [51] [50] [47]

Es ist zu erwähnen, dass Cuckoo für diese Arbeit in seinem Basiszustand genutzt wurde. Das bedeutet, dass keinerlei Anpassungen an der Sandbox vorgenommen wurde, um Funktionen hinzuzufügen, was jedoch eine der größten Vorteile dieser Sandbox ist. Der Einsatz einer verbesserten Version könnte dazu führen das Cuckoo bessere Ergebnisse erzielt. Dafür könnten verschiedenste Verbesserungen sorgen, wie zum Beispiel die Verbesserung der Anti-Evasion Techniken. Eine der bereits in Kapitel 4.3 erwähnten Evasion Techniken ist die Erkennung der Ausführungsumgebungs. Die Erkennung der Umgebung durch die Malware kann unterbunden oder zumindest erschwert werden, indem einige Änderungen an Cuckoo vorgenommen werden. [27, S. 116–124] [27, S. 7]

Abgesehen von dem Hinzufügen neuer Features durch Anpassung des Quellcodes, können ebenfalls die Konfigurationsdateien von Cuckoo weiter angepasst werden, um Verbesserungen hervorzurufen. Es gibt eine Vielzahl von Konfigurationsdateien und somit eine große Anzahl an Optionen, welche aktiviert und deaktiviert werden können. Eine weitere Anpassung der Optionen kann ebenfalls dazu führen, dass Cuckoo bessere Ergebnisse liefert. [27, S. 25]

Weiterhin sollten mehr Sandbox Systeme für einen noch aussagekräftigeren Vergleich hinzugezogen werden. Dadurch können die Unterschiede zwischen kommerziellen Sandboxen noch besser hervorgehen. Bisher kam es zu einer Verdeutlichung der bei einer kostenpflichtigen Sandbox vorhanden Features durch den Vergleich mit kostenlosen Sandbox Systemen. Es konnte jedoch kein Vergleich zwischen Sandbox Systemen, welche nicht kostenlos sind, aufgestellt werden. Das liegt daran, dass nur von einer Vergleichssandbox die kostenpflichtig Version ausgewählt wurde. Es könnte beispielsweise die Vollversion von Joe Sandbox hinzugefügt werden, um zu vergleichen, wie eine weitere kommerzielle Sandbox gegen Any.Run abschneidet. Darüber hinaus können neben Joe Sandbox viele weitere hinzugefügt werden, um letztendlich die beste für jeden Nutzen herauszufiltern. Weiterhin könnte ein Vergleich zu Cuckoo aufgestellt werden, indem beispielsweise die CAPE Sandbox hinzufügt wird. Diese ist ebenfalls, wie es bei Cuckoo der Fall ist eine Sandbox, welche kostenlos verfügbar ist, jedoch selbst implementiert werden muss. Anschließend könnten die Funktion dieser beiden verglichen werden, um zu sehen, welche Sandbox in dieser Klasse besser abschneidet oder ob die CAPE Sandbox besser gegen andere Konkurrenten abschneidet als es Cuckoo tut. [9] [76]

In der Vergleichsmatrix sind nur ausgewählte Metriken aufgeführt, welche potenziell nicht alle Aspekte abdecken, die sich von Firmen oder Kunden gewünscht werden. Ansprüche an ein Sandbox System sind von Person zu Person unterschiedlich. Es wurde darauf verwiesen,

dass immer mehr und neue Evasion Techniken von Malware benutzt werden. Das führt dazu, dass die in der Matrix aufgeführten Techniken in naher Zukunft nicht mehr aktuell sein könnten, sodass die Metriken zu einem späterem Zeitpunkt keine Aussagekraft mehr oder nur einen begrenzten Nutzen für die Auswahl haben. Somit ist es sinnvoll, diese Matrix auf dem neusten Stand zu halten und bei Bedarf neue Metriken für den Vergleich hinzuzufügen. [6, S. 5]

Anhang A: Bereitstellen einer Cuckoo Sandbox auf Ubuntu 20.04

Diese Implementation von Cuckoo basiert auf der Basis eines Ubuntu Version 20.04 Betriebssystems.

Installieren von essentiellen Paketen, Programmen und Konfiguration von tcpdump:

```
$ sudo apt get install  y libjpeg dev zlib1g dev swig curl libjpeg dev zlib1g dev swig mongodb
    postgresql libpq dev ssdeep tcpdump libcap2 bin net tools python dev libffi dev libssl dev
    libfuzzy dev libtool flex autoconf libjansson dev git virtualbox iptables persistent apparmor
    utils
$ curl https://bootstrap.pypa.io/pip/2.7/get pip.py  o get pip.py
$ sudo python get pip.py
$ sudo  H pip install distorm3==3.4.4 yara python==3.6.3 pydeep openpyxl ujson jupyter setuptools
    cuckoo
$ git clone https://github.com/volatilityfoundation/volatility.git
$ sudo volatility/python setup.py build
$ sudo volatility/python setup.py install
$ sudo setcap cap\_net\_raw,cap\_net\_admin=eip /usr/sbin/tcpdump
$ getcap /usr/sbin/tcpdump
$ sudo aa disable /usr/sbin/tcpdump
```

[77]

Installation und Konfiguration von VirtualBox, sowie die persistente Bereitstellung dieser vorgenommenen Konfigurationen:

```
$ vboxmanage hostonlyif create
$ vboxmanage hostonlyif ipconfig vboxnet0   ip 192.168.56.1
$ sudo nano /opt/systemd/vboxhostonly
!/bin/bash
hostonlyif create
vboxmanage hostonlyif ipconfig vboxnet0   ip 192.168.56.1
$ sudo chmod a+x /opt/systemd/vboxhostonly
$ sudo touch /etc/systemd/system/vboxhostonlynic.service
$ sudo nano /etc/systemd/system/vboxhostonlynic.service
Description=Setup VirtualBox Hostonly Adapter \newline
After=vboxdrv.service
[Service]
Type=oneshot
ExecStart=/opt/systemd/vboxhostonly
[Install]
WantedBy=multi user.target
$ systemctl daemon reload
$ systemctl enable vboxhostonlynic.service
```

[77]

Nachdem die Konfiguration und Installation erfolgreich abgeschlossen wurde, muss die .iso Datei von Windows 7 auf die Ubuntu Instanz übertragen und in die VM integriert werden. Dafür wird eine Windows 7 Maschine mit dem Namen “cuckoo1“ in VirtualBox erstellt. Diese Maschine verfügt über 8 GB RAM, 8 CPU Kerne und 80 GB Speicherplatz. [78] [77]

Nun muss die Einstellung “Nested VT-x/AMD-V“ für die VM aktiviert werden.

```
$ VBoxManage modifyvm cuckoo1   nested hw virt on
```

[78]

Alle nötigen Einstellungen wurden getroffen und die VM kann anschließend ausgeführt und das Windows 7 Betriebssystem installiert werden. Nach der Installation, müssen einige Windows Features deaktiviert werden, welche eine Malware Analyse verhindern würden. Diese Einstellungen sind unter “Gruppenrichtlinie bearbeiten“ zu finden. [77]

Die erste Einstellung “Benutzerkontensteuerung: Verhalten der Eingabeaufforderung für erhöhte Rechte für Administratoren im Administratorengenehmigungsmodus“ muss unter “Computerkonfiguration“ > “Windows-Einstellungen“ > “Sicherheitseinstellungen“ > “Lokale Richtlinien“ > “Sicherheitsoptionen“ geändert werden. Die auszuwählende Option lautete “Erhöhte Rechte ohne Eingabeaufforderung“. Weiterhin müssen in diesem Pfad die Optionen “Benutzerkontensteuerung: Anwendungsinstallationen erkennen und erhöhte Rechte anfordern“ und “Benutzerkontensteuerung: Alle Administratoren im Administratorengenehmigungsmodus ausführen“ auf “Deaktiviert“ gestellt werden. [78] [77]

Die nächste Einstellung befindet sich unter “Computerkonfiguration“ > “Administrative Vorlagen“ > “Windows-Komponenten“ > “Windows Update“ und lautet “Automatische Updates konfigurieren“. Dabei muss das erste Feld auf “Aktiviert“ und das zweite auf “2 - Vor download und automatischer Installation benachrichtigen“ gesetzt werden. [78] [77]

Eine weitere Einstellung muss unter “Computerkonfiguration“ > “Administrative Vorlagen“ > “Netzwerk“ > “Netzwerkverbindungen“ > “Windows Defender Firewall“ > “Domänenprofil“ verändert werden. Diese ist unter dem Namen “Windows Defender Firewall: Alle Netzwerkverbindungen schützen“ zu finden. Bei dieser Einstellung muss die Option “Deaktiviert“ ausgewählt werden. [78] [77]

Die letzte Einstellung heißt “Deaktivieren von Windows Defender Antivirus“ und ist unter dem Pfad “Computerkonfiguration“ > “Administrative Vorlagen“ > “Windows-Komponenten“ > “Windows Defender Antivirus“ zu finden. Diese Einstellung muss auf “Aktiviert“ gestellt werden. [78] [77]

Nun sollte die Gast Erweiterung installiert werden, um den vollen Funktionsumfang zu erhalten. Dazu zählt beispielsweise der Datenaustausch zwischen der Host und Gast Maschine. Anschließend müssen einige Dateien heruntergeladen und auf die Windows VM verschoben werden. Dafür wird ein geteilter Ordner benutzt, welche in dem VirtualBox Manager für die VM erstellt werden kann. [78] [77]

Zu den herunterzuladenden Dateien gehört ’Python Version 2.7.8’ und ’Python Pillow Version 2.5.3’, welche in den geteilten Ordner verschoben werden müssen. Nachdem diese beiden Dateien in den Ordner verschoben wurden, müssen sie auf der Windows 7 Maschine ausgeführt werden. Anschließend kann weitere Software installiert werden, welche das System mehr wie ein Nutzersystem als wie ein Analyse System aussehen lässt. Bei der in diesem Fall installierten Software handelt es sich um Adobe Reader, Open Office und Java. [78] [77] [44]

Nun muss die 'agent.py' Datei von der Ubunut Host Maschine in den geteilten Ordner übertragen werden. Anschließend muss diese Datei aus dem geteilten Ordner in den Pfad "C:\Users\USERNAME\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup" verschoben werden. Damit wird sie bei dem Start der Maschine ebenfalls ausgeführt. [77]

Nach der Installation der gewünschten Software, sowie der Verschiebung der Python Datei in den vorgegebenen Pfad, muss die Windows VM neu gestartet werden. Bei dem Start der Maschine sollte eine schwarzes Fenster aufgehen. Dabei handelt es sich um die Agent Datei von Cuckoo. [78] [77]

Folgend muss die Netzwerkkonfiguration für die Windows Maschine vorgenommen werden. Dies ist essentiell, um die Malwareausbreitung auf die Windows Maschine zu beschränken und eine vollkommen isolierte Umgebung verwenden zu können. Dafür muss in 'VirtualBox' in den Einstellungen der Windows Maschine eine Änderung in der Kategorie Netzwerk vorgenommen werden. Dieses muss auf "Host-only Adapter" gestellt werden und der Name des Netzwerkes ist "vboxnet0". [78] [77]

```
$ iptables  A FORWARD  o ens6  i vboxnet0  s 192.168.56.0/24  m conntrack   ctstate NEW  j ACCEPT
$ iptables  A FORWARD  m conntrack   ctstate ESTABLISHED,RELATED  j ACCEPT
$ iptables  A POSTROUTING  t nat  j MASQUERADE
$ echo 1 | sudo tee  a /proc/sys/net/ipv4/ip\_forward
$ sudo sysctl  w net.ipv4.ip\_forward=1
$ sudo su
$ iptables save > /etc/iptables/rules.v4
```

[77]

Weiterhin müssen einige Adressen der Windows VM geändert werden. Dazu wird "Netzwerkverbindungen anzeigen" in Windows geöffnet. Sobald das Fenster geöffnet wurde, sollten die Eigenschaften der angezeigten LAN Verbindung mit einem rechten Maus Klick geöffnet werden. Unter "Internetprotokoll version 4" können die erwünschten Änderungen vorgenommen werden. [79] [77]

IP-Adresse: 192.168.56.101
Subnetzmaske: 255.255.255.0
Standardgateway: 192.168.56.1
Bevorzugter DNS Server: 8.8.8.8
[78]

Danach ist die Gast Maschine vollständig konfiguriert und es kann ein Snapshot mit dem Namen "Snapshot1" von dieser erstellt werden. [78]

Abschließend müssen einige Konfigurationsdateien von Cuckoo angepasst werden. Diese sind unter folgendem Pfad verfügbar: "~/.cuckoo/conf" [77]

cuckoo.conf
machinery = virtualbox
memory_dump = yes
ip = 192.168.56.1 [78] [77]

auxiliary.conf
sniffer enabled = yes

processing.conf
memory enabled = yes

reporting.conf
enabled = yes
Report.html enabled = yes
mongodb enabled = yes [78] [77]

memory.conf
basic guest_profile = Win7SP1x64

virtualbox.conf
virtualbox mode = gui
machines = cuckoo1
optional = label = cuckoo1 and platform = windows
ip = 192.168.56.101
snapshot = Snapshot1 [78]

Nun muss zuerst Windows heruntergefahren, danach Ubuntu neu gestartet und folgender Befehl eingegeben werden.

```
$ cuckoo community
```

[78] [77]

Jetzt sollte Cuckoo und ein Webserver gestartet werden, um eine Analyse durchzuführen. Dieser Webserver erlaubt es dem Benutzer Malware Samples über eine Website hochzuladen und die Ergebnisse dort anschließend auswerten zu können. Dafür muss lediglich ein Befehl auf jeweils einer Ubuntu Konsole ausgeführt werden:

```
$ cuckoo
$ cuckoo web runserver 0.0.0.0:8000
```

[78]

Sobald diese Kommandos ausgeführt wurden, kann Mozilla Firefox geöffnet werden. Dort gibt man die URL “https://127.0.0.1:8000“ ein und hat Zugriff auf die Benutzeroberfläche von Cuckoo und kann mit der Malware Analyse starten. [78] [77]

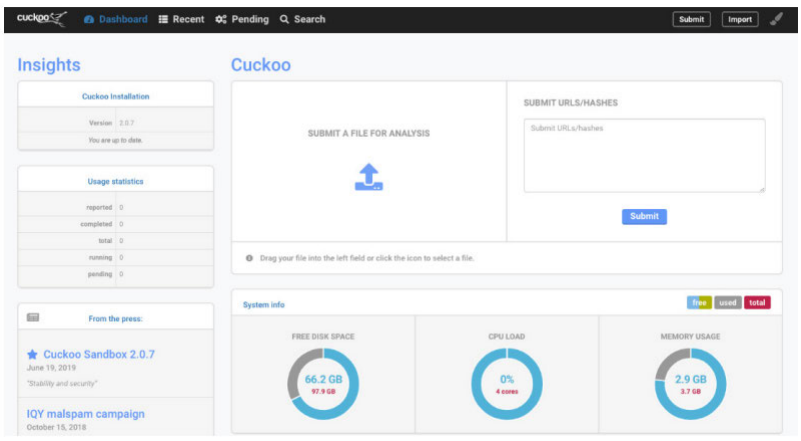

Abbildung A.1: Über Internet erreichbare Cuckoo Benutzeroberfläche [10, S. 100]

Literaturverzeichnis

[1] A.-J. Turner, *Find Quotes*, (Zugegriffen am: 26.01.2023). Adresse: https://www.goodreads.com/quotes/search?page=2&q=malware.

[2] A. Mohanta und A. Saldanha, *Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware*. 2020, ISBN: 978-1-4842-6192-7.

[3] Bundesamt für Sicherheit in der Informationstechnik (BSI), „Die Lage der IT-Sicherheit in Deutschland 2022“, Bundesamt für Sicherheit in der Informationstechnik (BSI), Techn. Ber., Okt. 2022.

[4] C. A. B. de Andrade, C. G. de Mello und J. C. Duarte, „Malware Automatic Analysis“, Mil. Eng. Inst. (IME), Techn. Ber., 2013.

[5] L. T. Gutierrez, „Malware Sandbox Deployment, Analysis and Development“, Magisterarb., UCLouvain, 2020.

[6] A. Jadhav, D. Vidyarthi und H. M., „Evolution of Evasive Malwares: A Survey“, Defence Institute of Advanced Technology, Techn. Ber., 2016.

[7] S. Bova, „An evaluation of anti-evasion techniques implemented in malware analysis sandboxes and debuggers“, Magisterarb., ING - Schule für Industrie- und Informationstechnik, 2020.

[8] R. Gupta, R. Guerra, L. Guiteau und B. Farrington, *Comparison of Various Sandboxes for Basic Dynamic Analysis*, Okt. 2020. Adresse: https://www.linkedin.com/pulse/comparison-various-sandboxes-basic-dynamic-analysis-rose-gupta/.

[9] C. Pernet, *ANY.RUN vs. Joe Sandbox: Malware analysis tools comparison*, (Zugegriffen am: 09.01.2023), März 2022. Adresse: https://www.techrepublic.com/article/anyrun-vs-joe-sandbox/.

[10] D. Barker, *Malware Analysis Techniques*. Packt, 2021, ISBN: 978-1-83921-227-7.

[11] Monnappa K A, *Learning Malware Analysis*. Packt, 2018, ISBN: 978-1-78839-250-1.

[12] Kaspersky, *Sandbox*, (Zugegriffen am: 10.01.2023). Adresse: https://www.kaspersky.com/enterprise-security/wiki-section/products/sandbox.

[13] M. Sikorski und A. Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. William Pollock, 2012, ISBN: 1-59327-290-1.

[14] Cisco, *What is Malware?*, (Zugegriffen am: 01.12.2022). Adresse: https://www.cisco.com/c/en/us/products/security/advanced-malware-protection/what-is-malware.html.

[15] E. Amberg und D. Schmid, *Hacking: Der umfassende Praxis-Guide*, 2. Aufl. mitp, 2022, ISBN: 978-3-7475-0482-6.

[16] R. Lionell, *The Motives Behind Cyber Crime*, (Zugegriffen am: 23.11.2022), Aug. 2015. Adresse: https://gramener.com/blog/posts?post=08-21-2015-cyber-crime-the-motives.md.

[17] P. Krempel, „Hackerkultur“, Bachelor’s Thesis, University of Cologne, 2014.

[18] X. Li, „A review of Motivations of Illegal Cyber Activities“, *Kriminologija Socijalna Integracija*, Feb. 2017.

[19] Bundesamt für Sicherheit in der Informationstechnik (BSI), *Social Engineering - der Mensch als Schwachstelle*, (Zugegriffen am: 12.01.2023). Adresse: https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Social-Engineering/social-engineering_node.html.

[20] M. Tremmel, *Der Bad-USB-Stick Rubber Ducky wird noch gefährlicher*, (Zugegriffen am: 03.01.2023), Aug. 2022. Adresse: https://www.golem.de/news/hacking-der-bad-usb-stick-rubber-ducky-wird-noch-gefaehrlicher-2208-167713.html.

[21] J. Regan, *Die bittere Wharheit über USB-Sticks*, (Zugegriffen am: 12.12.2022), Feb. 2020. Adresse: https://www.avg.com/de/signal/the-dirty-truth-about-usbs.

[22] M. Tischer u. a., „Users Really Do Plug in USB Drives They Find“, University of Illinois, Urbana Champaign; University of Michigan, Techn. Ber., 2016.

[23] Bundesamt für Sicherheit in der Informationstechnik (BSI), *Spam - zwielichtige E-Mails und Falschmeldungen*, (Zugegriffen am: 28.12.2022). Adresse: https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Spam-Phishing-Co/Spam/spam_node.html.

[24] G. Torres, *What Is a Computer Virus?*, (Zugegriffen am: 22.01.2023), Dez. 2017. Adresse: https://www.avg.com/en/signal/what-is-a-computer-virus.

[25] Bundesamt für Sicherheit in der Informationstechnik (BSI), *Adware und Spyware - wo liegen die Unterschiede?*, (Zugegriffen am: 12.01.2023). Adresse: https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Schadprogramme/Adware-und-Spyware/adware-und-spyware_node.html.

[26] A. Kleymenov und A. Thabet, *Mastering Malware Analysis: A malware analyst's practical guide to combating malicious software, APT, cybercrime, and IoT attacks*, 2. Aufl. Packt, 2022, ISBN: 978-1-80324-024-4.

[27] D. Oktavianto und I. Muhardianto, *Cuckoo Malware Analysis: Analyze malware using Cuckoo Sandbox*. Packt, 2013, ISBN: 978-1-78216-923-9.

[28] R. Sihwail, K. Omar und K. A. Z. Ariffin, „A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis“, *International Journal on Advanced Science Engineering and Information Technology*, Sep. 2018.

[29] C. Hummert, „Forensik in der digitalen Welt“, in D. Labudde und M. Spranger, Hrsg. Springer Spektrum, 2017, Kap. Malware Forensik, S. 199–214, ISBN: 978-3-662-53800-5.

[30] G. Cabau, M. Buhu und C. Oprisa, „Malware Classification Based on Dynamic Behavior“, Technical University of Cluj-Napoca, Techn. Ber., 2016.

[31] VMRAY, „Defeating Evasive Malware“, Techn. Ber., 2020.

[32] A. Loo, *What is a Hash Function?*, (Zugegriffen am: 11.01.2023), Nov. 2022. Adresse: https://corporatefinanceinstitute.com/resources/cryptocurrency/hash-function/.

[33] M. Vasilescu, L. Gheorghe und N. Tapus, „Practical malware analysis based on sandboxing“, University Politehnica of Bucharest, Techn. Ber., 2014.

[34] VMRAY, „Why Malware Sandboxing Matters“, Techn. Ber., 2021. Adresse: https://www.vmray.com/resource/why-malware-sandboxing-matters-whitepaper/.

[35] Hybrid Analysis, *Analysis Overview*, (Zugegriffen am: 28.01.2023). Adresse: https://www.hybrid-analysis.com/sample/a0f4d346a674b2703ce0640803a4e3c8c5da9d9329bc100fc6f33b51a4258da8.

[36] VMRAY, *VMRay Detection Analysis Technologies*, (Zugegriffen am: 13.12.2022). Adresse: https://www.vmray.com/detection-analysis-technologies/#machine-learning.

[37] M. Lechtik, „Inside Cuckoo Sandbox: A view into Cuckoo's internals and emulation flow“, *eForensics Magazine*, Jg. 8, Nr. 3, 2019.

[38] J. Sugeng, „Automated Malware Analysis with Cuckoo Sandbox“, *eForensics Magazine*, Jg. 8, Nr. 3, 2019.

[39] ANY.RUN, *Plans that fit your business strategy*, (Zugegriffen am: 29.01.2023). Adresse: https://app.any.run/plans/.

[40] E. Menahem, *Sandboxing is Limited. Here's Why and How to Best Stop Zero-Day Threats*, (Zugegriffen am: 13.11.2022), Apr. 2020. Adresse: https://www.catonetworks.com/blog/sandboxing-is-limited-heres-why-and-how-to-best-stop-zero-day-threats/.

[41] VMRAY, *Reaktion auf Vorfälle_de*, (Zugegriffen am: 13.01.2023). Adresse: https://www.vmray.com/de/losungen/reaktion-auf-vorfalle/.

[42] A. Mills und P. Legg, „Investigating Anti-Evasion Malware Triggers Using Automated Sandbox Reconfiguration Techniques“, *Journal of Cybersecurity and Privacy*, 2020.

[43] Cuckoo Foundation, *Requirements*, (Zugegriffen am: 23.01.2023). Adresse: https://docs.cuckoosandbox.org/en/latest/installation/guest/requirements/#additional-software.

[44] Cuckoo Foundation, *Sandboxing*, (Zugegriffen am: 23.01.2023). Adresse: https://cuckoo.readthedocs.io/en/latest/introduction/sandboxing/.

[45] Hybrid Analysis, *Hybrid Analysis*, (Zugegriffen am: 28.01.2023). Adresse: https://www.hybrid-analysis.com/.

[46] R. Hat, *What is open source?*, (Zugegriffen am: 03.01.2023), Okt. 2019. Adresse: https://www.redhat.com/en/topics/open-source/what-is-open-source.

[47] Cuckoo Foundation, *Cuckoo Sandbox 2.0.7*, (Zugegriffen am: 09.01.2023), Juni 2019. Adresse: https://cuckoosandbox.org/blog/207-interim-release.

[48] Cuckoo Foundation, *Installation*, (Zugegriffen am: 10.01.2023). Adresse: https://cuckoo.readthedocs.io/en/latest/installation/.

[49] Cuckoo, *Cuckoo Sandbox Book*, Cuckoo Foundation. Adresse: https://cuckoo.sh/docs/installation/host/requirements.html.

[50] L. Hande, *Python Libraries Every Programming Beginner Should Know*, (Zugegriffen am: 22.01.2023), Aug. 2022. Adresse: https://learnpython.com/blog/python-libraries-for-beginners/.

[51] S. Miller, *Python 2 vs. Python 3: Which Should You Learn?*, (Zugegriffen am: 23.01.2023), Juli 2021. Adresse: https://www.codecademy.com/resources/blog/python-2-vs-python-3/.

[52] ANY.RUN, *Malware research with ANY.RUN*, (Zugegriffen am: 07.01.2023). Adresse: https://any.run/why-us/.

[53] ANY.RUN, *FASTEST MALWARE ANALYSIS*, (Zugegriffen am: 12.12.2022). Adresse: https://any.run/features/.

[54] ANY.RUN, *Guide to Malware Detection with ANY.RUN*, (Zugegriffen am: 22.01.2023). Adresse: https://any.run/cybersecurity-blog/malware-detection-guide/.

[55] The MITRE Corporation, *ATT&CK Matrix for Enterprise*, (Zugegeriffen am: 11.01.2023). Adresse: https://attack.mitre.org/.

[56] B. L. Pearson, *The Complete Guide to Working With JSON*, Apr. 2021. Adresse: https://www.nylas.com/blog/the-complete-guide-to-working-with-json/.

[57] VMRAY, „Buyer's Guide:Malware Sandbox Solutions", Techn. Ber., 2021.

[58] Devcon, *A Window into Malicious Advertising - 61 % of malvertising targets Windows devices*, (Zugegriffen am: 22.01.2023), Nov. 2019. Adresse: https://adtech-security.com/blog/2019/11/26/a-window-into-malicious-adversiting-61-of-malvertising-targets-windows-devices.

[59] T. Fisher, *What Is an APK File?*, (Zugegriffen am: 22.01.2023), Okt. 2022. Adresse: https://www.lifewire.com/apk-file-4152929.

[60] M. Chand, *What Is the Most Popular Operating System?*, (Zugegriffen am: 20.01.2023), Juni 2019. Adresse: https://www.c-sharpcorner.com/article/what-is-the-most-popular-operating-system/.

[61] S. Williams-Shaw, *What is Security Automation? A Beginner's Guide*, (Zugegriffen am: 13.01.2023), Juli 2022. Adresse: https://swimlane.com/blog/security-automation.

[62] A. S. Gillis, *REST API (RESTful API)*, (Zugegriffen am: 10.01.2023). Adresse: https://www.computerweekly.com/de/definition/RESTful-API.

[63] ANY.RUN, *MITM proxy and FakeNet*, (Zugegriffen am: 10.01.2023), Apr. 2022. Adresse: https://any.run/cybersecurity-blog/mitm-proxy-fake-net/.

[64] M. Mierke, *HTTP-Fehler 404 beheben: Seite nicht gefunden*, (Zugegriffen am: 22.01.2023), Dez. 2019. Adresse: https://www.heise.de/tipps-tricks/HTTP-Fehler-404-beheben-Seite-nicht-gefunden-4606405.html.

[65] abuse.ch, *Statistics*, (Zugegriffen am: 30.01.2023). Adresse: https://bazaar.abuse.ch/statistics/.

[66] ANY.RUN, *Malware Trends Tracker*, (Zugegriffen am: 09.01.2023). Adresse: https://any.run/malware-trends/.

[67] W. Shanks, „Enhancing incident response through forensic, memory analysis and malware sandboxing techniques", SANS Institute, Techn. Ber., 2021.

[68] Cuckoo Foundation, *Cuckoo Sandbox 2.0 Release Candidate 1*, (Zugegriffen am: 01.12.2022), Jan. 2016. Adresse: https://cuckoosandbox.org/blog/cuckoo-sandbox-v2-rc1.

[69] Cuckoo Foundation, *Cuckoo Sandbox 2.0.6*, (Zugegriffen am: 01.12.2022), Juni 2018. Adresse: https://cuckoosandbox.org/blog/206-interim-release.

[70] L. Tung, *Windows 10 vs Windows 7: Microsoft's newer OS is almost 'twice as secure'*, (Zugegriffen am: 30.01.2023), März 2018. Adresse: https://www.zdnet.com/article/windows-10-vs-windows-7-microsofts-newer-os-is-almost-twice-as-secure/.

[71] Cuckoo Foundation, *REST API*, (Zugegriffen am: 30.01.2023). Adresse: https://cuckoo.readthedocs.io/en/latest/usage/api/.

[72] Hybrid Analysis, *Network Simulation now live on Hybrid-Analysis!*, (Zugegriffen am: 30.01.2023), Dez. 2020. Adresse: https://hybrid-analysis.blogspot.com/2020/.

[73] Hybrid Analysis, *Benchmarking some popular public malware analysis services regarding their "Anti-VM" technology*, (Zugegriffen am: 30.01.2023), Feb. 2015. Adresse: https://hybrid-analysis.blogspot.com/2015/02/benchmarking-some-popular-public.html.

[74] Hybrid Analysis, *Falcon Sandbox Public API*, (Zugegriffen am: 30.01.2023). Adresse: https://www.hybrid-analysis.com/docs/api/v2.

[75] G. Yip, *What, why, and when do we automate?*, (Zugegriffen am: 30.01.2023), Jan. 2018. Adresse: https://www.infoworld.com/article/3251068/what-why-and-when-do-we-automate.html.

[76] Cuckoo Foundation, *CAPE Sandbox Book*, (Zugegriffen am: 30.01.2023). Adresse: https://capev2.readthedocs.io/en/latest/.

[77] UtopianKnight, *Cuckoo Installation on Ubuntu 20*, (Zugegriffen am: 30.01.2023), Juni 2022. Adresse: https://utopianknight.com/malware/cuckoo-installation-on-ubuntu-20/.

[78] ForeGuards, *Cuckoo-Installation-Guide*, (Zugegriffen am: 30.01.2023), Jan. 2022. Adresse: https://github.com/ForeGuards/Cuckoo-Installation-Guide/blob/main/installation.txt.

[79] S. Wilkins, *How to Configure a Static IP Address in Windows 7*, (Zugegriffen am: 31.01.2023), Aug. 2010. Adresse: https://www.pluralsight.com/blog/it-ops/windows-7-ip-addressing.

Eidesstattliche Erklärung

Hiermit versichere ich – Jonas Schneidewind – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 31. Januar 2023 

Ort, Datum