HAW HAMBURG

BACHELORTHESIS
Dominik Hipke

Möglichkeiten zur Erkennung von Crypto-Ransomware durch File-Integrity-Monitoring

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Dominik Hipke

Möglichkeiten zur Erkennung von Crypto-Ransomware durch File-Integrity-Monitoring

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter: Prof. Dr. Bettina Buth

Eingereicht am: 16. Juli 2025

Dominik Hipke

Thema der Arbeit

Möglichkeiten zur Erkennung von Crypto-Ransomware durch File-Integrity-Monitoring

Stichworte

File-Integrity-Monitoring, Crypto-Ransomware, Detektionstechniken

Kurzzusammenfassung

Kryptographische Ransomwares sind in der modernen Bedrohungslandschaft eine ernstzunehmende Gefahr, die häufig zu systemweiten Ausfällen führt. Trotz umfangreicher Sicherheitsmaßnahmen können solche Angriffe nicht vollständig ausgeschlossen werden, weshalb die Erkennung eine maßgebliche Rolle spielt, um den Schaden zu begrenzen. In diesem Zusammenhang werden zunächst verschiedene Techniken zur Detektion von Crypto-Ransomware aus bestehender Literatur erarbeitet. Auf dieser Grundlage wird ein hostbasierter File-Integrity-Monitor für das Windows-Betriebssystem zu entwickeln, der bei einer erkannten Verschlüsselung in konfigurierten Verzeichnissen eine Alarmierung auslöst. Die Detektion erfolgt unter Nutzung der Entropie in Form der Chi-Square-Statistik und der Veränderung des Dateityps, wobei zu Beginn in einer Baseline die Werte der zu überwachenden Dateien als gutartig angenommene Referenz gespeichert und mit den Werten einer modifizierten Dateivariante verglichen werden. Als zusätzlichen Faktor werden versteckte Decoys in dem Dateisystem platziert. Auffällige Ereignisse werden von einem entworfenen Modell anhand von Regeln und Grenzwerten bewertet, die gegebenenfalls zu einer Alarmierung führen. In der Evaluation wurde der File-Integrity-Monitor basierend auf 15 Ransomware-Samples aus verschiedenen Familien getestet. Dabei konnte die Verschlüsselung in 14 der getesteten Fälle detektiert und eine entsprechende Alarmierung ausgelöst werden. In erster Linie ist jedoch eine mangelnde Robustheit des File-Integrity-Monitors gegenüber einer Manipulation durch Ransomware festgestellt worden. Die Merkmale und die Platzierung der Decoys erwiesen sich als effektive Möglichkeiten, auch wenn vereinzelt Schwachstellen und Ausnahmen kenntlich wurden.

Dominik Hipke

Title of Thesis

Possibilities for detecting crypto ransomware through file integrity monitoring

Keywords

File Integrity Monitoring, Crypto Ransomware, Detection Techniques

Abstract

Cryptographic ransomware is a serious threat in the modern threat landscape, frequently leading to system-wide outages. Despite extensive security measures, such attacks cannot be completely ruled out, making detection crucial to minimize damage. In this context various techniques for detecting crypto ransomware are initially compiled from existing literature to develop a host-based file integrity monitor for the Windows operating system that triggers an alert when encryption is detected in configured directories. Detection is performed using entropy in the form of the chi-square test and changes in file type. Initially the values of the files to be monitored are stored in a baseline, assumed to be benign, and are compared with the values of a modified file variant. Additionally hidden decoys are placed within the filesystem. A designed model evaluates discrepancies based on rules and thresholds, potentially triggering an alert. In the evaluation the file integrity monitor was tested using 15 ransomware samples from various popular families. The encryption of 14 samples was detected, and an alert was triggered. However, the monitor's lack of robustness against manipulation by ransomware was identified. The features and placement of the decoys proved to be effective, although a few vulnerabilities and exceptions were identified.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AGD	Automatically Generated Domain
BSI	Bundesamt für Sicherheit in der Informationstechnik
C2, C&C	Command und Control (Server)
DGA	Domain Generation Algorithm
FIM	File-Integrity-Monitor
ML	Machine Learning
PE	Portable Executable

Glossar

Accuracy Verhältnis zwischen korrekt klassifizierten Vorhersagen zu gesamten Vorhersagen (Hasarat, 2024):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Drive-By-Download Automatisiertes herunterladen von Schadsoftware auf ein Endgerät, ohne dass der Nutzer den Vorgang bemerkt (Baker, 2023)

False-Positive-Rate Verhältnis der fälschlich als positiv erkannten Vorhersagen zu allen tatsächlich negativen Fällen:

$$False\text{-}Positive\text{-}Rate = \frac{FP}{FP+TN}$$

FileHandle Eine Referenz auf eine Dateiressource, auf die ein Programm zugreifen kann

F1-Score Beschreibt das harmonische Mittel zwischen Precision und Recall (Hasarat, 2024):

$$F1\text{-}Score = 2 * \frac{Precision*Recall}{Precision+Recall}$$

Precision Verhältnis der korrekt erkannten positiven Vorhersagen zu allen als positiv vorhergesagten Fällen (Hasarat, 2024):

$$Precision = \frac{TP}{TP+FP}$$

Recall Verhältnis der korrekt erkannten positiven Vorhersagen zu allen tatsächlich positiven Vorhersagen (Hasarat, 2024):

$$Recall = \frac{TP}{TP+FN}$$

1 Einleitung

Computer und Informationssysteme helfen uns dabei, Prozesse zu vereinfachen und die Effizienz zu steigern, sei es im privaten Alltag oder in der Arbeitswelt. So gut diese Vorteile auch sind, kommen sie jedoch auch mit Nachteilen daher. Darunter sind digitale Systeme und Netzwerke attraktive Ziele für Cyberkriminelle.

Eine von Cyberkriminellen häufig genutzte Malware-Variante ist die Ransomware, welche das Opfer typischerweise vor die Entscheidung stellt, entweder ein Lösegeld zu zahlen oder den Zugriff auf seine Daten zu verlieren (Kosinski, 2024). Immer wieder berichten die Nachrichten darüber, dass solche erpresserischen Angriffe auf Organisationen stattfinden. Selbst große Unternehmen wie die Continental AG (Verfürden & Tyborski, 2022) oder die Deutsche Bahn AG (Briegleb, 2017) sind Opfer von Ransomware geworden.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) sieht Ransomware-Angriffe als eine der gravierendsten Bedrohungen in der Cybersicherheit. Zudem stellt die Erpressung von Unternehmen den am schnellsten wachsenden Bereich in der Cyberkriminalität dar. Dies ergibt sich unter anderem daraus, dass es für die Angreifer ein lukratives Geschäft darstellt, bei dem sich durch die Erpressung in kurzer Zeit hohe Geldbeträge erzielen lassen. Außerdem sind die Hürden, um einen Ransomware-Angriff zu vollziehen, gering, da wenig Vorkenntnisse benötigt werden und sich die Strafverfolgung als schwierig gestaltet (BSI, o. D.).

Aus diesen Gründen ist es in den letzten Jahren immer wichtiger geworden, sich umfassend durch diverse Präventionsmaßnahmen vor Ransomware zu schützen. Allerdings sind diese derartige Schutzmaßnahmen nicht immer ausreichend, sodass es dennoch zu einer Kompromittierung eines Systems kommen kann (Gómez-Hernández et al., 2018). Daher ist es entscheidend, weitere Instanzen wie die Erkennung von Ransomware abzudecken, um zusätzlichen Schutz zu gewährleisten.

Auf einem mit Crypto-Ransomware infizierten System lassen sich in der Regel Auswirkungen auf das Dateisystem feststellen, da die vorhandenen Dateien durch die Verschlüsselung verändert werden. Dieses Verhalten kann zur Erkennung eines Angriffs genutzt werden, indem das Dateisystem direkt auf dem Endgerät überwacht wird. Die vorliegende Arbeit macht sich diesen Ansatz zunutze.

1.1 Ziel dieser Arbeit

Das Ziel dieser Arbeit besteht darin, einen hostbasierten File-Integrity-Monitor zu entwickeln, welcher die Eigenschaft hat, Ransomware zu erkennen. Die Erkennung zielt dabei konkret auf Crypto-Ransomware ab, eine Art von Ransomware, welche Dateien mittels kryptographischer Verfahren verschlüsselt.

Durch die Überwachung und Analyse von Änderungen an Dateien sollen Eigenschaften, wie diese bei einer Verschlüsselung durch Ransomware entstehen, detektiert werden. Bei der Erkennung dieser Eigenschaften wird der Benutzer umgehend benachrichtigt, um eine schnelle Reaktion zur Abwendung eines größeren Schadens zu ermöglichen.

Die entwickelte Software wird anhand von Tests mit realen Varianten verschiedener Ransomware-Programme, sogenannten Ransomware-Samples, in einer isolierten und sicheren Umgebung evaluiert, um ihre Wirksamkeit zu beurteilen.

1.2 Abgrenzung

Durch den Fokus auf die Erkennung von Ransomware grenzt sich diese Arbeit deutlich von präventiven Maßnahmen ab, bei denen beispielsweise Mitarbeiter regelmäßige Schulungen erhalten, Systeme stets aktualisiert oder Backup-Strategien aufgefahren werden (Denham & Thompson, 2023). Dabei liegt der Schwerpunkt ausschließlich auf der Detektion der Variante der kryptographischen Ransomware.

Die Entwicklung des Monitors ist für Windows vorgesehen, da dieses Betriebssystem neben macOS und Linux die meisten Ransomware-Angriffe verzeichnet. Maßnahmen wie das automatische Stoppen oder Isolieren von Prozessen im Falle einer Ransomware-Erkennung werden aufgrund des Umfangs nicht berücksichtigt.

Es werden Detektionstechniken auf Basis von Machine-Learning-Modellen einbezogen, ohne jedoch auf die Details und die zugrundeliegenden Machine-Learning-Techniken einzugehen. Auf den Einsatz von Machine-Learning wird bei der Implementierung ebenfalls verzichtet.

1.3 Zielgruppe

Diese Arbeit beinhaltet neben den grundlegenden Informationen auch technische Einblicke zu Ransomware und deren Erkennung. Damit richtet sich diese Arbeit nicht nur an Informatiker mit Interesse an der IT-Sicherheit bzgl. Ransomware, sondern auch primär an IT-Sicherheitsexperten, die sich mit der theoretischen und methodischen Seite der Bedrohungserkennung beschäftigen.

1.4 Struktur der Arbeit

Diese Arbeit gliedert sich in einen theoretischen und einen praktischen Teil und umfasst insgesamt sechs Kapitel.

Kapitel zwei widmet sich nach der Einleitung den Grundlagen der Ransomware. Dabei steht zum einen die Erklärung des Begriffs Ransomware und zum anderen die Betrachtung unterschiedlicher Arten von Ransomware im Fokus. Außerdem wird auf das Bedrohungspotenzial aufmerksam gemacht, bei dem etwaige Auswirkungen und Schäden beschrieben werden. Zuletzt erfolgen wichtige Erläuterungen zu der Funktionsweise von Ransomware.

Das dritte Kapitel behandelt die Techniken zur Detektion von kryptographischer Ransomware. Es werden verschiedene Ansätze aus wissenschaftlichen Publikationen betrachtet, um eine Auswahl der Detektionstechniken zu treffen, die für den hostbasierten File-Integrity-Monitor relevant sind. Diese Techniken werden in die statische und dynamische Analyse unterteilt, wobei im weiteren Verlauf zusätzlich die damit verbundenen Herausforderungen beleuchtet werden.

Im anschließenden praktischen Teil, der in Kapitel vier behandelt wird, liegt der Fokus auf der Umsetzung des Hauptziels dieser Arbeit: der Entwicklung eines hostbasierten File-Integrity-Monitors zur Erkennung von Crypto-Ransomware. Hierbei werden die funktionalen und nichtfunktionalen Anforderungen der Anwendung definiert, um eine klare Grundlage für die

Entwicklung zu schaffen. Ebenfalls wird das Konzept zur Erkennung einer Verschlüsselung durch Ransomware aufgezeigt, das bei der Entwicklung angewendet wird. Im weiteren Verlauf wird auf die Architektur, verwendete Tools und Technologien sowie prägnante Implementierungsdetails eingegangen.

Um die Wirksamkeit der Erkennung von Crypto-Ransomware zu messen, wird daraufhin in Kapitel fünf eine Evaluation des File-Integrity-Monitors stattfinden, die mittels realen Ransomware-Samples in einer gesicherten Testumgebung erfolgt.

Im letzten Kapitel findet eine abschließende Betrachtung der Arbeit statt, bei der die gesammelten Erkenntnisse in einem Fazit zusammengefasst werden und ein Ausblick weitere Optimierungsmöglichkeiten und Forschungsfragen aufzeigt.

2 Grundlagen zu Ransomware

Für die weiteren Kapitel dieser Arbeit werden die Grundlagen zu Ransomware benötigt. Darunter sind vor allem die Erläuterung des Begriffs, die verschiedenen Arten und die Funktionsweise von entscheidender Bedeutung. Weitere Unterkapitel wie das Bedrohungspotenzial verdeutlichen das Ausmaß der Gefahr, die von Ransomware ausgeht.

2.1 Begriffserklärung

Ransomware ist eine spezielle Form von schädlicher Software, deren Name sich aus den englischen Wörtern „Ransom“ (Lösegeld) und „Malware“ zusammensetzt. Letzteres ist eine Kurzform für „Malicious Software“ und beschreibt ein Programm mit bösartigen Absichten. Das Ziel von Ransomware ist es, durch die Erpressung einen Geldbetrag vom Opfer zu fordern (Kosinski, 2024).

Die Angreifer können zur Erpressung verschiedene Druckmittel verwenden. Diese beziehen sich grundsätzlich auf die Daten, die sich auf dem System des Opfers befinden, wodurch Ransomware auf die Verletzung der Schutzziele aus der Informationssicherheit ausgerichtet ist. Diese Schutzziele umfassen die Verfügbarkeit, die Integrität und die Vertraulichkeit der Daten und sind unter der CIA-Triade bekannt (CyberOne, 2020).

Die Verfügbarkeit kann beispielsweise verletzt werden, wenn die Ransomware das Opfer vor seinem eigenen Computer bzw. System aussperrt (Oz et al., 2022), sodass dieser kein Zugriff mehr auf seine Daten erhält. Eine Verschlüsselung der Daten durch Ransomware stellt eine unautorisierte Veränderung dar, womit die Integrität verletzt wird. Durch die Drohung, sensible und personenbezogene Daten zu veröffentlichen (Oelmaier et al., 2023, S. 31), kann ein Ransomware-Angreifer das Schutzziel der Vertraulichkeit zur Erpressung verwenden. Je

nachdem, welche Ransomware-Variante betrachtet wird, werden einige oder sogar alle der drei Schutzziele systematisch verletzt.

Die Zahlung des Lösegelds soll dazu führen, dass das Opfer wieder vollumfänglichen Zugang zu seinem System und seine Daten erhält bzw. eine Veröffentlichung der Daten verhindert wird. Viele Angreifer halten sich an ihre Zusagen, um nach außen hin eine gewisse Verlässlichkeit zu vermitteln und auch zukünftige Opfer zu einer Zahlung zu bewegen (Dargahi et al., 2019). Allerdings gibt es keine hundertprozentige Garantie dafür, dass sie ihren Versprechungen nachkommen.

2.2 Arten

In diesem Abschnitt geht es zunächst darum, die Arten von Ransomware kennenzulernen, ohne bereits auf Details der Umsetzung und Funktionsweise einzugehen.

Die International Business Machines Corporation (IBM) unterscheidet zwischen zwei allgemeinen Arten von Ransomware: der Crypto-Ransomware und der Locker-Ransomware. Des Weiteren unterscheiden sie zwischen Unterkategorien (siehe Abschnitt 2.2.3; Kosinski, 2024).

2.2.1 Crypto-Ransomware

Die am häufigsten verbreitete Ransomware-Art ist die Crypto-Ransomware, welche sich kryptographische Verfahren zunutze macht, um die Dateien des Opfers zu verschlüsseln. Die Daten befinden sich zwar in verschlüsselter Form auf dem System des Opfers, können von ihm jedoch weder eingesehen noch verwendet werden, bevor eine Entschlüsselung erfolgt (Kosinski, 2024).

Die Angreifer verwenden symmetrische, asymmetrische oder die Kombination aus beiden, ein hybrides Verschlüsselungsverfahren. Solange eine starke Kryptographie eingesetzt wird und der Schlüssel ausschließlich im Besitz des Angreifers ist, hat das Opfer in der Praxis keine Möglichkeit, die Dateien selbst zu entschlüsseln (Genç et al., 2018).

Im schlechtesten Fall werden wichtige Daten wie Dokumente oder Datenbanken verschlüsselt, von denen keine Sicherungskopie existiert. Dadurch sieht sich das Opfer dazu gezwungen, das Lösegeld zu zahlen, in der Hoffnung, wieder verlustfrei an seine Daten zu gelangen.

2.2.2 Locker-Ransomware

Ähnlich wie die Crypto-Ransomware verhindert auch die Locker-Ransomware den Zugriff auf die Daten des Opfers. Der wesentliche Unterschied besteht darin, dass die Daten nicht verschlüsselt werden, sondern der Zugriff auf das System verwehrt wird, wodurch der Benutzer nicht mehr auf Anwendungen und Dateien zugreifen kann. Ransomwares dieser Art erreichen dies beispielsweise durch das Persistieren eines Sperrbildschirms sowie dem Abfangen von diversen Tastatureingaben, die das Ausbrechen aus dem Sperrbildschirm ermöglichen könnten (Oz et al., 2022).

Der Schaden, der durch Locker-Ransomwares entsteht, ist in den meisten Fällen geringer als der von Crypto-Ransomwares. Das liegt daran, dass die Dateien nicht verschlüsselt sind und gerettet werden können, wodurch das Opfer das Lösegeld nicht zahlen muss. Daher ist diese Art von Ransomware auch weniger verbreitet (Gómez-Hernández et al., 2018).

2.2.3 Weitere Arten

Die folgenden Arten basieren häufig auf der Crypto- oder Locker-Ransomware und kombinieren diese mit weiteren Eigenschaften, um den erpresserischen Druck auf das Opfer zu erhöhen.

Leakware

Leakware, auch unter den Namen Doxware bekannt, verfolgt das Ziel der Datenexfiltration. Dabei werden die Daten des Opfers über das Netzwerk an den Angreifer übertragen, um anschließend mit einer Veröffentlichung dieser Daten zu drohen (Kosinski, 2024).

So nutzen Crypto-Ransomwares oftmals zur Verschlüsselung der Dateien das Druckmittel der Datenveröffentlichung. Die Vorgehensweise zielt darauf ab, das Opfer auch dann zur Zahlung zu bewegen, wenn eine Wiederherstellung der Daten über eine nicht kompromittierte Sicherheitskopie möglich ist. In diesem Kontext ist von einer Ransomware mit der Eigenschaft einer doppelten Erpressung die Rede (Oelmaier et al., 2023, S. 10; 31).

Wipers

Wipers zeichnen sich dadurch aus, die Daten auf dem infizierten System irreversibel zu zerstören bzw. zu löschen (Kosinski, 2024). Ein Beispiel findet sich bei der Ransomware namens „FrenchLocker“, welche nach der Verschlüsselung der Dateien alle zehn Minuten eine Datei löscht und so dem Opfer zusätzlichen Druck zur Lösegeldzahlung auferlegt (Dargahi et al., 2019).

Scareware

Scarewares versuchen das Opfer auf verschiedene Weisen in Angst und Panik zu versetzen, damit dieses einer Geldforderung nachkommt. Eine beispielhafte Vorgehensweise ist es sich als eine seriöse Organisation auszugeben und das Opfer einer Straftat zu beschuldigen, um daraufhin eine Geldstrafe zu verlangen (Kosinski, 2024).

Andere Varianten täuschen anhand einer Warnnachricht vor, dass sich Malware auf dem System befindet und der Erwerb eines vermeintlichen Antivirenprogramms notwendig sei. Dabei kann mit dem Download des vermeintlichen Schutzprogramms tatsächlich eine Ransomware auf das System gelangen und es infizieren. Damit ist Scareware nicht nur eine Art Ransomware, sondern kann auch zur Verbreitung von Ransomware genutzt werden (Kosinski, 2024).

2.3 Ransomware-gefährdete Betriebssysteme

Der jährliche Cybersicherheitsbericht von Trend Micro zeigt, dass Windows im Zeitraum von 2019 bis 2023 – unter den drei untersuchten Betriebssystemen Windows, macOS und Linux – stets die meisten Ransomware-Angriffe verzeichnete. Im Jahre 2023 wurden 92 % der Angriffe unter Windows, 5 % unter macOS und 3 % unter Linux erkannt (Trend Micro, 2024). Damit gehört Windows zu den stark gefährdeten Betriebssystemen im Hinblick auf einen Ransomware-Angriff.

Dass Ransomware unter macOS weniger verbreitet ist, spiegelt sich in der Anzahl bekannter Ransomware-Familien wider. Zum Zeitpunkt der Erstellung dieser Arbeit konnten lediglich sechs Ransomware-Familien identifiziert werden: KeRanger, Patcher, FindZip, EvilQuest,

ThiefQuest (Kowalczyk et al., 2024) und MacRansom (Becker, 2017). Ausgeschlossen davon sind Familien, die mit einer plattformübergreifenden Programmiersprache entwickelt wurden.

Unter dem iOS-Betriebssystem sind mehrere Fälle einer Erpressung bekannt, bei denen es sich jedoch nicht um reale Ransomware handelte. Anstelle einer Infektion mit Schadsoftware wurde die Apple-ID kompromittiert, wodurch Angreifer über die „Mein iPhone suchen"-Funktion die Möglichkeit hatten, das Gerät zu sperren und eine Nachricht über eine Lösegeldforderung anzuzeigen. Die Betroffenen konnten ihr Smartphone jedoch selbst wieder entsperren (Grabmair, 2016). Es wird vermutet, dass iOS nicht von Ransomware betroffen ist, weil die Applikationen vor der Veröffentlichung im App-Store sorgfältig überprüft werden (Oz et al., 2022).

Das Android-Betriebssystem ist im Vergleich zu iOS deutlich attraktiver für Angreifer. Kaspersky, ein Unternehmen, das Sicherheitssoftware entwickelt, hat im Jahre 2019 rund 68.000 App-Installationspakete für Android identifiziert, die Ransomware enthielten. Allerdings ist die Anzahl in den darauffolgenden Jahren deutlich zurückgegangen, sodass im Jahre 2023 nur noch 11.202 Ransomware-Installationsprogramme identifiziert wurden (Kivva, 2024; Shishkova & Kivva, 2022).

2.4 Bedrohungspotenzial

Das Ausmaß eines Ransomware-Angriffs variiert je nach Zielgruppe und kann unterschiedliche Schäden nach sich ziehen, die nicht nur das Opfer selbst betreffen, sondern auch indirekt weitere Akteure betreffen können. Das Ziel dieses Kapitels ist es, ein Verständnis für die möglichen Folgen zu vermitteln.

2.4.1 Zielgruppe

In der Vergangenheit waren die Angriffe vor allem auf einzelne Personen ausgerichtet, die ihre Endgeräte wie Smartphones oder Computer im Alltag verwenden. Mittlerweile sind Organisationen wie Finanzinstitute, Einrichtungen aus dem Gesundheits- und Bildungssektor, Regierungen sowie Unternehmen im Visier der Angreifer (Beaman et al., 2021).

Organisationen sind oft bevorzugt, weil der finanzielle Gewinn höher ausfällt (BSI, 2022). Dennoch sollte die Gefährdung von Einzelpersonen nicht vernachlässigt werden, da diese oft

über weniger umfangreiche sicherheitstechnische Vorkehrungen verfügen und deshalb schlechter vor einem Cyber-Angriff geschützt sind. Selbst viele kleine Lösegeldzahlungen von Einzelpersonen können den Angreifern insgesamt eine beträchtliche Summe einbringen. Insbesondere wenn Daten wie Erinnerungsfotos kompromittiert werden, die für das Opfer einen emotionalen Wert darstellen, kann die Bereitschaft zur Zahlung des Lösegelds erhöht sein (Oelmaier et al., 2023, S. 42).

Die primäre Motivation der Angreifer muss nicht zwangsläufig ein finanzieller Gewinn sein. Ransomware kann auch eingesetzt werden, um bewusst Schaden bei einem Opfer zu verursachen (BSI, 2022). Dadurch ist die Zielgruppe nicht nur durch die potenzielle Zahlungsfähigkeit definiert, sondern auch durch eine absichtliche Schädigung.

2.4.2 Wirtschaftlicher Schaden

Das BSI gibt in einem Bericht über die IT-Sicherheitslage 2024 an, dass durch Ransomware-Angriffe global rund 1,1 Milliarden US-Dollar an Lösegeld erlangt wurden. Dabei wird vermutet, dass die Dunkelziffer weitaus höher liegen dürfte (BSI, 2024). Neben der Zahlung des Lösegelds sind auch wirtschaftliche Schäden bei der Wiederherstellung der Systeme oder langen Betriebsunterbrechungen zu beachten.

Das Unternehmen Sophos hat zwischen Januar und Februar 2024 eine Umfrage unter 5.000 IT-Sicherheitsverantwortlichen in 14 Ländern durchgeführt. Dabei ist festgestellt worden, dass 59 % der Organisationen Opfer eines Ransomware-Angriffs geworden sind. Die durchschnittliche Höhe der Lösegeldforderung lag bei ca. 4,3 Millionen Dollar und die durchschnittlichen Kosten zur Wiederherstellung bei ca. 2,7 Millionen Dollar. Die Wiederherstellungskosten beziehen sich auf Faktoren wie die Ausfallzeit, Hardwarekosten und zusätzliche Arbeitszeit. Im Falle einer erfolgreichen Kompromittierung des Backups ist die Lösegeldforderung im Durchschnitt doppelt so hoch als bei den Angriffen, bei denen das Backup nicht kompromittiert wurde. Angemerkt sei, dass die Höhe der Lösegeldforderung und die Wiederherstellungskosten stark von der Größe und dem Umsatz der Organisation abhängt (Sophos, 2024).

Außerdem sollte ein möglicher Reputationsverlust von Unternehmen berücksichtigt werden, der als Folge eines Angriffs entstehen kann. Dies kann sich negativ auf die wirtschaftliche Lage

eines Unternehmens auswirken, etwa durch die Abwanderung bestehender Kunden sowie die Erschwernis der Neukundengewinnung (BSI, 2022).

Des Weiteren kann es zu wirtschaftliche Fremdschäden kommen, wenn Unternehmen aufgrund eines Angriffs ihre vertraglichen Verpflichtungen gegenüber anderen Unternehmen, wie etwa der pünktlichen Lieferung von Waren, nicht mehr erfüllen können (BSI, 2022).

Auch wenn Institutionen wie das BSI ausdrücklich davon abraten, das Lösegeld zu zahlen (BSI, o. D.), ist es für viele Unternehmen dennoch verlockend, insbesondere dann, wenn die Kosten für die Wiederherstellung des Systems deutlich höher sind als das Lösegeld an sich (Liska & Gallo, 2016, S. 27).

2.4.3 Auswirkungen auf die Gesellschaft und indirekt Betroffene

Insbesondere durch den Angriff auf kritische Organisationen wie Banken, Bildungseinrichtungen oder medizinische Einrichtungen können über den wirtschaftlichen Verlusten hinaus Auswirkungen für indirekt betroffene Parteien und die Gesellschaft bedeuten.

MacColl et al. (2024) untersuchen die weitreichenden Schäden, die durch Ransomware-Angriffe verursacht werden und beleuchten unter anderem die gesellschaftlichen Auswirkungen. Ein besonders relevanter Aspekt ist die Störung von essenziellen Dienstleistungen, die von verschiedenen Organisationen wie Krankenhäuser oder Schulen erbracht werden.

Laut einer Studie, die 374 Ransomware-Vorfälle auf medizinische Einrichtungen in den USA untersuchte, verdoppelte sich die Anzahl der jährlichen Angriffe zwischen 2016 und 2021 von 43 auf 91 (Neprash et al., 2022). Derartige Angriffe führten dazu, dass digitale Technologien ausfielen, entweder als direkte Folge des Angriffs oder als Sicherheitsmaßnahme durch das IT-Personal, wodurch die Patientenversorgung gestört wurde und es zu massiven Verzögerungen kam (Van Boven et al., 2023).

Des Weiteren betonen MacColl et al. (2024), dass auch Angriffe auf Schulen und Hochschulen in den letzten Jahren einen deutlichen Zuwachs zeigten. Zwar konnten keine langanhaltenden Auswirkungen auf die Lernerfolge nachgewiesen werden, allerdings kam es insbesondere bei den organisatorischen Abläufen zu massiven Störungen.

Ein weiterer wichtiger Aspekt sind Opportunitätskosten, die hauptsächlich im öffentlichen Sektor durch die Wiederherstellungskosten eines Angriffs entstehen. Es werden Ressourcen von anderen priorisierten Bereichen wie die Digitalisierung, Bildung oder soziale Unterstützung gestrichen und für die Bewältigung des Angriffs eingesetzt. Während diese Art von Auswirkung allerdings weniger greifbar für die Bürger ist, ist der Vertrauensverlust in die Verlässlichkeit von öffentlichen Diensten ein bedeutsamerer Punkt, was in einer Studie im Zusammenhang mit einem Ransomware-Angriff auf ein Düsseldorfer Krankenhaus festgestellt wurde (MacColl et al., 2024).

Selbst die Personen, die nur indirekt mit einer Organisation in Verbindung stehen, können Folgeschäden erleiden. Beispielsweise könnte eine Datenexfiltration eines Unternehmens dazu führen, dass die persönlichen Daten der Kunden auf dem digitalen Schwarzmarkt verkauft bzw. veröffentlicht werden. Folglich können diese Daten von Cyberkriminellen missbraucht werden, indem die Identität gestohlen wird, mit den Zahlungsinformationen Einkäufe getätigt werden (MacColl et al., 2024) oder die E-Mail-Adressen sowie Telefonnummern für Spam bzw. Phishing ausgenutzt werden. Ein weiteres Beispiel ist der Verlust von Arbeitsplätzen, wenn Unternehmen nicht in der Lage sind die wirtschaftlichen Schäden zu stemmen und deshalb eine Unternehmensschließung einleiten muss.

2.5 Funktionsweise

Liska und Gallo (2016) beschreiben den Ablauf eines Ransomware-Angriffs anhand von fünf Phasen, an denen sich dieses Kapitel orientiert. Es geht dabei um die Weise, wie sich Ransomware Zugang zu einem System verschafft, sich dort festsetzt, ggf. eine Kommunikationsverbindung zu den Angreifern aufbaut und den Schaden verursacht.

Der folgende Abschnitt behandelt eine detaillierte Erläuterung dieser einzelnen Phasen. Dabei sollte angemerkt werden, dass sich einzelne Punkte zwischen den Ransomware-Varianten unterscheiden können.

2.5.1 Verbreitung von Ransomware

In der Phase der Verbreitung von Ransomware versuchen die Angreifer auf verschiedene Weisen das Schadprogramm zu den potenziellen Opfern zu bringen, um diese zu infizieren. Die Angreifer können unter anderem zielgerichtete Angriffe beispielsweise auf wirtschaftlich starke Unternehmen oder breit angelegte Kampagnen starten, bei denen (schädliche) E-Mails verschickt werden (Liska & Gallo, 2016, S. 6-8).

Das Unternehmen Sophos zeigt in einer Statistik die am häufigsten genutzten Angriffsvektoren bei Ransomware-Angriffen auf Unternehmen. An der Spitze befindet sich das Ausnutzen von Sicherheitslücken mit 36 %, gefolgt von kompromittierten Anmeldeinformationen mit 29 %. Eine weitere Angriffsfläche stellen E-Mails dar, wobei 23 % bösartigen Schadcode oder einen Downloadlink zu Schadsoftware enthalten und 11 % auf Phishing-Mails entfallen, die den Empfänger täuscht und zur Preisgabe seiner (Anmelde-)Daten verleitet. Verhältnismäßig selten sind Brute-Force-Attacken mit 3 % (Sophos, 2024).

Neben diesen Angriffsvektoren ist auch das sogenannte Malvertising erwähnenswert, bei dem die Schadsoftware über bösartige Werbung auf Webseiten, entweder durch Download-Links oder Exploits, verteilt werden kann (Malwarebytes, o. D.).

Am Beispiel der im Jahre 2016 erstmals aufgetauchten Ransomware-Familie „Cerber“ werden einige Angriffsvektoren beispielhaft gezeigt. In der Vergangenheit versendete Cerber nicht nur bösartige E-Mails, sondern nutzte zudem Schwachstellen von im Internet bereitgestellten Diensten aus. Die E-Mails beinhalteten Dateianhänge in Form von Office-Dokumenten (z. B. .docx oder .dot), welche sogenannte Makros enthielten, die die Einbettung sowie Ausführung von Code innerhalb des Dokuments ermöglichen. Sobald die Makros aktiviert wurden, begann der Download der Ransomware von einem Server und der anschließenden Ausführung des Programms (SentinelOne, o. D.).

In den Anfangszeiten nutzte Cerber Schwachstellen der Software Adobe Flash aus, sodass bei dem Aufruf einer kompromittierten Website die Cerber-Ransomware automatisch über einen **Drive-By-Download** heruntergeladen und ausgeführt werden konnte (Bisson, 2017; Baker, 2023). Einige Jahre später wurde eine Schwachstelle in der Kollaborationsplattform Atlassian

Confluence ausgenutzt, wodurch unberechtigt Zugriff auf die Systeme erlangt und die Verschlüsselung gestartet wurde (Robles et al., 2023).

2.5.2 Installation

Wenn die Verbreitung der Ransomware erfolgreich war, startet die Installationsphase mit dem Ausführen des Schadcodes. In dieser Phase trifft die Ransomware in der Regel verschiedene Vorbereitungen und kann sich ggf. über das Netzwerk auf andere Geräte ausbreiten (Shaukat & Ribeiro, 2018).

Ähnlich wie andere Schadprogramme setzt Ransomware verschiedene Tarnmechanismen ein, um Antimalware-Systeme zu umgehen. Dazu werden z. B. Techniken wie das Injizieren des Schadcodes in einen vertrauenswürdigen Prozess verwendet (Kharraz et al., 2016). In einigen Fällen wird überprüft, ob das Programm in einer Malware-Analyseumgebung in Form einer Sandbox oder einer virtuellen Maschine ausgeführt wird. Falls eine solche Analyseumgebung identifiziert wurde, passt die Ransomware ihr Verhalten so an, dass sie harmlos erscheint, um nicht als bösartig eingestuft zu werden (Liska & Gallo, 2016, S. 9).

Zudem verwendet Ransomware Mechanismen, um persistent auf dem System zu bleiben, wodurch sie nach jedem Start des Betriebssystems automatisch ausgeführt wird und aktiv bleibt. Einige Crypto-Ransomwares nutzen dies, um eine durch einen Neustart unterbrochene Verschlüsselung fortzuführen oder neu erstellte Dateien des Benutzers zu verschlüsseln. Windows bietet verschiedene Möglichkeiten, die zum automatischen Starten eines Prozesses genutzt werden können. Darunter existiert z. B. die Möglichkeit über den Startup-Ordner, einen Eintrag in der Windows-Registry (Konfigurationsdatenbank), ein angelegter Task im Task Schedule oder die Nutzung von Windows-Diensten (Lemmou et al., 2020).

Insbesondere bei Angriffen auf Organisationen profitieren die Täter davon, wenn möglichst große Teile des Systems infiziert werden. Zu diesem Zweck setzt Ransomware auf Lateral Movement, um sich im Netzwerk auszubreiten und weitere Systeme zu infizieren (Dargahi et al., 2019). Diese Gruppe von Ransomwares wird daher auch als Ransomworm bezeichnet (Almashhadani et al., 2022) und nutzt unterschiedliche Ansätze zur Selbstausbreitung. Eine Möglichkeit besteht darin, dass geteilte Dateien im Netzwerk mit dem Schadcode versehen werden, sodass bislang nicht betroffene Systeme durch das Ausführen der Datei infiziert

werden. Eine weitere Methode ist die Ausnutzung von Sicherheitslücken, wie es die Ransomware-Familien NotPetya und WannaCry zeigten. Beide verwendeten unter Windows eine Schwachstelle in dem Server Message Block (SMB) Protokoll, das unter anderem Netzwerkdrucker verwaltet sowie geteilte Ordner und Dateien in einem Netzwerk zugänglich macht. Durch diese Sicherheitslücke gelang es den Angreifern das Schadprogramm aus der Ferne direkt auf die Computer im Netzwerk auszuführen. Auch das Remote-Desktop-Protocol (RDP), ein Protokoll für den Fernzugriff auf Computer, erwies sich als Angriffsvektor, bei dem ebenfalls Exploits zum Einsatz kamen (Dargahi et al., 2019).

Unter Windows ist standardmäßig der Volume Shadow Copy Service (VSS) aktiviert, der Schnappschüsse von Dateien erstellt. Die Schnappschüsse können als eine Art Backup dienen, da sie die Wiederherstellung älterer Dateiversionen ermöglichen, weshalb kryptographische Ransomwares diesen Dienst typischerweise vor der Verschlüsselung zurücksetzen (Oelmaier et al., 2023, S. 99).

2.5.3 Command and Control

Der Command and Control Server, auch C2 oder C&C Server genannt, ermöglicht die Kommunikation zwischen einem mit Malware infizierten System und den Angreifern. Dabei verschickt der C&C-Server nicht nur bösartige Anweisungen, sondern erhält auch Daten von dem System des Opfers (Lenaerts-Bergmans, 2023).

Ransomware nutzt eine solche Verbindung, um z. B. Informationen über das infizierte System abzufragen, wodurch die Angreifer abschätzen können, wie rentabel eine Erpressung ist. Im Rahmen von kryptographischer Ransomware kann der Verschlüsselungsprozess initiiert und Parameter wie die zu verschlüsselnden Dateitypen oder der Zeitpunkt der Verschlüsselung definiert werden. Außerdem werden in vielen Fällen die kryptographischen Schlüssel zwischen dem Angreifer und dem Opfer ausgetauscht (Liska & Gallo, 2016, S. 10f). Die Schlüssel können entweder in der Domäne des Angreifers erzeugt werden und über den C&C-Server zum Opfer übermittelt werden oder umgekehrt (Berrueta et al., 2019).

Der Verbindungsaufbau erfolgt in der Regel über IP-Adressen bzw. Domains, die in dem Schadcode eingebettet sind, oder über zur Laufzeit generierten Domains, sogenannten Automatically Generated Domains (AGD) (Cebere et al., 2024). Ein Nachteil eingebetteter

Adressen aus Sicht der Angreifer ist, dass diese entdeckt und von diversen Sicherheitslösungen blockiert werden können. AGDs kompensieren diesen Nachteil, da diese immer wieder neu generiert werden und nicht vorhersehbar sind (Liska & Gallo, 2016).

Häufig werden Netzwerkprotokolle wie HTTP/S oder DNS für die Kommunikation zwischen dem C2-Server und dem Zielsystem verwendet, weil diese Protokolle aufgrund ihrer weit verbreiteten Nutzung durch legitime Anwendungen weniger auffällig und daher schwieriger zu identifizieren sind (Oelmaier et al., 2023, S. 78). Außerdem ist die Wahrscheinlichkeit geringer, dass diese Protokolle von Sicherheitstechnologien wie einer Firewall blockiert werden.

Bestimmte Ransomware-Varianten verursachen auch dann Schaden, wenn der C&C-Server nicht erreicht werden konnte, indem sie auf Fallback-Mechanismen zurückgreifen. Andere Varianten hingegen bleiben inaktiv, solange die Verbindung aussteht (Tang et al., 2020; Begovic et al., 2023).

2.5.4 Schadensverursachung

In dieser Phase entfaltet die Ransomware ihre schädliche Wirkung, indem sie nicht nur Dateien verschlüsselt oder den Benutzer den Zugang zum Computer verwehrt, sondern auch ggf. Daten entwendet oder Dateien unwiderruflich löscht. Im Nachfolgenden werden unter anderem die technischen Aspekte der verschiedenen Schadensarten dargestellt.

Blockieren des Zugangs zum System

Harun et al. (2021) beschreiben drei Arten, wie Locker-Ransomwares den Zugang zum System blockieren. Zum einen existiert die Möglichkeit des Screen-Lockings, bei dem die Benutzeroberfläche auf Basis des Betriebssystems blockiert wird. Unter Android kann dies beispielsweise durch das Deaktivieren der Navigationsleiste realisiert werden.

Eine weitere Möglichkeit beschränkt sich auf den Webbrowser und wird daher Browser Locking genannt. Das Opfer wird dabei auf eine bösartige Website geleitet, dessen JavaScript-Code jegliche Interaktion wie dem Schließen des Browsers oder der Eingabe von Tastaturbefehlen unterbindet (Pornasdoro, 2014; Oz et al., 2022).

Die letzte Möglichkeit beschreibt eine Sperrung auf Basis des Master Boot Records (MBR). Das MBR ist dafür zuständig, dass das installierte Betriebssystem beim Systemstart geladen wird. Die Schadsoftware manipuliert hierbei das MBR oder verschlüsselt dieses, sodass das Starten des Betriebssystems nicht mehr möglich ist (Oz et al., 2022).

Verschlüsselung von Dateien

Crypto-Ransomware verschlüsselt üblicherweise Dateien bestimmter Typen, wie Dokumente oder Datenbanken, die sich auf dem betroffenen System oder auf verbundenen Datenspeichern wie externen Festplatten und Netzlaufwerken befinden (Dargahi et al., 2019). Dateien, die für das Betriebssystem essenziell sind, bleiben von einer Verschlüsselung verschont, um die Funktionstüchtigkeit des Betriebssystems weiterhin zu gewährleisten (Shaukat & Ribeiro, 2018).

Zur Verschlüsselung werden die Implementierungen aus kryptographischen Bibliotheken verwendet. Einige Crypto-Ransomwares nutzen die Bibliothek des Betriebssystems, während andere eine Bibliothek in ihr Programm einbinden und damit ausliefern (Begovic et al., 2023). In der Vergangenheit wurden teilweise eigene Implementierungen eines Verschlüsselungsalgorithmus genutzt, die sich jedoch als unsicher erwiesen, da Programmierfehler eine Entschlüsselung ohne die Zahlung des Lösegelds ermöglichten (Kolodenker et al., 2017).

Wie bereits im Abschnitt der Ransomware-Arten beschrieben, kommt entweder die symmetrische, asymmetrische oder die hybride Verschlüsselungsform zum Einsatz, wobei jede Form auf unterschiedliche Art und Weise implementiert werden kann. Sowohl symmetrische als auch asymmetrische Schlüssel können zur Laufzeit lokal auf dem System des Opfers oder auf dem C&C-Server generiert werden. In der Regel werden die Schlüssel zwischen dem Angreifer und dem Opfer übertragen und ggf. nach der Verschlüsselung von dem System des Opfers wieder gelöscht. Eine weitere Möglichkeit ist, dass der Schlüssel bereits in der Ransomware-Binary eingebettet ist. Sollte ein symmetrischer Schlüssel eingebettet sein, so kann der Schlüssel durch Reverse-Engineering extrahiert werden und eine Entschlüsselung ohne eine Zahlung des Lösegelds erfolgen. In Bezug auf den asymmetrischen Schlüssel stellt dies kein Problem dar, weil ausschließlich der öffentliche Schlüssel eingebettet ist und der private Schlüssel bei dem Angreifer verbleibt. Wenn die Angreifer jedoch immer denselben öffentlichen Schlüssel in die Binary einbetten, kann der private Schlüssel veröffentlicht werden, weil beispielsweise jemand

das Lösegeld bezahlt hat. Dadurch hätten alle anderen Opfer ebenfalls die Chance, ihre Daten wiederzuerlangen (Bajpai & Enbody, 2020).

Ransomwares, die die symmetrische Verschlüsselung einsetzen, können entweder für jede Datei einen individuellen Schlüssel erzeugen und die Datei verschlüsseln oder alle Dateien mit demselben Schlüssel verschlüsseln. In einigen Fällen wurden die Schlüssel direkt in den (verschlüsselten) Dateien oder in Ordnern versteckt, was jedoch leicht entdeckt werden kann (Aboud & Mariyappn, 2021; Ploszek et al., 2021).

Andere Ransomware-Varianten hingegen setzen ausschließlich auf eine asymmetrische Verschlüsselung, obwohl diese in der Regel langsamer ist. Aus Sicht der Angreifer kann der asymmetrische Ansatz effektiver sein als der symmetrische (oder hybride) Ansatz, etwa wenn das Schlüsselpaar bei dem Angreifer erstellt wird und der private Schlüssel niemals die Maschine des Angreifers verlässt. Das liegt daran, dass die auf dem System des Opfers erzeugten symmetrischen Schlüssel theoretisch abgefangen und zur Entschlüsselung verwendet werden können (Kolodenker et al., 2017).

Am weitesten verbreitet ist das hybride Verschlüsselungsverfahren, da es eine Balance zwischen Sicherheit und Performance schafft. Üblicherweise wird bei diesem Verfahren für jede Datei ein individueller symmetrischer Schlüssel erzeugt und die Datei verschlüsselt. Anschließend wird der symmetrische Schlüssel gesichert, indem dieser mit dem öffentlichen Schlüssel des Angreifers verschlüsselt wird (Aboud & Mariyappn, 2021). Der verschlüsselte Schlüssel wird dann in die (verschlüsselte) Datei eingebettet (Kolodenker et al., 2017).

Datenexfiltration

Bei der Datenexfiltration stehen häufig personenbezogene Daten, E-Mails und Finanzdaten im Vordergrund der Angreifer. Dabei können die von Unternehmen ausgeflossenen Datenmengen bis in den Terabyte-Bereich reichen (Oelmaier et al., 2023, S. 9).

Die C2-Verbindung wird insbesondere bei einer groß angelegten Datenexfiltration üblicherweise nicht verwendet. Stattdessen kommen effiziente Protokolle wie das File Transfer Protocol (FTP) oder Secure Copy Protocol (SCP) zum Einsatz. Teilweise verwendet Ransomware

legitime Datenübertragung-Tools wie Rclone, die gegebenenfalls auf das System des Opfers installiert werden (Oelmaier et al., 2023, S. 97).

Neben der Erpressung einer Veröffentlichung der Daten, können die Informationen auch dazu genutzt werden, um die Höhe der Lösegeldforderung zu bestimmen, insbesondere wenn Daten vorliegen, die die finanzielle Lage des Opfers beschreiben (Williams, 2024).

Weitere Mittel zur Schadensverursachung

Die Verschlüsselung, Datenexfiltration und das Blockieren des Endgeräts sind nicht die einzigen Varianten zur Schadensverursachung. Einige Ransomwares setzen eine dreifache Erpressung an, bei der zusätzlich zur Verschlüsselung und Datenexfiltration ein DDoS-Angriff angedroht wird, welcher zur Lahmlegung von Internet-Diensten führt. Neben einem DDoS-Angriff kann auch mit der Löschung der Daten oder der Benachrichtigung von Geschäftspartnern, die aus den entwendeten Daten einer Organisation hervorgehen, gedroht werden (Oelmaier et al., 2023, S. 46f).

2.5.5 Erpressung

Im letzten Schritt, der Erpressung, macht sich die Ransomware oftmals bemerkbar, indem der Desktophintergrund geändert wird, ein Pop-Up-Fenster aufkommt oder Textdateien in Ordnern abgelegt werden, die häufig Namen wie „Restore-My-Files“ oder „!! READ ME !!“ tragen. Dabei wird die Information überbracht, dass das System infiziert wurde und mit welchen Druckmitteln die Erpressung erfolgt. Zum Teil druckten die Angreifer den Bedrohungstext über die vorkonfigurierten Netzwerkdrucker aus (Oelmaier et al., 2023, S. 112).

Eine zu zahlende Summe wird nicht immer direkt in dem Erpresserschreiben gefordert, stattdessen soll der Kontakt zu den Angreifern gesucht werden, bei dem die Lösegeldsumme verhandelt wird. Sofern eine Lösegeldsumme gefordert wird, werden in der Regel Kryptowährungen als Zahlungsmittel verwendet, da diese zur Anonymität der Angreifer beitragen (Oz et al., 2022; Oelmaier et al., 2023, S. 11).

3 Techniken zur Detektion von Crypto-Ransomware

In diesem Kapitel liegt der Fokus auf der Erläuterung der Techniken zur Detektion von kryptographischer Ransomware, die in wissenschaftlichen Publikationen untersucht wurden. Dabei werden die Techniken in die statische und dynamische Analyse gegliedert. Die Struktur (Abbildung 1) orientiert sich an den Autoren Kok et al. (2020) und Cen et al. (2023) und wurde durch zusätzliche Anpassungen ergänzt.

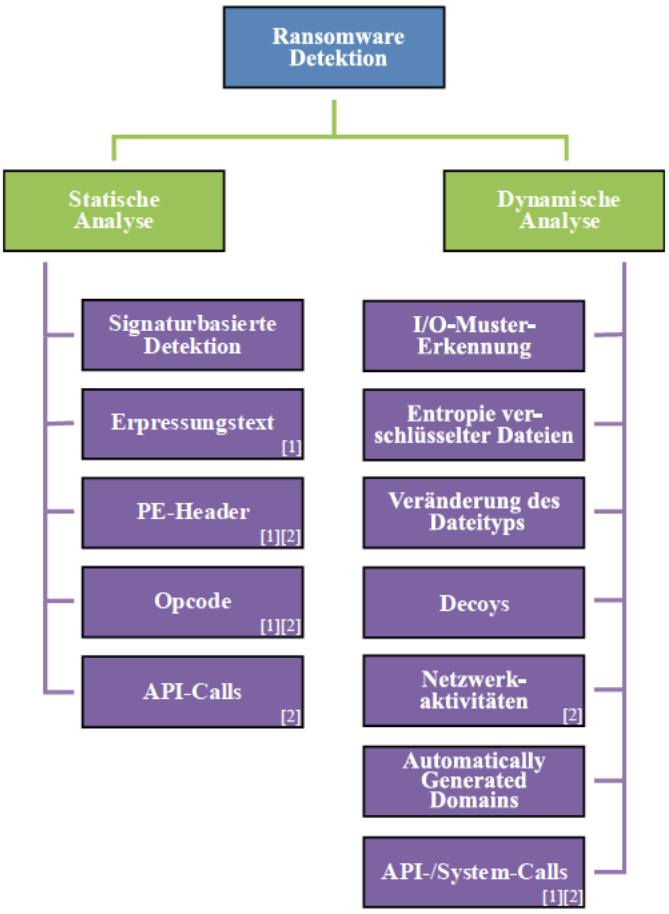

Abbildung 1: Aufbau der Ransomware-Detektion in Anlehnung an [1]: Kok et al. (2020) und [2]: Cen et al. (2023). Abbildung in Anlehnung an Kok et al. (2020)

Das grundlegende Ziel der meisten Autoren besteht darin, eine (binäre) Klassifizierung zwischen Ransomware und legitimer Software zu erreichen. Sowohl im statischen als auch dynamischen Analyseansatz gilt es Eigenschaften zu untersuchen und einzusetzen, die zur Differenzierung beider Klassen beitragen und eine hohe Allgemeingültigkeit bieten. Bei der Vorhersage entstehen vier mögliche Ereignisse:

(1) **True Positive (TP):** Das Modell klassifiziert eine Ransomware korrekt
(2) **True Negative (TN):** Das Modell klassifiziert eine legitime Software korrekt
(3) **False Positive (FP):** Das Modell klassifiziert eine legitime Software fälschlicherweise als Ransomware
(4) **False Negative (FN):** Das Modell klassifiziert eine Ransomware fälschlicherweise als gutartige Software

Für die einzelnen Detektionstechniken werden die Evaluationsergebnisse der Arbeiten entweder in Form des **F1-Scores** oder der **Accuracy** bekanntgegeben. Beides sind Metriken, die die Leistung der Klassifikationsmodelle unter der Berücksichtigung der o. g. Ereignisse bewerten und in Prozent angegeben werden. Der F1-Score wird bevorzugt, da diese Metrik aussagekräftiger für ungleichmäßig verteilte Datensätze ist, die zur Evaluation genutzt werden (Hasarat, 2024).

Einige Arbeiten kombinieren mehrere Techniken, wobei nicht eindeutig beschrieben wird, inwiefern eine Technik zum Ergebnis beigetragen hat. In diesen Fällen wird auf die Angabe der Resultate verzichtet.

3.1 Detektion auf Basis der statischen Analyse

Die statische Analyse beschreibt eine Methode zur Untersuchung von potenziell bösartigen Dateien und Programmen, ohne dass diese explizit ausgeführt werden. Hierfür werden Informationen und Eigenschaften direkt aus der Datei extrahiert, um zur Detektion Indikatoren oder unerwünschtes Verhalten der Anwendung zu entdecken.

3.1.1 Signaturbasierte Detektion

Die signaturbasierte Detektion ist in vielen Sicherheitssoftwares eine etablierte Praxis und wird zur Erkennung jeglicher Malware verwendet. Die Signatur stellt hierbei einen eindeutigen Fingerabdruck einer Malware auf Byte-Ebene dar. Sie kann auf vielfältige Weise erstellt werden und mehrere Eigenschaften beinhalten. Dazu zählen beispielsweise das Vorhandensein einer spezifischen Sequenz von Bytes, der Hashwert von Teilen oder des gesamten Dateiinhalts, die Dateigröße, importierte und exportierte Funktionen oder auch eingebettete Strings (SentinelOne, 2021).

In einer Datenbank werden die Signaturen von bekannten Schadprogrammen gesammelt (Edwards, 2023), sodass durch das Durchsuchen einer potenziell schädlichen Datei nach einer in der Datenbank vorhandenen Signatur eine Klassifikation in eine bösartige oder unauffällige Datei ermöglicht wird.

Oftmals werden durch die Angreifer mehrere Varianten entwickelt, wodurch verschiedene Dateien einer Malware-Familie entstehen. In diesem Fall versuchen Analysten oder ein automatisierter Prozess eine Signatur für möglichst viele Varianten zu finden (SentinelOne, 2021).

Ein offensichtlicher Nachteil besteht darin, dass in der Datenbank nur gegen bekannte Malware geprüft wird, wodurch diese Methode nicht gegen neue oder unbekannte Schadsoftware wirksam ist. Dennoch ist die signaturbasierte Erkennung weit verbreitet, weil sie sich durch ihre Effizienz sowie einer niedrigen False-Positive-Rate (FPR) auszeichnet (Sihwail et al., 2018). Um weiterhin False Positives zu verringern, können Signaturen von gutartigen Dateien erstellt werden. Dabei sollte darauf geachtet werden, auf welcher Basis die Signatur erstellt wird, da bspw. gutartige Bytesequenzen leicht von Malware integriert und somit missbraucht werden können, wenn diese bekannt sind, während Hashes über den gesamten Dateiinhalt schwieriger zu missbrauchen sind.

3.1.2 PE-Header

Einige Arbeiten nutzen das Portable Executable (PE) Dateiformat unter der Windows-Betriebssystemfamilie für die Erkennung von Ransomware. Das PE-Format ist ein standardisiertes Format, welches notwendige Informationen enthält, damit der eingebettete Programmcode geladen

und ausgeführt werden kann (Sikorski & Honig, 2012, S. 14f). Der wohl bekannteste Dateityp, der dem PE-Format entspricht, ist die .exe, jedoch zählen auch .dll- (Programmbibliothek) oder .drv-Dateien (Treiberdatei) dazu.

Die Arbeiten beschränken sich damit speziell auf das Windows-Betriebssystem und Dateien, die dem PE-Format entsprechen. Obwohl es unter Linux und vielen Unix(-ähnlichen) Betriebssystemen das Executable and Linkable Format (ELF) gibt, was das vergleichbare Format ist, ist die Forschungslage in diesem Bereich bislang weniger stark vertreten, wenn es um die Erkennung von Ransomware geht.

Im Nachfolgenden wird vorerst das PE-Format näher betrachtet, da es nicht nur für diesen Abschnitt von Bedeutung ist, sondern auch hilfreich zum Verständnis der nachfolgenden Abschnitte ist. Aufgrund der komplex verschachtelten Struktur beschränkt sich die Erläuterung auf die grundlegenden Aspekte.

In Abbildung 2 ist die Struktur einer Portable Executable dargestellt, wobei sich das Format grundlegend in zwei Teilen einordnen lässt: den Header und den Sektionen. In den Sektionen befinden sich jegliche Daten bezüglich des Programmcodes. Dazu gehören unter anderem der auszuführende Programmcode (.text), globale Daten wie Variablen und Konstanten (.data, .rdata) sowie Ressourcen (.rsrc), z. B. Icons. Die Header bieten die nötigen Meta-Daten, sodass das korrekte Ausführen des Programms anhand der Sektionen ermöglicht wird und unterteilt sich in den DOS-, PE-, Optional- und Section-Header. Während der DOS-Header aus historischen Gründen zur Rückwärtskompatibilität existiert und heutzutage keine Anwendung mehr findet, enthalten der PE- und Optional-Header adressraumspezifische und weitere strukturelle Informationen, wie die Anzahl der Sektionen oder die benötigte CPU-Architektur. Für jede Sektion existiert zudem ein Section-Header, der Meta-Daten über die jeweilige Sektion angibt, beispielsweise die Größe der Sektion (Sikorski & Honig, 2012, S. 14f; Gibert et al., 2020; PE Format, 2025).

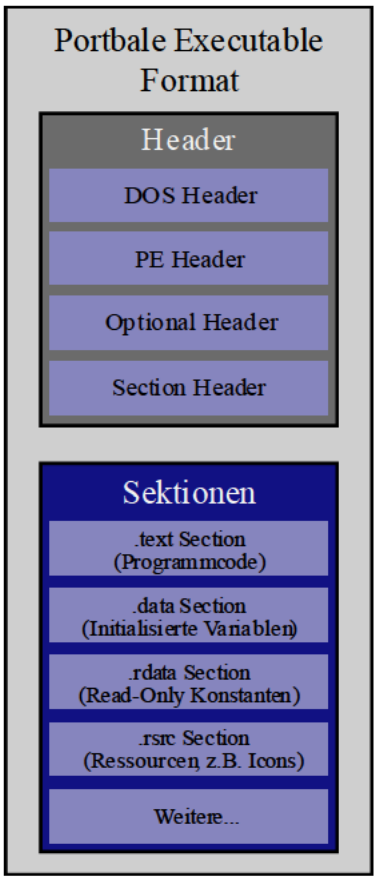

Abbildung 2: Struktur des PE-Formats

Deng et al. (2023) haben 14 wichtige Merkmale aus dem Header identifizieren können, die Unterschiede zwischen harmloser Software und Ransomware aufweisen. Diese Merkmale stammen ausschließlich aus dem Optional- und PE-Header, da die Informationen aus dem DOS- und Section-Header ineffektiv zur Erkennung seien. Auffällig war z. B., dass Ransomware keine Debugging-Informationen enthält, wodurch der Wert des Feldes „DebugSize“ 0 beträgt. Zudem nutzte Ransomware häufig den Wert 4096 für die virtuelle Adresse zur Tabelle, die die Adressen der im Programmcode verwendeten externen Funktionen angibt. Worin die Unterschiede der weiteren Eigenschaften bestehen, wurde jedoch nicht aufgeführt. In Tabelle 1 befindet sich eine Beschreibung von sieben weiteren Merkmalen, die in das Modell einbezogen wurden. Anhand von 27.118 harmlosen Programmen und 35.367 Ransomware-Samples

wurde ein Datensatz über die Merkmale erstellt und ein Machine-Learning Modell erstellt, welches ein F1-Score von 97,7 % erzielte.

Merkmal	*Beschreibung*
Machine	Gibt an, für welche CPU-Architektur die PE kompiliert wurde
MajorOSVersion	Hauptversionsnummer des benötigten Betriebssystems
MajorImageVersion	Versionsnummer der Datei (benutzerspezifischer Wert)
ResourceSize	Größe der .rsrc-Sektion
NumberOfSections	Anzahl der Sektionen
SizeOfStackReserve	Die zu reservierende Größe des Stacks beim Programmstart
ExportSize	Größe der Tabelle, die zuständig für die exportierten Funktionen des Programms zuständig ist

Tabelle 1: Beschreibung sieben weiterer Merkmale, die Deng et al. (2023) verwenden, basierend auf (PE Format, 2025)

3.1.3 Erpressungstext

Wenn ein Programm z. B. eine Ausgabe auf der Kommandozeile generiert, eine Verbindung zu einem Server aufbaut oder eine Datei in einen spezifischen Ordner kopiert, sind die dafür notwendigen Informationen in der Regel als Strings in der Binary enthalten (Sikorski & Honig, 2012, S. 11).

Der Erpressungstext der Ransomware ist häufig ebenfalls als String in der Binary integriert, weshalb sich einige Arbeiten darauf konzentriert haben diesen Text zu identifizieren. Beispielsweise entwickelten Andronio et al. (2015) einen Prototyp namens HelDroid zur Erkennung von Ransomware für das Android-Betriebssystem. Dabei haben sie unter anderem ein „Threatening Text Detector“ eingeführt, welcher auf generischer Weise den Bedrohungstext von Ransomware in einer App-Installationsdatei erkennen soll. Der Detector wurde anhand eines Machine-

Learning-Modells realisiert und anhand von (sprachlichen) Sätzen trainiert, die in den Erpresserschreiben der Angreifer vorkommen.

Alzahrani et al. (2018) verfolgen einen ähnlichen Ansatz wie HelDroid, verzichten jedoch auf den Einsatz von Machine-Learning. Stattdessen berechnen sie die Ähnlichkeit von Strings in der Binary mit vordefinierten Sätzen wie „you are hacked“ und „your files are encrypted“, die in einer Datenbank hinterlegt sind. In der Evaluation erwies sich dieses Verfahren jedoch als anfällig, da Erpressungstexte, die in einer Fremdsprache geschrieben wurden, oder Rechtschreibfehler enthielten, als harmlos klassifiziert wurden. Zudem neigten harmlose Applikationen mit einer großen Menge an Textmaterial (z. B. Bücher) dazu, fälschlicherweise als Ransomware eingestuft zu werden.

Obwohl sich die eben genannten Arbeiten auf das Android-Betriebssystem beziehen, sind diese auch auf andere Formate wie das PE-Format übertragbar (Kok S. et al., 2019), wie es die Abbildung 3 beispielhaft zeigt.

Adresse		Größe	Ty	String
Filter	Filter	Fil…	…	Filter
1400f8186	Bereich(2)['.data']	06	A	Y,~~4←
1400f8192	Bereich(2)['.data']	05	A	←T 1;
1400f9080	Bereich(2)['.data']	0b	A	Hi friends,
1400f908f	Bereich(2)['.data']	0100	A	Whatever who you are and what your title is if you're reading this it means …
1400f9190	Bereich(2)['.data']	4a	A	, we have taken a great amount of your corporate data prior to encryption.
1400f91de	Bereich(2)['.data']	da	A	Well, for now let's keep all the tears and resentment to ourselves and try t…
1400f92bc	Bereich(2)['.data']	0100	A	1. Dealing with us you will save A LOT due to we are not interested in …
1400f93bd	Bereich(2)['.data']	93	A	ber insurance, let us know and we will guide you how to properly use it. …
1400f9452	Bereich(2)['.data']	0100	A	2. Paying us you save your TIME, MONEY, EFFORTS and be back on track within …
1400f9553	Bereich(2)['.data']	b2	A	rsation. If you decide to recover on your own, keep in mind that you can …
1400f9607	Bereich(2)['.data']	0100	A	3. The security report or the exclusive first-hand information that you will…
1400f9708	Bereich(2)['.data']	35	A	into, identify backup solutions and upload your data.
1400f973f	Bereich(2)['.data']	0100	A	4. As for your data, if we fail to agree, we will try to sell personal …
1400f9840	Bereich(2)['.data']	58	A	ed in our blog - https://…
1400f989a	Bereich(2)['.data']	8b	A	5. We're more than negotiable and will definitely find the way to settle thi…
1400f9929	Bereich(2)['.data']	80	A	If you're indeed interested in our assistance and the services we provide yo…
1400f99ad	Bereich(2)['.data']	5d	A	1. Install TOR Browser to get access to our chat room - https://…
1400f9a0c	Bereich(2)['.data']	5c	A	2. Paste this link - https://…
1400f9a6a	Bereich(2)['.data']	3c	A	3. Use this code - 6729-HK-NIZN-WOPQ - to log into our chat.

Abbildung 3: Erpressungstext der Ransomware-Familie "Akira", der mit dem statischen Analysetool Detect-It-Easy (DiE) (horsicq, 2025) ausgelesen wurde

Wenn es um die Erkennung eines Erpressertextes geht, sei hier anzumerken, dass dieser nicht unbedingt hartkodiert in der Binary der Ransomware vorkommen muss, sondern auch zur Laufzeit über den C&C-Server übertragen werden kann, wodurch eine solche Analyse erfolglos wäre. Darüber hinaus verwenden Angreifer häufig Techniken zur Verschleierung der Strings, um weniger aufzufallen, wobei Tools wie FLOSS (FLARE Obfuscated String Solver) dabei helfen können, die tatsächlichen Strings zu erschließen (Shaukat & Ribeiro, 2018).

3.1.4 Opcode

Opcode ist die Kurzform für Operation Code und stellt einen Maschinenbefehl dar, der eine konkrete Anweisung an die CPU repräsentiert (Sikorski & Honig, 2012, S. 70). Letztendlich setzen sich Programme aus einzelnen Maschinenbefehle zusammen, sodass das Verhalten eines Programms durch das Analysieren der Opcodes nachvollzogen werden kann und wodurch auch gleichzeitig das Potenzial zur Erkennung von Schadsoftware besteht.

Die Detektion anhand der Opcodes wird überwiegend durch den Einsatz von Machine Learning realisiert, da die Modelle komplexe Muster in den riesigen Datenmengen erkennen können. Im Vergleich zur signaturbasierten Detektion kann das Modell anstelle von einer festen Signatur das zugrundeliegende Verhalten erlernen, sodass auch bislang unbekannte Ransomwares erkannt werden können.

Es sind verschiedene Ansätze bekannt, wie die aus den Opcodes extrahierten Informationen zur Unterscheidung zwischen Ransomware und gutartiger Software eingesetzt werden können. Einige Arbeiten konzentrieren sich beispielsweise auf die Sequenz von Opcodes oder die Opcode-Density (Oz et al., 2022). Die Operanden, mit denen ein Opcode rechnet, sind in den meisten Arbeiten nicht von Interesse.

Die Arbeit von Baldwin und Dehghantanha (2018) legt den Fokus auf die Berechnung der Opcode-Density, bei der die Häufigkeit je Opcode in einem Ransomware-Sample erfasst und anhand dieser Informationen das Modell trainiert wird. Die Autoren geben an, einen F1-Score von 100 % erzielt zu haben.

Zhang et al. (2019) haben ein ML-Modell auf Basis einer 3-Gramm Opcode-Sequenz trainiert, welches einen F1-Score von 87,3 % erzielt hat, basierend auf der Evaluation von 92 legitimen Programmen und 1.521 Ransomware-Samples. Unter N-Gramm versteht man die Zerlegung eines Textes – in diesem Fall Opcodes – in N aufeinanderfolgende Elemente, was dazu beiträgt, mehr Kontextinformationen zu berücksichtigen. Abbildung 4 verdeutlicht den Prozess der Extraktion von Opcodes und der Zerlegung am Beispiel einer 3-Gramm Opcode Sequenz.

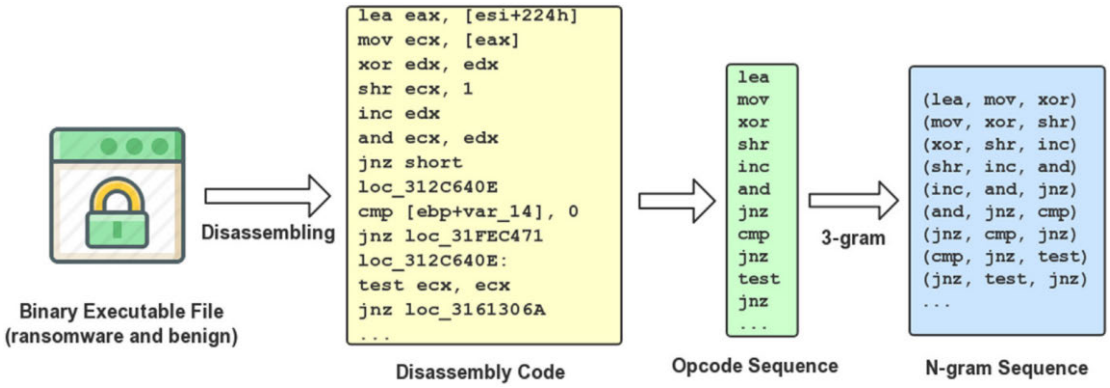

Abbildung 4: Von der Extraktion der Opcodes bis zur 3-Gramm Opcode-Sequenz
Quelle: (Zhang et al., 2019)

Auch Cesario et al. (2024) und Dolesi et al. (2024) entwickelten ein auf Opcodes gestütztes ML-Modell und behaupten, dass im Gegensatz zu anderen statischen Detektionsmechanismen die Detektion auf Basis von Opcodes weniger anfällig für Verschleierungsmethoden ist, die Ransomware bzw. Malware einsetzt, um ihr bösartiges Verhalten zu verbergen. Zur Begründung wird angeführt, dass selbst bei dem Einsatz einer Verschleierungstechnik Opcodes verwendet werden, um die bösartigen Prozesse auszuführen. Jedoch wurden in den Arbeiten keine Tests mit verschleierter Ransomware durchgeführt, weshalb die Aussage zu untersuchen ist.

3.1.5 API-Calls

Jedes Programm, welches auf einem Betriebssystem ausgeführt wird, muss mit diesem in der Regel mithilfe einer API interagieren, damit grundlegende Funktionen wie Netzwerk-, Dateisystem- oder Speicherzugriffsoperationen verwendet werden können. Für den Anwendungsentwickler wird z. B. unter Windows die Win32-API mit Funktionen wie CreateProcess, CreateFile, ReadFile und WriteFile zur Verfügung gestellt. Linux und macOS basieren auf der standardisierten POSIX-Schnittstelle und verwenden z. B. die Funktion read zum Lesen und write zum Schreiben einer Datei (Tanenbaum & Bos, 2016, S. 84; 88; 96f). Die nachfolgenden Ausführungen konzentrieren sich auf die API-Calls von Windows, da diese in der wissenschaftlichen Literatur in Bezug auf die Erkennung von Ransomware weiterverbreitet sind.

Wenn im Programmcode ein API-Call einer Windows-Programmbibliothek erfolgt, wie etwa ein Aufruf der kryptographischen Bibliothek, die häufig von Ransomware verwendet wird (Medhat et al., 2018), wird der Funktionsaufruf sowie die dazugehörige Library in der Portable Executable vermerkt. Im Detail wird dies in einer Datenstruktur in der .idata- oder .rdata-Sektion organisiert (siehe Abschnitt 3.1.2). Da die API-Calls das Verhalten eines Programms widerspiegeln, ist es auf dieser Weise möglich, Crypto-Ransomware zu identifizieren (Al-Rimy et al., 2019).

Die Arbeit von Sheen und Yadav (2018) erzielte einen F1-Score von 98,5 %, indem die Frequenz der vorkommenden API-Calls zum Training verschiedener Machine-Learning-Modelle verwendet wurde. Dabei wurde berücksichtigt, dass nur relevante API-Calls einbezogen wurden, was diejenigen sind, die mindestens 10 % häufiger in Ransomware auftreten als in legitimer Software. Insgesamt waren davon 160 API-Calls betroffen, darunter wurden beispielsweise Aufrufe wie SHEmptyRecycleBinA, SHGetFileInfo, EncryptFileA und OpenEncryptedFileRawA erfasst. Der Trainings- und Testdatensatz basiert auf 16.243 Ransomware-Samples und 3.620 gutartigen Programmen, wobei unklar ist, inwiefern gepackte oder obfuskierte Ransomware-Samples behandelt wurden.

Im Gegensatz zur zuvor erwähnten Arbeit, beschäftigt sich Medhat et al. (2018) nicht mit der Häufigkeit, sondern mit dem Vorhandensein von API-Calls anhand definierter Regeln im YARA-Framework[1]. Dabei wurde insbesondere auf das Vorhandensein der Ver- und Entschlüsselungsfunktionen der kryptographischen Bibliothek von Microsoft sowie die Funktionen FindFirstFile und FindNextFile geachtet, die Ransomware für die Suche nach spezifischen Dateien verwendet. Diese Merkmale allein erwiesen sich jedoch anfällig für False Positives, weshalb die Autoren zusätzliche Indikatoren aus anderen Detektionsansätzen einbauten.

3.2 Detektion auf Basis der dynamischen Analyse

Die dynamische Analyse zielt darauf ab, ein potenziell schädliches Programm in einer sicheren Umgebung auszuführen und das Verhalten sowie die Auswirkungen auf das System während

[1] Ein Framework des Unternehmens VirusTotal, mit dem Malware-Samples auf Basis von definierten Regeln identifiziert werden können. (VirusTotal, 2025)

der Laufzeit zu analysieren. Dabei können verschiedene Eigenschaften wie Netzwerkaktivitäten, Dateisystemaktivitäten und aufgerufene API-Funktionen für die Detektion von Interesse sein.

3.2.1 Mustererkennung von I/O-Dateioperationen

Ahmed et al. (2021) integrierten in ihrem Detektionssystem namens „Peeler“ eine Mustererkennung der I/O-Zugriffe (Input/Output-Zugriffe) auf das Dateisystem, um typische Muster einer Verschlüsselung durch Ransomware zu identifizieren. Die Autoren geben vier generische Muster bekannt:

(1) **Memory-to-File with Post-Overwrite:** Eine Datei wird gelesen, die Bytesequenz verschlüsselt und die originale Datei überschrieben, gefolgt von einer Dateiumbenennung.
(2) **Memory-to-File with Pre-Overwrite:** Wie Punkt 1, mit dem Unterschied, dass die Datei vor der Verschlüsselung und Überschreibung umbenannt wird.
(3) **File-to-File with Delete:** Die originale Datei wird gelesen, die Bytesequenz verschlüsselt und anschließend in eine neu erstellte Datei geschrieben. Anschließend wird die originale Datei gelöscht.
(4) **File-to-File with Rename and Delete:** Wie Punkt 3, mit dem Unterschied, dass bevor die originale Datei gelöscht wird, die neu erstellte Datei umbenannt wird. Das Muster zeigt sich beispielsweise bei der Ransomware-Familie WannaCry, bei der die neu erstellte Datei zunächst mit der Dateiendung .WNCRYT versehen wird und nach der Verschlüsselung die Endung .WNCRY erhält.

Die Evaluation hat einen F1-Score von 97,53 % ergeben unter der Berücksichtigung von Anwendungen, die ein ähnliches Muster wie Crypto-Ransomwares erzeugen können, wie Datenvernichter-, Archivierungs- und Verschlüsselungsprogramme.

3.2.2 Entropie verschlüsselter Dateien

In der Informationstheorie wird die Entropie als ein Maß des Informationsgehalts beschrieben (Jung & Won, 2018). Im Rahmen der Detektion von Crypto-Ransomware bezeichnen andere die Entropie als einen Wert zur Bestimmung der Zufälligkeit von Daten (Beaman et al., 2021). Dieses Maß wird verwendet, um zu erkennen, ob eine Datei verschlüsselt ist, da verschlüsselte

Daten üblicherweise zufällig sind und damit die Eigenschaft einer hohen Entropie besitzen. Zur Berechnung der Entropie existieren verschiedene Methoden, darunter die bekannte Formel des Mathematikers Claude Shannon (Davies et al., 2022a).

Insbesondere die alleinige Anwendung der Entropie, um Crypto-Ransomware zu erkennen, weist mehrere Herausforderungen auf. Zum einen kann nicht zwischen einer gutartigen und bösartigen Verschlüsselung unterschieden werden, wodurch legitime Software als Ransomware eingestuft werden könnte. Dasselbe trifft für komprimierte Daten zu, denn auch diese besitzen eine hohe Entropie und sind nur schwierig von verschlüsselten Daten zu unterscheiden. Eine weitere Herausforderung stellt die Umgehung einer hohen Entropie dar, indem die Ransomware beispielsweise Dateien nur teil- bzw. abschnittsweise verschlüsselt oder die Verschlüsselung einem Encoding (z. B. Base64) folgt (Davies et al., 2022a).

Davies et al. (2022a) haben 53 Methoden zur Berechnung der Entropie untersucht, um zu prüfen, ob eine Unterscheidung zwischen gutartigen Dateien, darunter auch komprimierte Dateien, und Dateien, die von Ransomware verschlüsselt wurden, möglich ist. Die Untersuchung erfolgte anhand von 42 gebräuchlichen Dateitypen, wobei jeder Dateityp 5.000 Dateien umfasste. Unter den mathematischen Formeln hat sich gezeigt, dass die Chi-Square-Formel mit einem F1-Score von 79 % am besten Abschnitt und neun von zwölf Ransomwares anhand der verschlüsselten Dateien erkennen konnte. Von 25 Dateitypen, die typischerweise einen hohen Entropiewert aufweisen, wurden zwei fälschlicherweise als Ransomware erkannt, nämlich das .rar und .webp Format. Die Entropie als einziges Merkmal zur Erkennung von Crypto-Ransomware einzusetzen ist damit nicht optimal, wie die Autoren beschreiben, jedoch kann es ein wichtiger Indikator zusammen mit anderen Merkmalen sein.

3.2.3 Veränderung des Dateityps

Scaife et al. (2016) entwickelten einen hybriden Prototyp, bei dem mehrere Indikatoren zum Einsatz kommen, darunter die Entropie und die Analyse der Veränderung des Typs einer Datei. Dateien entsprechen üblicherweise einem bestimmten Typ, der in diesem Kontext allerdings nicht mit einer Dateiendung wie .pdf zu verwechseln ist, sondern eine charakteristische Bytefolge (Signatur) im Dateiinhalt meint. Die Autoren geben an, dass sich der Typ einer Datei normalerweise nicht über die Lebensdauer ändert, was zur Erkennung von Crypto-

Ransomware genutzt werden kann, da die Verschlüsselung einer Datei einen Einfluss auf dessen Typ hat. Inwiefern sich der Dateityp ändert, wurde nicht näher ausgeführt.

Die Autoren testeten die Indikatoren anhand von 492 Ransomware-Samples, wobei eine Ransomware erst als erkannt gilt, wenn alle Indikatoren – einschließlich der Dateitypenänderung – auffällig sind. Die Auswertung ergab, dass 93 % der Samples erkannt werden konnten.

3.2.4 Decoys

Im Rahmen der Ransomware-Detektion sollen Decoys Ransomware zielgerichtet täuschen, um diese zu detektieren oder weitere Informationen zur Analyse zu erhalten. Dies wird dahingehend realisiert, dass Dummy-Dateien und ggf. Dummy-Ordner im Dateisystem platziert werden und der Zugriff auf diese Decoys als eine Verschlüsselung durch Ransomware interpretiert wird. Aus diesem Grund ist es wichtig, dass kein legitimer Zugriff auf die Decoys erfolgt, um eine fehlerhafte Erkennung zu verhindern (Denham & Thompson, 2023).

Decoys haben den Vorteil, dass sie eine effiziente Ressourcennutzung bieten, wenn sie z. B. mit dem Monitoring von Netzwerkaktivitäten verglichen werden. Allerdings kann nicht vorhergesagt werden, welche Dateien bzw. Decoys von einer potenziell neuen Crypto-Ransomware verschlüsselt werden (Moore, 2016).

Um die Decoys für den Benutzer transparenter zu machen, können diese im Dateisystem versteckt hinterlegt werden. Fragwürdig wäre dabei, ob die Angreifer bewusst versteckte Dateien ignorieren, um so die Decoys zu umgehen (Denham & Thompson, 2023) oder gezielt verschlüsseln, da Benutzer auf diese Weise sensible Dateien verstecken könnten. Falls die Decoys nicht versteckt werden, muss der Benutzer jedoch anhand anderer Informationen erkennen können, ob es sich um eine Decoy-Datei handelt, beispielsweise anhand des Dateinamens, um nicht (unbeabsichtigt) auf diese zuzugreifen.

Denham und Thompson (2023) analysierten verschiedene Decoy Strategien, indem sie Decoys in Form einer Decoy-Datei, eines leeren Decoy-Ordners und eines Decoy-Ordners, der eine Decoy-Datei enthält, anhand von 10 Ransomware-Samples testeten. Zudem wurde untersucht, ob das Überwachen eines Lese- oder Schreibzugriffs auf die Decoy-Datei effektiver ist. Die Decoys wurden in jedem Test im Wurzelverzeichnis des Laufwerkes, im Desktopverzeichnis

und Benutzerverzeichnis platziert. Die Ergebnisse zeigten, dass das Monitoring von Lesezugriffen auf einen Decoy-Ordner, der eine Decoy-Datei enthält, am effektivsten war. Neun Ransomware-Samples konnten erfolgreich erkannt und gestoppt werden, wodurch im Durschnitt 91,79 % weniger Dateien verschlüsselt wurden. Das Monitoring von schreibenden Zugriffen auf Decoys hingegen detektierte nur drei Ransomware-Samples.

Die o. g. Autoren machten jedoch keine Angaben über den konkreten Aufbau der Decoys hinsichtlich des Dateiinhalts, des Dateinamens, der Dateigröße oder des Dateityps, welche darüber entscheiden können, ob eine Verschlüsselung durch Ransomware erfolgt.

Einige Arbeiten konzentrieren sich darauf, Decoys so zu platzieren, dass diese möglichst als erstes verschlüsselt werden, um die Ransomware schnell zu stoppen und die Anzahl der verschlüsselten Dateien zu verringern. Zu diesem Zweck wurden die Traversierungsalgorithmen untersucht, die von Ransomware verwendet werden, um das Dateisystem zu durchlaufen, wobei vor allem die Breiten- und Tiefensuche angewendet werden (Sheen et al., 2022). Des Weiteren verschlüsseln einige Ransomwares basierend auf der Sortierung nach dem Dateityp, der Dateigröße (Ganfure et al., 2023) oder der alphabetischen Reihenfolge der Dateinamen (Genç et al., 2019).

3.2.5 Netzwerkaktivitäten

Neben den Netzwerkaktivitäten, die bei der Kommunikation mit dem C2-Server entstehen, um z. B. den Schlüssel auszutauschen, treten auch bei der Exfiltration von Daten Netzwerkaktivitäten auf (Tang et al., 2020). Derzeit mangelt es jedoch an wissenschaftlichen Studien bezüglich der Erkennung einer Datenexfiltration durch Ransomware, da der Fokus stets auf den reinen kryptographischen Ransomware-Typen liegt, wie McIntosh et al. (2024) kritisieren.

Eine einfache Möglichkeit, um einige Ransomwares zu erkennen und zu stoppen, basiert auf bekannten IP-Adressen von aktiven C2-Servern, die durch Sicherheitssysteme blockiert werden können, sodass keine Kommunikation aufgebaut werden kann. Ransomware, die zwangsweise eine Verbindung benötigt, kann an dieser Stelle keine weiteren Anweisungen empfangen und ihre Angriffskette nicht fortführen (Microsoft, 2022). Allerdings besteht die Gefahr, dass ein C2-Server eine bislang unbekannte IP-Adresse verwendet, die nicht blockiert werden kann.

Für die C2-Kommunikation verwenden die Angreifer verschlüsselte Protokolle, wodurch Detektionstechniken, die sich auf die Nutzdaten der Netzwerkpakete beziehen, nicht anwendbar sind (Cusack et al., 2018). Deshalb haben es sich Modi et al. (2020) zum Ziel gemacht, ein ML-Modell aus anderweitigen Merkmalen des Netzwerkverkehrs zu trainieren, mit dem Fokus auf HTTPS. Im Detail wurden die Netzwerkpakete zu Netzwerk-Flows aggregiert, wobei ein Flow durch mehrere Netzwerkpakete definiert wird, die die gleiche Ziel- und Quell-IP-Adresse, die gleichen Ziel- und Quell-Ports sowie dasselbe Protokoll aufweisen. Letztendlich wurden insgesamt 28 Merkmale aus drei übergeordneten Gruppen extrahiert:

(1) Verbindungseigenschaften, z. B. die Anzahl der aus- und eingehenden Pakete und Durchschnittszeit in einem Flow, Größe der Nutzdaten, diverse Standardabweichungen
(2) Verschlüsselungseigenschaften, wie das Verhältnis zwischen der Anzahl der SSL- und Nicht-SSL-Pakete sowie das Verhältnis selbstsignierter Zertifikate zu insgesamt verwendeten Zertifikaten in einem Flow
(3) Zertifikatsmerkmale, etwa Gültigkeit, Durchschnittsalter, Länge der öffentlichen Schlüssel und Vorhandensein des Common Name (CN) der verwendeten Zertifikate

Aus 666 Ransomware-Samples und 45 gutartigen Samples wurde ein Datensatz erstellt, welcher für das Training als auch die Evaluation verwendet wurde, wobei die Evaluierung ein F1-Score von 100 % erzielte.

Wie bereits im Abschnitt 2.5.4 beschrieben, kann Ransomware bei der Infizierung eines Hosts die Dateien der Netzlaufwerke verschlüsseln. Die Arbeit von Berrueta et al. (2022) hat das Ziel, den Netzwerkverkehr des NFS[2]- und SMB-Protokolls zwischen den Hosts und einem Dateiserver auf Anzeichen einer Verschlüsselung zu analysieren. Die Autoren zeichneten den Netzwerkverkehr auf, bei denen die Hosts durch gutartige Programme auf die Dateien zugreifen und bei denen die Ransomware die geteilten Dateien verschlüsselt. Aus den Netzwerkpaketen konnten Informationen gewonnen werden, etwa ob es sich um einen Lese- oder Schreibzugriff auf eine Datei handelt, eine Dateiumbenennung erfolgt oder eine Datei gelöscht wird. Insbesondere letztere beide trugen zum Erfolg der Arbeit bei, da Ransomware innerhalb kurzer Zeit

[2] Network File System

Dateien umbenennt oder die originale (unverschlüsselte) Datei löscht. Anhand dieser Merkmale wurde ein Datensatz erstellt und verschiedene ML-Modelle trainiert, wobei das Erfolgreichste einen F1-Score von über 99 % erzielte.

Netzwerkaktivitäten können zwar das Potenzial dazu haben, noch vor der Verschlüsselung die Ransomware zu erkennen und Maßnahmen zu ergreifen (Modi et al., 2020), jedoch benötigen nicht alle Ransomwares eine Verbindung zum C2-Server, weshalb nicht ausschließlich auf diese Detektionstechnik zurückgegriffen werden sollte. Zudem sind die genannten Ansätze stark abhängig von dem eingesetzten Protokoll, das zwischen den Ransomwares variieren kann.

3.2.6 Automatically Generated Domains

Im Abschnitt 2.5.3 wurde bereits der Einsatz von Automatically Generated Domains (AGD) durch Ransomware beschrieben. Die Algorithmen, die die AGDs erzeugen, werden als Domain Generation Algorithm (DGA) bezeichnet und basieren auf pseudozufälligen Algorithmen (Cebere et al., 2024).

Salehi et al. (2018) präsentieren einen Ansatz zur Erkennung von DGA-basierter Ransomware, bevor die Verschlüsselung eingeleitet wird. Typischerweise erzeugen DGAs Domainnamen mit zufälligen Zeichenfolgen (z. B. vopzzhonlnxeytmzqcobwx.com). Diese Zufälligkeit stellt das erste Erkennungsmerkmal der Arbeit dar, wobei die Autoren darauf hinweisen, dass auch legitime Domains diese Eigenschaft besitzen können. Deshalb wurden weitere Merkmale berücksichtigt, wie die Frequenz neu generierter Domains innerhalb eines bestimmten Zeitraums, da DGA-basierte Ransomware oft viele AGDs erzeugt. Ein weiteres Merkmal ist die Wiederholung bereits zuvor aufgetretener Domains, weil einige Ransomware-Varianten bestimmte Adressen erneut erzeugen. Eine Evaluation anhand von 26 Ransomware-Samples ergab, dass 14 davon auf einem DGA basierten und erfolgreich erkannt wurden. Die Autoren behaupten eine False-Positive-Rate von 0 % erzielt zu haben, ohne jedoch die entsprechenden Tests zu beschreiben, die dieses Ergebnis stützen.

Selbst wenn AGDs vollständig von regulären Domains unterscheidbar wären, müssen diese nicht ausschließlich von Malware oder zu anderweitigen bösartigen Zwecken verwendet werden. Zum Beispiel finden AGDs in Benchmark Tools für das Domain Name System (DNS)

Verwendung, was zu False Positives führen könnte (Cebere et al., 2024). Ein weiterer Nachteil ist, dass diese Detektionstechnik nur Ransomwares einschließt, die einen Domain Generation Algorithm implementiert und anwendet (Modi et al., 2020).

3.2.7 API- und System-Calls

Die Möglichkeit, Crypto-Ransomware anhand der API-Calls zu identifizieren, wurde bereits im Rahmen der statischen Analyse behandelt (siehe Abschnitt 3.1.5). Darauf aufbauend werden in diesem Abschnitt Unterschiede und weiterführende Erkenntnisse im Kontext der dynamischen Analyse beleuchtet.

Das Extrahieren von API-Calls im statischen Analyseansatz kann durch einfache Verschleierungstechniken, wie etwa die Obfuskation des Programmcodes, verhindert werden. Aus diesem Grund fokussieren sich einige Arbeiten auf die Detektion von API-Calls im Rahmen der dynamischen Analyse, bei der die API-Calls zur Laufzeit einer Ransomware erfasst und analysiert werden.

Ahmed et al. (2020) legen den Fokus auf System-Calls, eine Art von API-Call, der einen Funktionsaufruf auf der niedrigsten Ebene eines Betriebssystems beschreibt. Dabei handelt es sich um einen Aufruf, der einen Service im Kernel des Betriebssystems verwendet, um eine privilegierte Operation auszuführen, wie bspw. Datei- und Netzwerkoperationen oder der Zugriff auf Hardwareressourcen (Tanenbaum & Bos, 2016, S. 96f). Die von den Autoren vorgestellte Methode ist auf das Windows-Betriebssystem ausgelegt und adressiert das Problem, dass Ransomware häufig irrelevante und redundante System-Calls zur Laufzeit durchführt, die zu einer Beeinträchtigung der Detektionsmechanismen führt. Dazu wird ein zweistufiger Prozess angewendet. Im ersten Schritt werden System-Calls gefiltert, die keine starke Indikation für das kritische Verhalten der Ransomware liefern, während im zweiten Schritt eine spezielle ML-Technik verwendet wird, die relevante Merkmale identifiziert und redundante entfernt. Die System-Calls-Sequenzen wurden dann in eine N-Gram Repräsentation transformiert und ein ML-Modell mithilfe von 1.354 Ransomware-Samples und 1.358 gutartigen Samples trainiert, bei dem ein F1-Score von 98,6 % erreicht werden konnte.

3.3 Limitierungen der Detektionstechniken

Bei der Beschreibung der einzelnen Detektionsansätze wurde bereits teilweise auf spezifische Limitierungen und Umgehungsstrategien hingewiesen. In der Literatur wird des Weiteren immer wieder auf Herausforderungen der statischen und dynamischen Analyse sowie der Machine-Learning-Modelle aufmerksam gemacht, die in diesem Abschnitt auf die Detektionsansätze übertragen werden.

3.3.1 Statische Analyse

Die Detektionstechniken auf Basis der statischen Analyse haben eine Reihe von Herausforderungen zu überwinden, denn die Ersteller von Malware setzen auf diverse Verschleierungsmethoden, um eine Erkennung zu erschweren.

Zur Verschleierung setzen die Angreifer beispielsweise auf metamorphe, polymorphe und oligomorphe Techniken. Metamorphe Malware modifiziert ihren Programmcode bei jeder Replikation, sodass das ursprüngliche (schädliche) Programmverhalten beibehalten wird. Polymorphe und oligomorphe Malware verschlüsseln den Programmcode und entschlüsseln ihn zur Laufzeit mithilfe einer Entschlüsselungsroutine. Der Unterschied beider Techniken liegt darin, dass die oligomorphe Technik nur über eine endliche Menge an Entschlüsselungsroutinen besitzt, während die polymorphe Technik über eine potenziell unendliche Menge verfügt. Dies ist deshalb relevant, da auch die Entschlüsselungsroutinen zur Erkennung von Malware adressiert werden können (Malik et al., 2022, S. 45-47).

Des Weiteren kann zur Verschleierung ein Packer eingesetzt werden, welcher den Schadcode komprimiert und erst zur Laufzeit wieder dekomprimiert, was dazu führt, dass die Einsicht und die Analyse des Codes nicht mehr möglich sind (Zalesskiy, 2024). Auch hier lässt sich prinzipiell über die Routine des Entpackens eine Signatur erstellen, allerdings kann die Routine metamorph aufgebaut und somit bei jeder Replikation unterschiedlich sein.

Eine weitere Herausforderung ist, dass die statische Analyse lediglich auf Dateien abzielt und nicht gegen Angriffe schützt, die den schädlichen Code direkt in dem Arbeitsspeicher ausführen, wie es bei Fileless-Ransomware der Fall ist. Diese Gruppe nutzt native Skriptsprachen wie JavaScript, PHP und PowerShell, um den Programmcode auszuführen (Ganfure et al., 2023).

Darunter können z. B. auch Ransomware-Angriffe fallen, bei denen die Angreifer durch einen Buffer-Overflow bereits Zugang zum System erlangen konnten.

Darüber hinaus funktionieren einige statische Detektionstechniken, wie Opcodes, nur bei einem spezifischen Format wie dem PE-Format, wodurch beispielsweise schädliche Skriptdateien nicht analysiert werden können.

Herausforderungen / **Detektionstechnik**	**Packer**	**Meta-, poly- und oligomorphe Techniken**	**Fileless-Malware**
Signaturbasierte Detektion	○	X	X
PE-Header	✓	✓	X
Erpressungstext	X	X	X
Opcode	○	○	X
API-Calls	X	X	X

Tabelle 2: Übersicht der Herausforderungen der Detektionstechniken auf Basis der statischen Analyse (X: Einschränkung vorhanden, ○: Einschränkung teilweise vorhanden, ✓: Einschränkung nicht vorhanden)

Die Tabelle 2 stellt die Detektionstechniken aus Abschnitt 3.1 mit den oben genannten Herausforderungen anhand von drei Kategorien gegenüber. Einige Kategorisierungen sind nicht selbstverständlich und werden daher näher ausgeführt. In gepackter Malware ist es beispielsweise möglich, dass eine Signatur über den eingesetzten Packer erstellt wird und die Malware dadurch erkannt wird. Allerdings werden Packer auch für gutartige Softwares eingesetzt, sodass eine Signatur über den Packer keine Unterscheidung zwischen schädlicher und gutartiger Software erlaubt, wenn Malware legitime Packer verwendet. Wie bereits erwähnt erschwert der Einsatz von metamorphen Packern ebenfalls eine Erkennung.

Der PE-Header ist eher unempfindlich gegenüber Verschleierungstechniken, wie es die Arbeit von Rezaei et al. (2021) zeigt. Die Autoren haben in ihrer Arbeit zur Detektion von verschiedenen Malware-Typen den PE-Header genutzt und in der Evaluationsphase zusätzliche Tests bezüglich metamorpher und gepackter Malware durchgeführt. Dabei hat sich gezeigt, dass der F1-Score lediglich einen Prozentpunkt schlechter ausfällt als bei nicht verschleierter Malware.

Die Wirksamkeit von Verschleierungstechniken bezüglich der Opcode-Detektionstechnik ergibt sich aus den Aussagen von Cesario et al. (2024) und Dolesi et al. (2024), die behaupten, dass diese weniger anfällig für verschleierte Ransomware ist.

Da bei der statischen Analyse eine Datei zur Untersuchung herangezogen wird, sind alle Detektionstechniken auf Basis der statischen Analyse nicht gegen Fileless-Malwares wirksam.

3.3.2 Dynamische Analyse

Aufgrund der Hürden der statischen Analyse wird oftmals die dynamische Analyse herangezogen, wobei auch diese ihre Vor- und Nachteile hat. Wenn Malware dynamisch analysiert wird, werden oftmals isolierte Umgebungen wie virtuelle Maschinen oder Sandboxes verwendet. Einige Malwares suchen gezielt nach Hinweisen auf eine virtuelle Umgebung, um ihr Laufzeitverhalten gezielt anzupassen (Liska & Gallo, 2016, S. 9). Im Kontext von Ransomware könnte dies bedeuten, dass der Verschlüsselungsprozess erst nach einer bestimmten Zeit oder gar nicht erfolgt, was dazu führen könnte, dass diese als nicht bösartig eingestuft wird.

Zur dynamischen Analyse von Malware können unter anderem Debugger-Tools verwendet werden, um den Code zu untersuchen. Allerdings setzt Malware häufig Anti-Debugging-Techniken ein, indem z. B. der normale Programmablauf manipuliert wird, was die Codeanalyse erschwert (Sikorski & Honig, 2012, S. 351).

Nicht vernachlässigt werden sollte zudem, dass eine dynamische Analyse in der Regel mit einem höheren Ressourcenverbrauch verbunden ist, sei es durch das Monitoring von Merkmalen zur Laufzeit (Medhat et al., 2018) oder der Verwendung einer virtuellen Umgebung.

Herausforderungen / Detektionstechnik	Evasion vor einer virtuellen Umgebungen	Anti-Debugging-Techniken	Ressourcenverbrauch
Dateisystemaktivitäten	X	✓	○
Netzwerkaktivitäten	X	✓	○
Algorithmically Generated Domains	X	✓	✓
API-/System-Calls	○	○	○

Tabelle 3: Übersicht der Herausforderungen der Detektionstechniken auf Basis der dynamischen Analyse (X: Einschränkung vorhanden, ○: Einschränkung teilweise vorhanden, ✓: Einschränkung nicht vorhanden)

Ähnlich wie in Tabelle 2 werden in Tabelle 3 die Detektionstechniken anhand der in diesem Abschnitt aufgeführten Herausforderungen gegenübergestellt. Techniken, die sich auf das Dateisystem beziehen, wie beispielsweise die Entropieanalyse oder das I/O-Zugriffsmuster, sind unter dem Begriff „Dateisystemaktivitäten“ zusammengefasst.

Die Evasion vor einer virtuellen Umgebung hat zur Folge, dass beispielsweise weder Netzwerk- noch Verschlüsselungsaktivitäten im Dateisystem auftreten, weshalb die meisten Techniken ineffektiv sind. API- bzw. System-Calls weisen diesbezüglich teilweise Einschränkungen auf, wie es die Arbeit von Coglio et al. (2023) zeigt. Die Autoren konzentrierten sich unter anderem darauf Ransomware anhand der Funktionsaufrufe zu detektieren, die bei dem Versuch entstehen, eine virtuelle Umgebung zu erkennen. Die Evaluationsergebnisse der Arbeit deuten mit einem F1-Score von 81 % jedoch darauf hin, nicht zuverlässig genug zu sein.

Des Weiteren können die API-Calls eingeschränkt werden, etwa wenn Debugging Tools zur Extraktion der API-Calls verwendet werden. Jedoch können effektivere Methoden zum Monitoring eingesetzt werden, die das Problem umgehen, beispielsweise durch das Abfangen von API-Calls mittels API-Hooking. Dabei wird der Funktionsaufruf unterbrochen und zusätzlicher

Code eingeschleust, sodass das ursprüngliche Verhalten des Funktionsaufrufs manipuliert wird (Kolodenker et al., 2017).

Insbesondere wenn die Techniken während des laufenden Betriebs direkt auf einem Host zum Einsatz kommen, ist der Ressourcenverbrauch ein wichtiges Kriterium, der jedoch stark von der Hostumgebung abhängig ist. Beispielsweise finden auf einigen Systemen häufig Aktivitäten am Dateisystem oder dem Netzwerk statt, wodurch der Rechenaufwand steigt, da mehr Daten analysiert werden müssten. In solch einem Fall kann beispielsweise die Berechnung einer Entropie über die gesamte Datei als Erkennungsmerkmal zu aufwendig sein.

3.3.3 Machine-Learning-Modelle

Viele der Detektionstechniken auf Basis eines ML-Modells verzichten auf die Evaluierung ihres Modells in einer praktischen Umgebung. Stattdessen werden die zur Evaluierung eingesetzten Datensätze in einen Trainings- und Testdatensatz aufgeteilt und das Modell anhand des Testdatensatzes evaluiert. Dabei bleibt unklar, wie gut sich das Modell in der Praxis verhält, beispielsweise bezüglich des Ressourcenverbrauchs oder der Dauer bis zur Erkennung von Ransomware, wenn die Daten abweichen.

In der Theorie können außerdem sogenannte Adversarial Attacken auf ML-Modelle erfolgen. Bei dieser Art von Angriff wird der Input in das ML-Modell minimal verändert, sodass eine fehlerhafte Klassifikation erfolgt, die dem Angreifer einen Vorteil verschafft. In Bezug auf die Detektion von Malware bedeutet dies, dass die Schadsoftware so modifiziert werden könnte, dass sie von dem ML-Modell als harmlos eingestuft wird. Das Durchführen einer Adversarial Attacke gestaltet sich jedoch als schwierig, da die Angreifer normalerweise keine Informationen über den Aufbau des ML-Modells besitzen, wodurch sie einem Trial-and-Error-Verfahren folgen müssten (Aryal et al., 2024).

Für die in Abschnitt 3.1 und 3.2 beschriebenen Detektionstechniken, die ein ML-Modell einsetzen, ist eine Adversarial Attacke theoretisch möglich, wenn keine entgegenwirkenden Maßnahmen bei der Erstellung eines ML-Modells getroffen wurden. Als effektive Gegenmaßnahme hat sich beispielsweise das Adversarial Training ergeben, bei dem leicht modifizierte Daten zum Trainingsdatensatz hinzugefügt werden, die dem Input einer Adversarial Attacke ähneln (Shafahi et al., 2019).

4 Entwicklung eines hostbasierten File-Integrity-Monitors zur Erkennung einer Verschlüsselung durch Ransomware

IT-Systeme sind der potenziellen Gefahr ausgesetzt, dass Angreifer in diese eindringen und einen Crypto-Ransomware-Angriff starten, der weitreichende Schäden verursacht (siehe Abschnitt 2.4). Präventive Sicherheitsvorkehrungen können zwar das Risiko einer Infizierung minimieren, schließen diese jedoch nicht aus (Gómez-Hernández et al., 2018). Daher ist das Ziel dieses Kapitels, einen hostbasierten File-Integrity-Monitor (FIM) für das Windows-Betriebssystem zu entwickeln, der in der Lage ist, ein mit Crypto-Ransomware infiziertes System in der Phase der Verschlüsselung zu erkennen. Bei einer erkannten Verschlüsselung soll eine Alarmierung erfolgen, sodass ggf. frühzeitig Maßnahmen ergriffen werden können, um die Schäden zu begrenzen. Die entwickelte Anwendung wird im darauffolgenden Kapitel evaluiert und mit echten Ransomware-Samples getestet.

4.1 Anwendungsfälle eines File-Integrity-Monitors

Ein File-Integrity-Monitor ist ein Werkzeug, das in der IT-Sicherheit vielfältige Anwendungsfällen abdecken kann:

(1) **Schutz der Integrität kritischer Dateien:** Daten können vor einem Verlust oder der Manipulation geschützt werden, indem z. B. die Dateiänderungen autorisiert sein müssen (Enginsight, o. D.).
(2) **Erkennung von Cyber-Angriffen:** Durch ungewöhnliche Änderungen an Dateien können Cyber-Angriffe erkannt und zuständige Sicherheitsexperten alarmiert werden, um auf die Bedrohung zu reagieren und ein größerer Schaden abzuwenden (Imtiaz, 2024).
(3) **Einhaltung von Compliance-Anforderungen:** Ein FIM kann dabei helfen Compliance-Anforderungen zu erfüllen, was Standards, Vorschriften oder gesetzliche Regularien sind, die Unternehmen einhalten müssen (Imtiaz, 2024).

(4) **Aufdeckung von Schwächen in der IT-Infrastruktur:** Ein FIM kann präventiv sicherheitsrelevante Schwachstellen identifizieren und melden, z. B. wenn der Administrator (versehentlich) die Zugriffsrechte kritischer Dateien verändert hat (Imtiaz, 2024).

Den Rahmen des zu entwickelnden FIM wird durch den Anwendungsfall aus Punkt 2 definiert, allerdings ist dieser nicht für jegliche Cyber-Angriffe vorgesehen, sondern speziell für die Erkennung von Crypto-Ransomware-Angriffen.

4.2 Funktionsweise eines File-Integrity-Monitors

Ein File-Integrity-Monitor überwacht sensible Dateien auf unbefugte Integritätsverletzungen, die von nicht autorisierten Nutzern oder Cyberangriffen hervorgerufen werden und meldet diese Aktivitäten gegebenenfalls. Die Überwachung findet hierbei gezielt auf verschiedene kritische Verzeichnisse oder Dateien statt, die dafür entsprechend konfiguriert wird (Imtiaz, 2024).

Im Wesentlichen verwendet ein FIM eine sogenannte Baseline, um zu entscheiden, ob eine Datei manipuliert wurde. Die Baseline beschreibt für jede zu überwachende Datei einen Referenzzustand, von dem angenommen wird, dass dieser nicht kompromittiert ist, oder er wird aus anderen Gründen als Referenz festgelegt, weil er beispielsweise einen Sollzustand einer Datei zu einem bestimmten Zeitpunkt repräsentiert. Entspricht der aktuell festgestellte Zustand einer Datei der gespeicherten Baseline, so erfolgte seit der Erstellung der Baseline keine Veränderung der Datei. Bei einer Abweichung wird die Veränderung der Datei festgestellt, woraufhin der FIM mittels definierter Regeln bewertet, ob diese Veränderung einer Integritätsverletzung entspricht (Imtiaz, 2024).

In Bezug auf den zu entwickelnden FIM werden Änderungen an überwachten Dateien durch Benutzer akzeptiert, während Änderungen in Form einer Verschlüsselung durch Crypto-Ransomware erkannt und gemeldet werden sollen.

Eine einfache technische Möglichkeit, um den Zustand einer Datei zu erfassen und diesen in der Baseline aufzunehmen, ist die Verwendung von kryptographischen Prüfsummen, wodurch nicht der gesamte Dateiinhalt abgespeichert und verglichen werden muss. Neben der

Prüfsumme können zusätzlich Meta-Daten wie die Dateigröße, das Änderungs- und Erstelldatum oder auch die Zugriffsrechte aufgenommen werden (Imtiaz, 2024).

4.3 Anforderungsanalyse

Die Anforderungen bilden die Grundlage für die Implementierung. Hierfür werden die Anforderungen in die funktionalen und nichtfunktionalen Anforderungen gegliedert.

4.3.1 Abgrenzung des Umfangs

Da auch Sicherheitswerkzeuge wie der FIM nicht von einem Angriff ausgenommen sind, ist es unentbehrlich, diesen möglichst robust zu gestalten und vor der Manipulation zu schützen, worunter beispielsweise der Schutz der Baseline zählt. Aufgrund des Umfangs der vorliegenden Arbeit wird dieser Aspekt allerdings nicht bei der Implementierung berücksichtigt.

Der Einfachheit halber beziehen sich die Anforderungen auf eine einzige Benutzerrolle: dem Systemnutzer, der mit den Funktionen des FIM arbeitet und die grundlegenden Kenntnisse der IT-Sicherheit hat. Eine Trennung zwischen einem Administrator und den Nutzern eines Hosts stellt eine sinnvolle Erweiterung für zukünftige Arbeiten dar, weil die Nutzer eines Hosts nicht zwingend über die nötigen administrativen Kenntnisse verfügen und an der Möglichkeit, den FIM zu konfigurieren, gehindert werden sollten.

4.3.2 Funktionale Anforderungen

Tabelle 4 zeigt die funktionalen Anforderungen auf, die das Verhalten und die Funktionen des File-Integrity-Monitors widerspiegeln.

Position	***Titel***	***Beschreibung***
FA-1	Überwachung von Verzeichnissen	Der Benutzer kann auswählen, welche Verzeichnisse überwacht werden sollen, wobei sich die Überwachung auf die in den ausgewählten Verzeichnissen enthaltenen Dateien und Unterverzeichnissen bezieht
FA-2	Aufzeichnung von Dateioperationen	Der File-Integrity-Monitor protokolliert Dateioperationen wie das Modifizieren, Löschen oder Hinzufügen von Dateien
FA-3	Erstellung der Baseline einleiten	Der Benutzer kann die Erstellung der Baseline über die konfigurierten Verzeichnisse einleiten und neue Verzeichnisse zur Baseline hinzufügen
FA-4	Benachrichtigung bei erkannten Ransomware-Aktivitäten	Der Benutzer soll bei der Erkennung von Crypto-Ransomware-Aktivitäten per E-Mail und einer visuellen Nachricht auf dem Host alarmiert werden
FA-5	Aktivieren und deaktivieren des Monitorings	Der Benutzer kann das Monitoring aktivieren und deaktivieren sowie den jeweiligen Status einsehen
FA-6	Visualisierung von protokollierten Dateiaktivitäten und Baseline-Inhalten	Der Benutzer kann einsehen, welche Dateien modifiziert, gelöscht, hinzugefügt oder umbenannt worden sind sowie welche Daten in der Baseline enthalten sind

Tabelle 4: Funktionale Anforderungen des File-Integrity-Monitors

4.3.3 Nichtfunktionale Anforderungen

Tabelle 5 beschreibt die nichtfunktionalen Anforderungen, die sich auf die Qualität und Leistung des Systems beziehen.

Position	***Titel***	***Beschreibung***
NFA-1	Ausführung auf dem Windows-Betriebssystem	Der FIM soll auf Windows 10 lauffähig sein
NFA-2	Stabilität des Datei- und Betriebssystems	Das Betriebs- und Dateisystem bleibt auch bei eventuell auftretenden Fehlern im File-Integrity-Monitor weiterhin stabil und funktionstüchtig
NFA-3	Zuverlässiges Logging	Das Logging von Dateioperationen soll zuverlässig sein, d. h. keine Veränderungen in einem zu überwachenden Verzeichnis auslassen
NFA-4	Unmittelbare Erfassung von Dateiänderungen	Dateiänderungen in einem zu überwachenden Verzeichnis sollen unmittelbar vom FIM erkannt werden
NFA-5	Zuverlässige Erkennung von Ransomware-Aktivitäten	Mindestens 90 % der getesteten Ransomware-Samples sollen erkannt werden

Tabelle 5: Nichtfunktionale Anforderungen des zu entwickelnden File-Integrity-Monitors

4.4 Konzept

4.4.1 Zeit- und echtzeitbasierte Überwachung von Dateiänderungen

Das Monitoring von Änderungen in einem zu überwachenden Verzeichnis kann entweder durch einen zeit- oder echtzeitbasierten Ansatz erfolgen. Bei dem zeitbasierten Ansatz werden die Dateien in zufälligen oder regelmäßigen Zeitintervallen auf potenzielle Veränderungen überprüft, während der echtzeitbasierte Ansatz hingegen die Möglichkeit einer sofortigen Reaktion bietet, wenn eine Veränderung stattfindet (Zlatkovski et al., 2018).

Die nichtfunktionale Anforderung NFA-4 erfordert eine sofortige Erfassung einer Dateiänderung, um die potenzielle Modifizierung einer Datei durch Crypto-Ransomware möglichst schnell zu registrieren. Die Erfüllung dieser Anforderung macht den Einsatz des

echtzeitbasierten Ansatzes erforderlich. Auch die Anforderung NFA-3, die ein zuverlässiges Logging verlangt, sodass keine Veränderungen unbeachtet bleiben, wird von dem echtzeitbasierten Ansatz erfüllt, da bei dem zeitbasierten Ansatz die Gefahr der Nichterfassung von Änderungen besteht, wenn die Prüfungen genau zwischen den Intervallen erfolgen (Zlatkovski et al., 2018).

4.4.2 Einsatz einer graphischen Benutzeroberfläche

Die definierten Anforderungen erfordern eine Reihe an Interaktionsprozessen mit dem Benutzer. Um diese möglichst benutzerfreundlich zu gestalten, werden die Funktionalitäten in einer graphischen Benutzeroberfläche integriert. Hierbei wird der Benutzer über diverse Schaltflächen neue Verzeichnisse hinzufügen und entfernen (FA-1), die Baseline erstellen (FA-3) und das Monitoring aktivieren bzw. deaktivieren (FA-5) können.

Bei der Erstellung einer Baseline wird berücksichtigt, dass der Benutzer eventuell bereits eine Baseline angelegt hat, weshalb dieser eine Überschreibung per Dialogfenster bestätigen muss. Außerdem wird der Fall beachtet, dass der Benutzer ein weiteres Verzeichnis überwachen möchte, welches in die Baseline hinzugefügt werden soll.

Die Visualisierung der protokollierten Dateiaktivitäten und den Inhalten der Baseline (FA-6) wird ebenfalls in der Benutzeroberfläche auf zwei voneinander getrennten Ansichten realisiert.

4.4.3 Benachrichtigung des Benutzers

Die funktionale Anforderung FA-4 sieht eine Benachrichtigung des Benutzers bei Crypto-Ransomware-Aktivitäten sowohl per E-Mail als auch einer visuellen Nachricht auf dem Hostsystem vor. Die visuelle Nachricht wird durch eine Toast-Benachrichtigung über das vom Windows bereitgestellte Benachrichtigungscenter realisiert.

Die Benachrichtigung per E-Mail wird derart umgesetzt, dass der Benutzer einen SMTP-Server und die dazugehörigen Authentifizierungsdaten in der Benutzeroberfläche eingeben kann, um E-Mails an einen beliebigen Empfänger oder sich selbst zu versenden. Die Authentifizierungsdaten werden verschlüsselt in einer Datei hinterlegt und lediglich für die Dauer des

Versandprozesses unverschlüsselt im Arbeitsspeicher gehalten, um das Risiko zu minimieren, dass Angreifer durch das Auslesen des Prozessspeichers an sensible Daten gelangen.

Im gleichen Zuge wird dem Benutzer die Möglichkeit gegeben, den Benachrichtigungstext für den Inhalt der E-Mail und der Push-Benachrichtigung zu definieren.

4.4.4 Merkmale zur Erkennung von Ransomware

Die Anforderung NFA-5 fordert eine zuverlässige Erkennung von Ransomware-Aktivitäten und setzt voraus, dass mindestens 90 % der im Rahmen der Evaluation getesteten Ransomware-Samples erkannt werden. Da aus dem Kapitel 3 bekannt ist, dass jedes Merkmal sowohl Vor- als auch Nachteile besitzt und daher unterschiedlich starke Indikatoren bilden, integriert der FIM drei verschiedene Merkmale zur Erkennung von Crypto-Ransomware auf Basis eines Bewertungsmodells, sodass weniger geeignete Merkmale einen niedrigeren und stärker korrelierende Merkmale einen höheren Einfluss haben.

Als Merkmale werden die Manipulation von Decoys, die Veränderung von Entropiewerten und die Veränderung des Dateityps implementiert, die jeweils in den Abschnitten 3.2.4, 3.2.2 und 3.2.3 näher beschrieben wurden. Die Werte der Merkmale werden in die Baseline aufgenommen, um im Anschluss mit den Werten einer modifizierten Variante der Datei verglichen und bewertet zu werden.

Entropie

Um eine potenzielle Verschlüsselung zu erkennen, wird die Entropie verwendet. Zur Berechnung dieser wurde die Chi-Square-Statistik herangezogen, da sich diese als einer der effektivsten Methoden erwiesen hat (siehe Abschnitt 3.2.2). Um den Chi-Square-Wert zu berechnen, wird zunächst die Häufigkeit für jede der 256 verschiedenen Byte-Kombinationen einer Datei gezählt und folgende Funktion angewendet:

$$x^2 = \sum \frac{(O_i - E_i)^2}{E_i} = \sum_{i=0}^{255} \frac{\left(O_i - \frac{n}{256}\right)^2}{\frac{n}{256}}$$

Variable O_i beschreibt die Häufigkeit des i-ten Bytes und E_i die erwartete Häufigkeit, welche in diesem Fall eine möglichst gleichmäßige Byteverteilung entspricht und somit als $\frac{n}{256}$ definiert werden kann, wobei n die Anzahl der Bytes bzw. die Größe der Datei repräsentiert. Der Wertebereich erstreckt sich zwischen null (gleichmäßige Byteverteilung) und $n * 255$ (ungleichmäßige Byteverteilung).

Um einen Grenzwert zur Bestimmung einer potenziell verschlüsselten Datei zu erfassen, wurde ein Test anhand des NapierOne Datensatzes von Davies et al. (2022b) durchgeführt, der über 9.000 Dateien und 43 verschiedene Dateitypen enthält, wobei jede Datei bis zu einer Größe von 154 MB variiert. Dieser Datensatz wurde verwendet, da dieser unter anderem ausdrücklich für Untersuchungen im Zusammenhang mit Ransomware konzipiert worden ist. Bei dem Test wurde jede Datei mit dem AES- und DES-Algorithmus verschlüsselt und die Entropie der unverschlüsselten sowie beider verschlüsselter Varianten berechnet. Abbildung 5 zeigt ein Histogramm der Chi-Square-Werte der Dateien, die mit dem AES-Algorithmus verschlüsselt wurden, wobei knapp 89 % der Dateien einen Wert zwischen 219 und 292 aufweisen. Die Entropiewerte der mit dem DES-Algorithmus verschlüsselten Dateien befinden sich nahezu in demselben Wertebereich. Der Grenzwert wurde auf 350 festgelegt, sodass der Entropiewert einer modifizierten Datei kleiner als dieser sein muss, um als verschlüsselt angenommen zu werden.

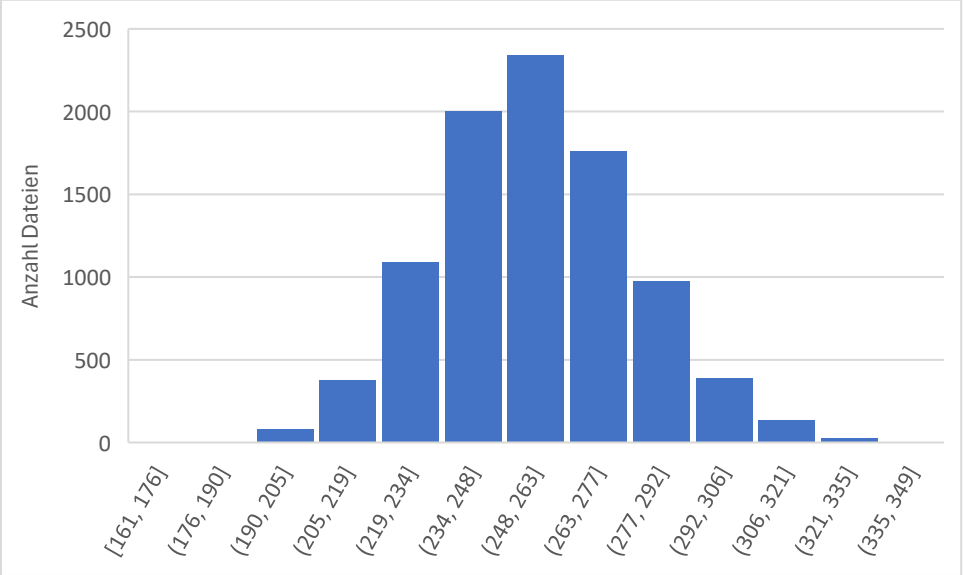

Abbildung 5: Histogramm der Chi-Square-Werte von 9191 Dateien nach der Verschlüsselung mit dem AES-Algorithmus

Neben den verschlüsselten Dateitypen weisen teilweise auch unverschlüsselte Typen, wie komprimierte Archive eine hohe Entropie[3] auf und sind daher nicht immer voneinander zu unterscheiden. In Bezug auf den NapierOne Datensatz betrifft dies rund 10 % der Dateien, dessen Chi-Square-Wert kleiner als der o. g. Grenzwert von 350 ist. Abhilfe hierfür leistet die Berechnung des Chi-Square-Wertes auf den ersten 256 Bytes, da von den 10 % ca. 83 % der Dateien einen Wert von über 400 zeigten, der somit über dem Grenzwert liegt und zur Bewertung hinzugezogen werden kann.[4] Die Abbildung 6 stellt diesen Zusammenhang dar.

[3] Unter einer hohen Entropie sind zufällige Daten gemeint. Eine hohe Entropie spiegelt einen niedrigen Chi-Square-Wert wider.

[4] Die Idee die ersten 256 Bytes zu nutzen, stammt von Davies et al. (2021), die die Differenzfläche zwischen der Entropiekurve des Dateiheaders und der Entropiekurve, die aus rein zufälligen Daten generiert wurde, verwenden, um Dateien mit einer hohen Entropie und verschlüsselten Dateien zu unterscheiden. Es war angedacht diesen Ansatz in diesem Kontext zu übertragen, jedoch scheiterte der Versuch aufgrund fehlender Implementierungsdetails.

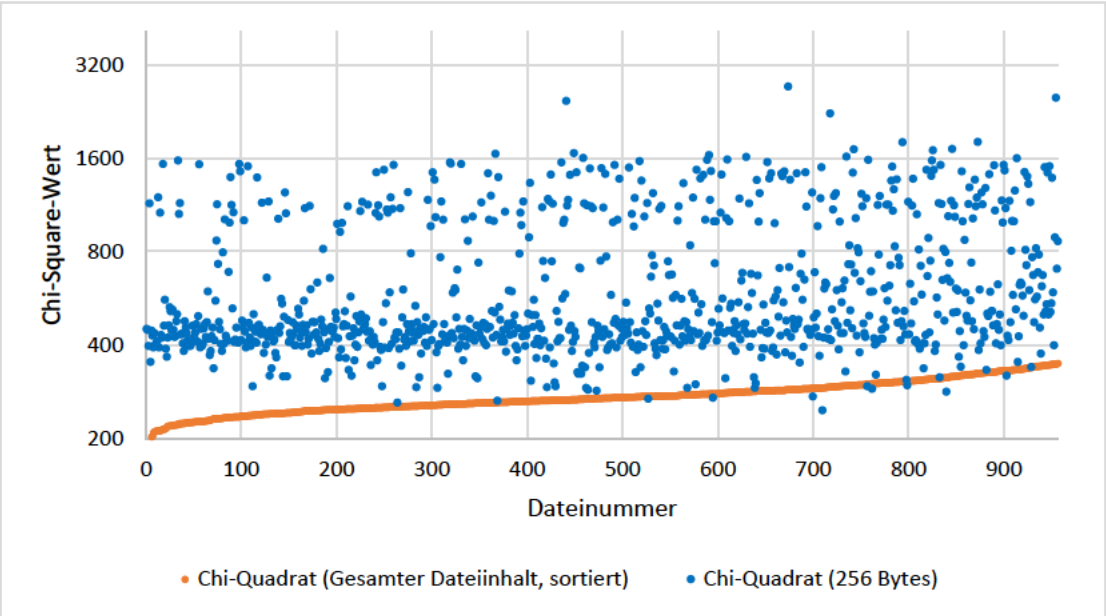

Abbildung 6: Vergleich der Chi-Square-Werte auf den gesamten Dateiinhalt und der ersten 256 Bytes für Dateien mit einem Chi-Square-Wert kleiner 350 (gesamter Inhalt)

Decoys

Auf dem Hostsystem werden statische Decoy-Dateien aus einer vorgefertigten Sammlung mit echten Benutzerdateien verteilt, die beliebig vergrößert werden kann. Die Sammlung beinhaltet zwölf Decoys aus dem NapierOne-Datensatz, dessen Inhalt minimal verändert wurde, um das Änderungsdatum zu aktualisieren und das Risiko einer Umgehung durch die Erstellung eines digitalen Fingerabdrucks seitens der Ransomware zu minimieren. Dabei wurden nicht nur Dateien mit einem Dokument- oder Medienformat berücksichtigt (z. B. .pdf, .png, .jpg, .ods, .docx), sondern ebenfalls Formate technischer Natur (z. B. Datenbankdateien und kryptographische Schlüssel), da einige Ransomwares auch an diesen Formaten interessiert sind.

Aus der Sammlung werden jeweils die größte und kleinste Datei aus den technischen und den Dokument- bzw. Medienformaten ausgewählt und in das Verzeichnis des Benutzers hinterlegt. Außerdem werden zwei Ordner erstellt, dessen Namen so gewählt sind, dass einer in der Reihenfolge als erstes und einer als letztes erscheint. Auf beiden Verzeichnissen werden die

restlichen Decoys aus der Sammlung verteilt. Eine derartige Verteilung der Decoys soll dazu führen, dass diese vor den echten Benutzerdateien verschlüsselt werden (siehe Abschnitt 3.2.4).

Zunächst werden die Decoys im Dateisystem versteckt hinterlegt, um das Risiko einer unbeabsichtigten Änderung durch den Benutzer zu minimieren. Da jedoch unklar ist, inwiefern sich die versteckten Dateien bezüglich der Verschlüsselung durch Ransomware verhalten, ist dies erstmal als eine vorläufige Lösung anzusehen und in der Evaluation auszuwerten. Während lesende Zugriffe auf den Decoys als unverdächtig angesehen werden, weil legitime Anwendungen wie Antivirenprogramme auf diese zugreifen könnten, werden lediglich modifizierende Zugriffe als verdächtig betrachtet.

Des Weiteren wird dem Benutzer in der Benutzeroberfläche die Möglichkeit gegeben, die Decoys zu aktivieren oder deaktivieren.

Signaturbasierter Dateityp

Als letztes Erkennungsmerkmal wird der Typ einer Datei berücksichtigt, den Scaife et al. (2016) unter anderem in ihrem Prototyp einbauten und nach der Evaluation ein nutzbares Merkmal ist (siehe Abschnitt 3.2.3). Offen bleiben allerdings wesentliche Details wie die Implementierung oder auf welche Weise der Dateityp von einer Verschlüsselung beeinflusst wird. Der FIM wird zunächst auf der Annahme implementiert, dass nach der Verschlüsselung einer Datei kein Typ mehr gefunden werden kann, weil die Signatur verschlüsselt wurde. Ein weiteres Szenario ist ein abweichender Typ von dem in der Baseline befindlichen Typ, weil in der verschlüsselten Bytesequenz zufälligerweise eine Signatur eines Dateityps gefunden wird oder die Ransomware gezielt eine gefälschte Signatur als Umgehungsstrategie einfügt.

4.4.5 Bewertungsmodell

Das Bewertungsmodell (Tabelle 6) beschreibt unter Verwendung der Erkennungsmerkmale die konkreten Ereignisse, für dessen vorkommen jeweils Punkte vergeben werden und der Punktestand erhöht wird. Erst ab einem Punktestand von 200 wird die Benachrichtigung an den Benutzer ausgelöst, sodass die Ereignisse wiederholt auftreten müssen.

Nr.	***Ereignisbeschreibung***	***Erläuterung***	***Punkte***
D-1	Decoy gelöscht	Nach den Verschlüsselungsmustern von Ransomware (Abschnitt 3.2.1) sind alle drei Dateioperationen auf einem Decoy möglich	60
D-2	Decoy modifiziert		40
D-3	Decoy umbenannt		40
E-1	$x^2_{now} < 350$ und $x^2_{base} > 400$ [4]	In Bezug auf den gesamten Dateiinhalt	20
E-2	$x^2_{now} < 350$ und $x^2_{base} > 400$ [4]	In Bezug auf die ersten 256 Bytes einer Datei	30
T-1	Baseline: Dateityp vorhanden Aktuell: Kein Dateityp vorhanden	Nach der Verschlüsselung ist zu erwarten, dass kein Dateityp gefunden wird	40
T-2	Baseline: Kein Dateityp vorhanden Aktuell: Neuer Dateityp vorhanden	Im verschlüsselten Dateiinhalt wurde zufälligerweise eine Signatur gefunden oder die Ransomware fügt eine (gefälschte) Signatur hinzu	5
T-3	Baseline: Dateityp vorhanden Aktuell: Neuer Dateityp vorhanden		5

Tabelle 6: Beschreibung des Bewertungsmodells

In Bezug auf die Ereignisse E-1 und E-2 wird neben dem ursprünglichen Grenzwert von 350 ein zusätzlicher Grenzwert eingeführt, der eine zusätzliche Reserve von 50 Chi-Square-Einheiten umfasst. Dadurch werden Dateien, die in der Baseline bereits einen niedrigen Chi-Square-Wert aufweisen, von der Punktevergabe und dem Einfluss der Bewertung ausgeschlossen.

[4] x^2_{base} ist der Chi-Square-Wert in der Baseline und x^2_{now} der Chi-Square-Wert nach einer Modifikation

4.5 Implementierung

4.5.1 Programmiersprachen

Die Wahl der Programmiersprache fiel auf C#, da diese mit dem hauseigenen .NET-Framework eine gute Kompatibilität zu den nativen Windows-APIs bietet. Dazu ermöglicht C# das unkomplizierte Einbinden von C++ Code, falls ein direkter Zugriff auf den Speicher oder eine besonders leistungskritische Aufgabe erforderlich ist.

Wiederkehrende Aufgaben wurden mit kurzen Hilfsprogrammen in der Programmiersprache Python automatisiert, um den Arbeitsablauf zu vereinfachen. Diese sind jedoch nicht direkter Bestandteil des File-Integrity-Monitors, sondern wurden beispielsweise für das Beschaffen von Ransomware-Samples verwendet oder zur Evaluation eingesetzt.

4.5.2 Libraries und Frameworks

Neben den Standardbibliotheken, die C# anbietet, und dem .NET-Framework, wurde zur Überwachung von Verzeichnissen die Funktion ReadDirectoryChangesW (RDC) der Kernel32-Library von Windows verwendet. Diese nimmt als Eingabe ein Pfad eines Verzeichnisses im Dateisystem und benachrichtigt die aufrufende Applikation über die Erstellung, Löschung, Umbenennung und Modifizierung einer Datei oder einem Unterverzeichnis. Das I/O-Verhalten des Betriebssystems wird von der Funktion nicht beeinflusst, sodass das Dateisystem weiterhin unabhängig operiert und die Anforderung NFA-4 erfüllt wird, die erfordert, dass das Dateisystem im Fehlerfall des FIM weiterhin uneingeschränkt funktioniert.

Zur Realisierung der graphischen Benutzeroberfläche wird das native und moderne UI-Framework namens WinUI3 von Microsoft verwendet, da dieses mit C# kompatibel und nahtlos integrierbar ist. Um den Entwicklungsprozess zu beschleunigen, wird die Erweiterung „WinUI3 Template Studio“ verwendet, die eine grundlegende Projektstruktur aufbaut (Microsoft, 2024).

Um den Typ einer Datei anhand einer Signatur zu identifizieren, wurde die TrIDEngine Programmbibliothek eingebunden (Pontello, 2025), die als Input eine beliebige Datei annimmt und den Typ in Form einer Zeichenkette wie bspw. „Portable Network Graphics“ für eine .png-Datei zurückgibt. Zum Zeitpunkt der Erstellung dieser Arbeit umfasst die Library über 19.000

Signaturen und wird weiterhin aktualisiert. Zwar können nicht alle Dateitypen erkannt werden, doch insbesondere die gängigsten sind darunter abgedeckt.

Damit der Programmablauf nachvollzogen werden kann, werden Log-Dateien erstellt. Zum Einsatz kommt die Bibliothek NLog, welche ein performantes Logging bietet. Außerdem werden die Dateisystemaktivitäten und die Baseline mithilfe der SQLite-Bibliothek in einer lokalen Datenbank gespeichert, die ohne einen externen Server direkt auf dem System des Hosts operiert.

4.5.3 Technischer Kontext

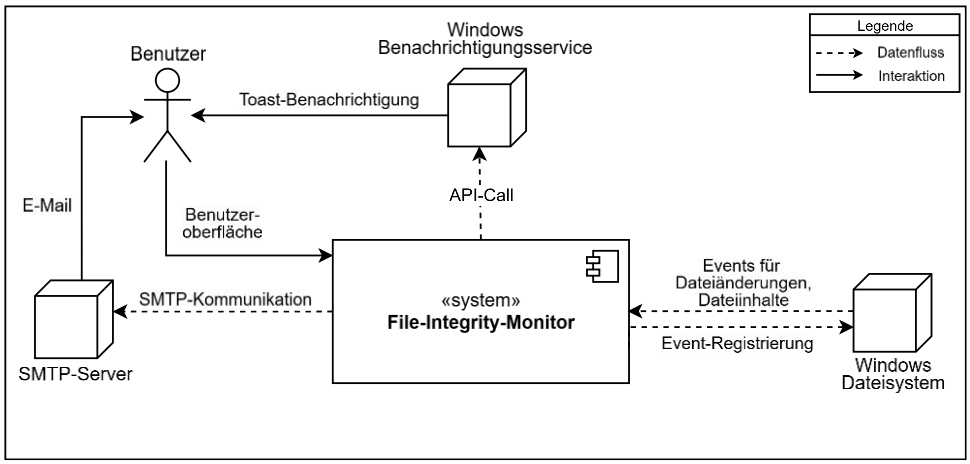

Abbildung 7: Technischer Kontext des File-Integrity-Monitors

In Abbildung 7 ist der technische Kontext (Starke & Hruschka , o. D.) dargestellt, der beschreibt, wie der File-Integrity-Monitor mit den umgebenden Systemen verbunden ist. Das Windows-Dateisystem liefert nach der Registrierung eines Event-Listeners die Ereignisse für die auftretenden Dateiänderungen. Zudem wird es verwendet, um den Inhalt der modifizierten Dateien zu lesen. Der SMTP-Server und der Windows-Benachrichtigungsservice sind für die Benachrichtigung des Benutzers zuständig. Während der SMTP-Server eine E-Mail versendet, wird bei dem Windows-Benachrichtigungsservice eine Toast-Benachrichtigung durch einen Funktionsaufruf ausgelöst.

4.5.4 Bausteinsicht

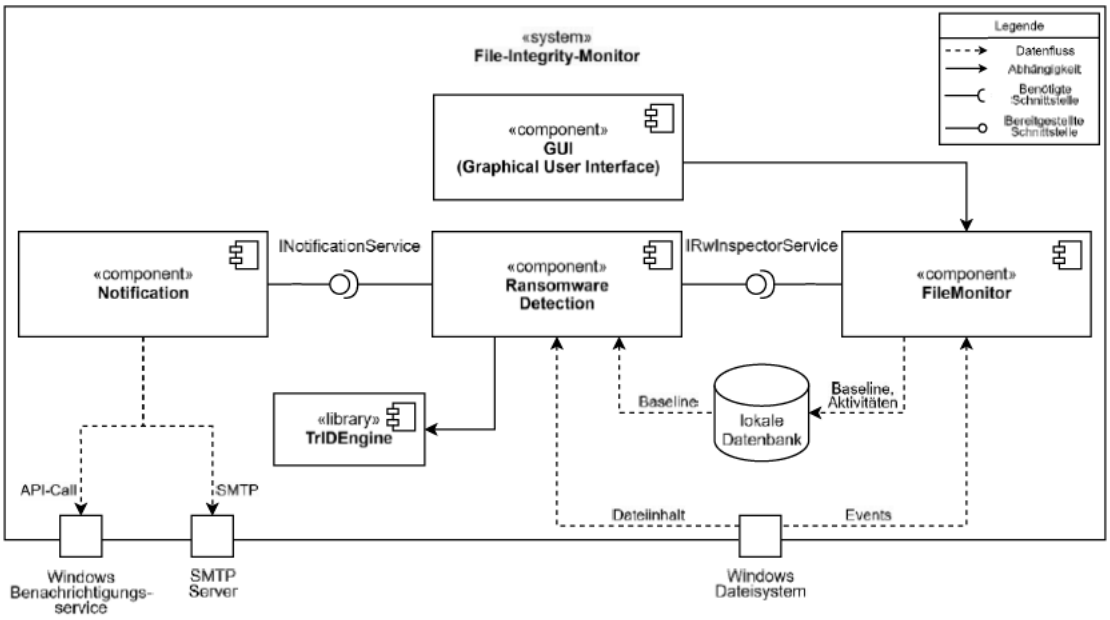

Abbildung 8: Bausteinsicht des File-Integrity-Monitors

Die in Abbildung 8 dargestellte Bausteinsicht (Starke & Hruschka , o. D.) dokumentiert die Architektur mithilfe der wichtigsten Komponenten, die jeweils spezifische Funktionalitäten enthalten, sowie deren Relationen zueinander und den externen Systemen.

Die FileMonitor-Komponente ist zuständig für das Erstellen und Speichern der Baseline in der lokalen Datenbank. Zusätzlich nimmt sie jede Aktivität einer Datei auf, welche ebenfalls in der Datenbank gespeichert wird.

Jegliche Aspekte bezüglich der Erkennungsmerkmale werden in der RansomwareDetection-Komponente gekapselt. Dazu gehört unter anderem die Berechnung der Entropie, das Beschaffen des Dateityps mithilfe der TrIDEngine-Library und die Funktion zur Verteilung sowie Überprüfung der Decoys. Außerdem vergleicht die Komponente die in der Baseline befindlichen Merkmalswerten mit den Werten einer modifizierten Datei und bewertet die einzelnen Vorfälle nach dem Bewertungsmodell.

Sowohl die Toast- als auch die E-Mail-Benachrichtigung erfolgt in der Notification-Komponente, während die GUI-Komponente für die visuelle Darstellung der Anwendung und die Verwaltung der Benutzerinteraktionen zuständig ist.

4.5.5 Programmablauf

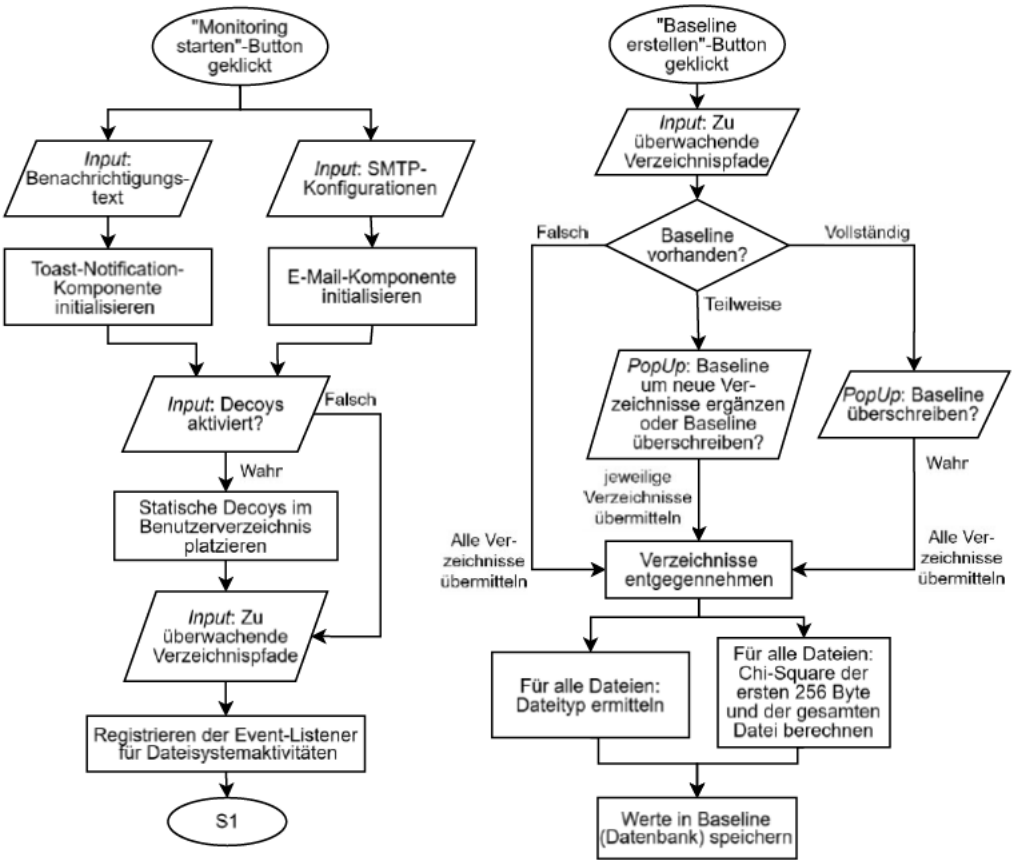

Abbildung 9: Vereinfachter Programmablauf bei dem Starten des Monitorings (links) und bei der Erstellung der Baseline (rechts)

Die Abbildung 9 gibt einen Überblick in Form eines vereinfachten Programmablaufplans wie sich der File-Integrity-Monitor bei dem Starten des Monitorings und bei der Erstellung einer Baseline verhält. Beide Prozesse sind voneinander getrennt, sodass der Benutzer zuerst eine Baseline erstellt und anschließend das Monitoring starten kann. Alle mit „Input“ bezeichneten

Elemente beziehen sich auf die von dem Benutzer in die graphische Benutzeroberfläche eingegebenen Daten.

Bei der Erstellung der Baseline werden die zu überwachenden Verzeichnispfade ausgelesen und in der Datenbank nach einer bereits existierenden Baseline geprüft. Falls keine Baseline vorhanden ist, erfolgt die Iteration über die in den Verzeichnissen enthaltenen Dateien, bei der die Entropiewerte und der Dateityp ermittelt und in die Baseline gespeichert wird. Wenn bereits alle Verzeichnisse in der Baseline enthalten sind, wird der Benutzer mithilfe eines Dialogfensters (Popup-Fenster) dazu aufgefordert, anzugeben, ob die Baseline neu erstellt werden soll. Sollte lediglich ein Teil der Verzeichnisse bereits in der Baseline sein, wird dem Benutzer die Option geboten, entweder nur neue Verzeichnisse in die Baseline aufzunehmen oder die gesamte Baseline zu überschreiben. In jedem Fall werden die Chi-Square-Werte der ersten 256 Bytes und des gesamten Dateiinhalts berechnet sowie der Dateityp ermittelt.

Wenn das Monitoring eingeleitet wird, werden die SMTP-Konfigurationen und der Benachrichtigungstext ausgelesen und die dazugehörigen Komponenten mit diesen Daten initialisiert. Anschließend wird geprüft, ob die Decoy-Funktion aktiviert ist. Im positiven Fall werden die Decoys im Benutzerverzeichnis nach dem im Abschnitt 4.4.4 beschriebenen Muster verteilt. Zuletzt werden die von dem Benutzer definierten Verzeichnisse an die ReadDirectoryChangesW Funktion übermittelt, mit dem der FIM die einzelnen aufgetretenen Dateiaktivitäten erhält.

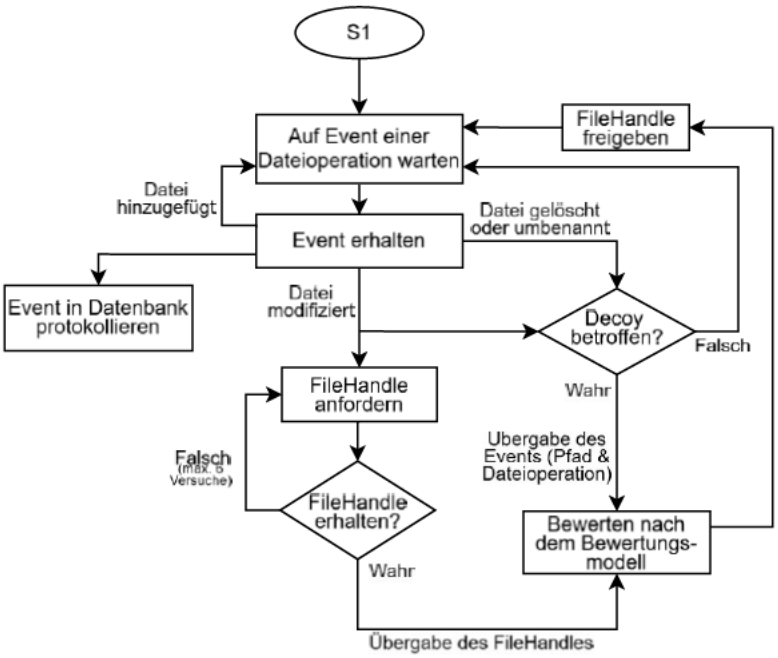

Abbildung 10: Vereinfachter Programmablaufplan bei der Dateiüberwachung

Nachdem der initiale Ablauf der Anwendung beschrieben wurde, veranschaulicht Abbildung 10 den weiteren Verlauf des Programms, wenn die Events von Dateiänderungen bei der aktiven Überwachung eintreffen.

Dateien, die zu einem überwachten Verzeichnis hinzugefügt wurden, werden abgesehen von der Protokollierung nicht weiter berücksichtigt. Modifizierte, gelöschte und umbenannte Dateien werden unmittelbar darauf geprüft, ob es sich um Decoys handelt. Ist dies der Fall, werden sowohl der Pfad zur Decoy-Datei als auch die dazugehörige Dateioperation an das Bewertungsmodell übergeben.

Bei modifizierten Dateien muss der FIM den **FileHandle** anfordern, um den Dateityp und die Entropie ermitteln zu können. Falls das Handle nicht erlangt werden kann, beispielsweise weil ein anderer Prozess die Datei blockiert, muss der FIM warten, bis die Blockierung aufgehoben ist. Daher folgen nach dem ersten fehlgeschlagenen Versuch fünf weitere Wiederholungsversuche nach den definierten Zeitintervallen von 100, 250, 750, 1500 und 3500 Millisekunden. Wenn der FileHandle erlangt werden konnte, wird dieser an das Bewertungsmodell übergeben,

um dort die Entropiewerte und den Dateityp zu ermitteln. Die Funktionsweise des Bewertungsmodells ist in dem nachfolgenden Programmablaufplan (Abbildung 11) dargestellt.

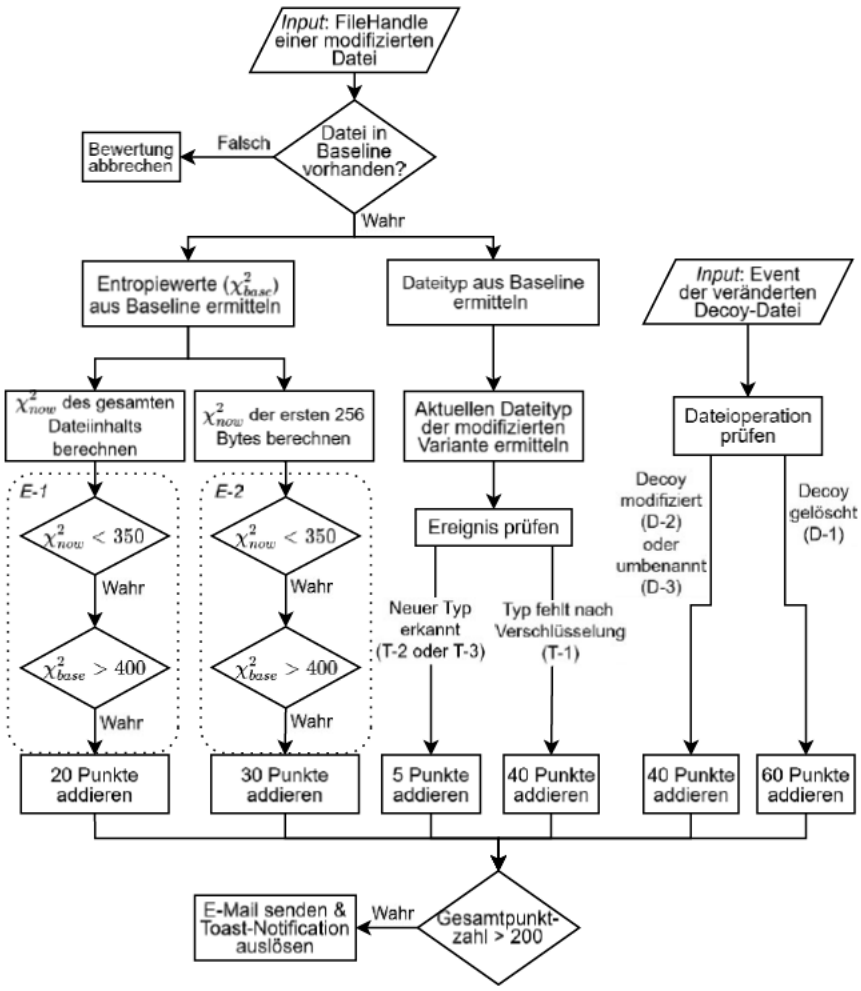

Abbildung 11: Vereinfachter Programmablaufplan des Bewertungsmodells

Um eine Bewertung anhand der Entropie und des Dateityps durchzuführen, wird die Datei zunächst mithilfe des FileHandles identifiziert und die dazugehörigen Werte aus der Baseline beschafft. Falls diese Werte nicht vorhanden sind, wird die Bewertung abgebrochen. Die Entropie der modifizierten Datei wird sowohl für die gesamte Datei als auch über die ersten 256 Bytes berechnet (x^2_{now}) und anschließend mit den Werten aus der Baseline (x^2_{base}) bezüglich

der festgelegten Grenzwerte geprüft. Wenn die Grenzwertprüfungen nicht zu wahr auswerten, werden keine Punkte addiert. Auf ähnliche Weise funktioniert die Bewertung anhand des Dateityps und der Decoys, allerdings werden hier keine Grenzwerte geprüft, sondern einfache Vergleiche zwischen Zeichenketten durchgeführt. Nach jeder Entropie- und Dateitypbewertung oder der Bewertung einer Decoy-Modifikation wird die Gesamtpunktzahl mit dem Grenzwert von 200 verglichen und bei einem größeren Wert eine Alarmierung ausgelöst.

Zur Bewertung der Veränderungen von Decoys wird kein FileHandle benötigt, da die Dateioperation und der Dateipfad aus dem Event extrahiert und direkt an das Bewertungsmodell übergeben werden.

4.6 Funktionsumsetzung

Im Nachfolgenden wird die fertige Umsetzung der Anwendung anhand von Screenshots gezeigt und der Ablauf aus Sicht des Benutzers detailliert beschrieben.

In Abbildung 12 ist ein Screenshot der Hauptansicht der Benutzeroberfläche zu sehen, in der die meisten Funktionalitäten integriert sind. In dem Bereich der Verzeichnisüberwachung kann der Benutzer Verzeichnisse über die Schaltfläche „Ordner hinzufügen“ einpflegen oder über die neben den jeweiligen Verzeichnissen angezeigte rote Schaltfläche wieder entfernen. Des Weiteren besteht die Möglichkeit, eine Baseline zu erstellen, um den Zustand der konfigurierten Verzeichnisse zu einem bestimmten Zeitpunkt als Referenz festzulegen oder neue Verzeichnisse zu einer bereits bestehenden Baseline hinzuzufügen, ohne dass eine eventuell vorhandene Baseline gelöscht wird. Alternativ kann sich der Benutzer auch dafür entscheiden die gesamte Baseline neu zu berechnen. Die Entscheidung zwischen beiden Optionen trifft der Benutzer mithilfe eines Dialogfenster, welches zur Laufzeit eingeblendet wird.

Das Monitoring der Verzeichnisse kann über die „Monitoring starten“-Schaltfläche aktiviert und bei erneuter Betätigung wieder gestoppt werden. Die Schaltfläche ändert dabei sowohl die Farbe als auch den Text, um dem Benutzer den aktuellen Status anzuzeigen.

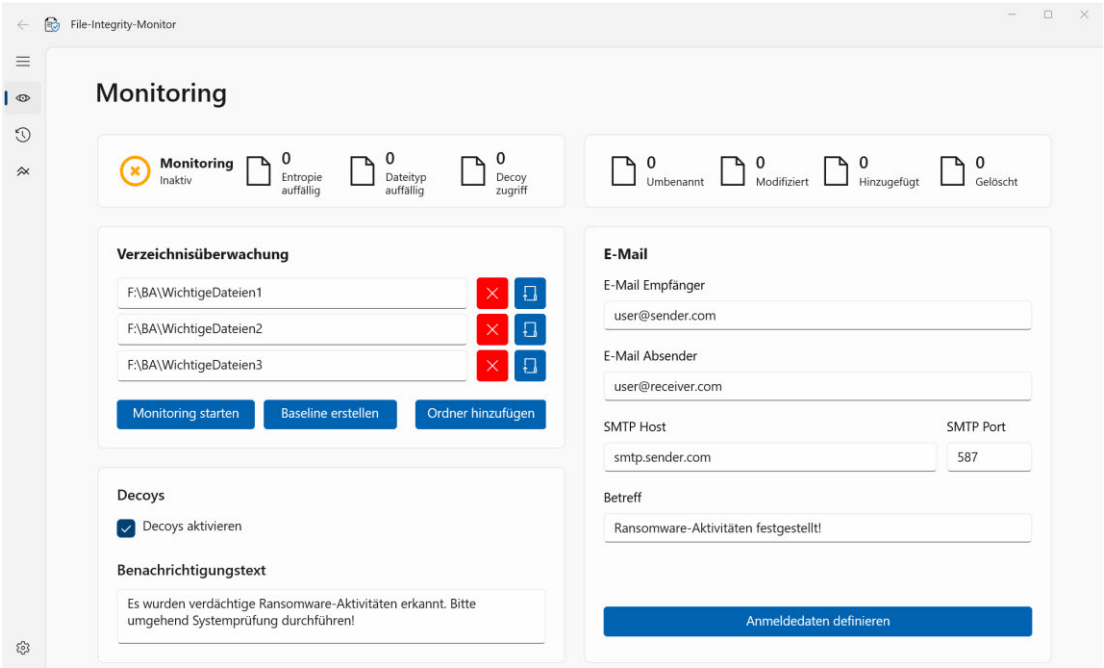

Abbildung 12: Hauptansicht des File-Integrity-Monitors

Im „E-Mail“-Bereich konfiguriert der Nutzer einen SMTP-Server, der für den Versand von Benachrichtigungen per E-Mail erforderlich ist. Zusätzlich sind Details wie die Empfängeradresse, die Absenderadresse und der E-Mail-Betreff festzulegen. Über die Schaltfläche „Anmeldedaten definieren“ öffnet sich ein Dialogfenster, wo der Nutzer die für den SMTP-Server benötigten Anmeldeinformationen eingibt.

In dem Bereich unten links, kann sich der Benutzer dafür entscheiden, die Decoys als Merkmal zur Erkennung zu aktivieren oder zu deaktivieren. Außerdem kann der Benachrichtigungstext definiert werden, der sowohl den Inhalt der E-Mail als auch der Toast-Notification beschreibt.

Im oberen Bereich erhält der Benutzer eine Übersicht über den aktuellen Status des Monitorings. Darüber hinaus wird ersichtlich wie viele der überwachten Dateien bei einer Modifizierung entropieauffällig (E-1 oder E-2) oder dateitypauffällig (T-1, T-2 oder T-3) sind und wie viele Zugriffe auf Decoys (D-1, D-2 oder D-3) erfolgten. In ähnlicher Weise ist auf der rechten Seite eine Übersicht über die Anzahl der Dateien, die seit dem letzten Monitoring umbenannt, modifiziert, hinzugefügt oder gelöscht worden sind.

Die Navigationsleiste erlaubt es dem Nutzer in die Dateiaktivitäten- und Baseline-Ansicht (siehe Abbildung 13 bzw. Abbildung 14) zu wechseln. Die Dateiaktivitäten-Ansicht bietet einen Überblick über die protokollierten Änderungen, darunter die ausgeführten Dateioperationen und der dazugehörige Zeitstempel, während die Baseline-Ansicht die Entropie- und die Dateitypwerte sowie den Erstellungszeitstempel anzeigt.

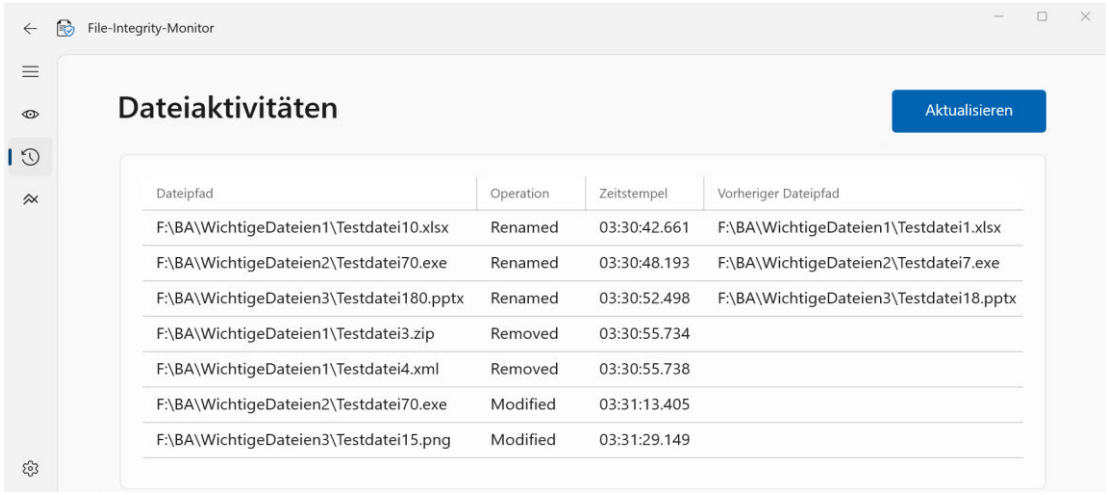

Abbildung 13: Dateiaktivitäts-Ansicht des File-Integrity-Monitors

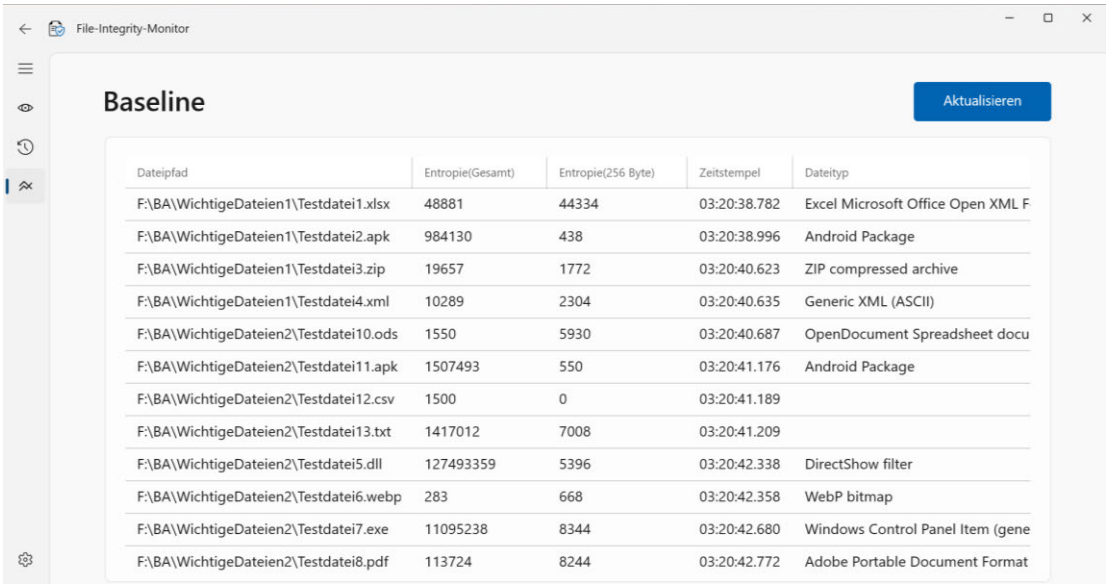

Abbildung 14: Baseline-Ansicht des File-Integrity-Monitors

5 Evaluierung des File-Integrity-Monitors

In der Evaluation soll untersucht werden, inwiefern der File-Integrity-Monitor die Fähigkeit zur Feststellung eines Ransomware-Angriffs hat. Einhergehend damit wird überprüft, ob das Programmverhalten der entwickelten Software auf bestimmter Weise von der Ransomware negativ beeinträchtigt wird, um beispielsweise einer Detektion zu umgehen.

Im Zuge dieses Kapitels sind drei Leitfragen zu beantworten:

(1) Wie viele der getesteten Ransomware-Samples wurden erfolgreich erkannt?
(2) Weshalb wurde ein Ransomware-Sample möglicherweise nicht erkannt, falls dies der Fall ist?
(3) Wie effektiv zeigten sich die eingesetzten Merkmale bei der Erkennung von Crypto-Ransomware in Bezug auf das eingesetzte Bewertungsmodell?

5.1 Testumgebung

Evaluiert wird der FIM anhand realer Ransomware-Samples in einer isolierten Umgebung mittels virtueller Maschinen (VM). Auf dem Host-System wird das Betriebssystem Ubuntu 22.04.5 LTS mit der Virtualisierungssoftware VMware Workstation Pro 17.6.1 ausgeführt. Darunter werden drei virtuelle Maschinen erstellt, die im Nachfolgenden beschrieben werden und deren Zusammenhang in Abbildung 15 verdeutlicht wird.

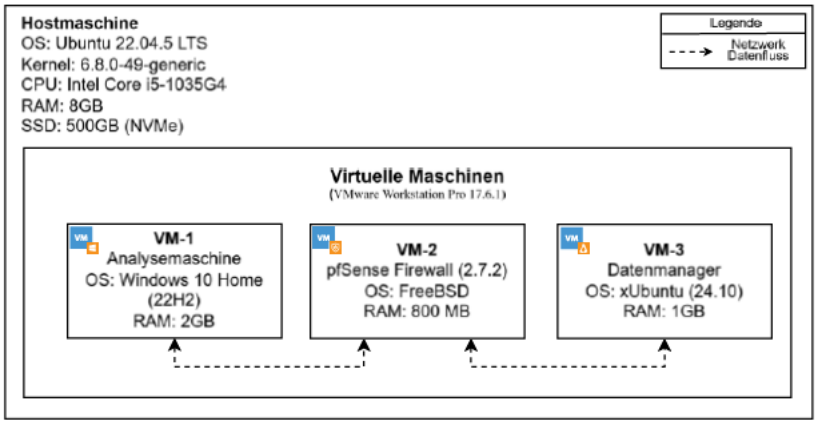

Abbildung 15: Aufbau der Testumgebung

VM-1 ist die virtuelle Maschine, auf denen die Ransomware-Samples ausgeführt werden. Die Basis dieser VM ist Windows 10 Home der Version 22H2. Es wurde gezielt darauf geachtet eine möglichst realistische Benutzerumgebung zu schaffen mittels Dummy-Dateien sowie gängigen Softwareanwendungen. Da Malwares virtuelle Maschinen erkennen können, wurden hier einige Maßnahmen getroffen: Der Open-Source-Patch „VMwareHardenedLoader“ wurde auf dem Gast-Betriebssystem ausgeführt, der dafür sorgen soll, dass alle möglichen Spuren von Strings wie „VMware“, „Virtual“ oder „VMWARE“ entfernt werden (hzqst, 2022). Zusätzlich wurde die MAC-Adresse des Netzwerkadapters geändert, da die ersten 24 Bit auf das Virtualisierungsprogramm zurückzuführen sind, und sämtliche Anti-Malware-Systeme wie der Windows-Defender und Windows-SmartScreen deaktiviert, die das Ausführen der Samples verhindern würden.

Auf die virtuelle Maschine VM-3 werden jegliche Dateien über das File Transfer Protocol (FTPS) übertragen, die nach dem Ausführen eines Ransomware-Samples zur Analyse benötigt werden. Außerdem stellt diese VM einen lokalen SMTP-Server mithilfe des Mail Transfer Agent Postfix bereit, um die E-Mail-Benachrichtigung zu testen. Die virtuelle Maschine basiert auf xUbuntu, eine Variante von Ubuntu mit geringeren Hardwareanforderungen.

Die zweite virtuelle Maschine (VM-2) ist für die Kommunikation zwischen VM-1 und VM-3 zuständig und enthält eine Firewall, die so konfiguriert ist, dass ausschließlich der Port für die Verbindung zum FTP- und dem SMTP-Server freigegeben ist, während alle anderen Verbindungen blockiert werden.

5.2 Auswahl der Ransomware Samples

Die Auswahl der Samples orientiert sich an den populären Crypto-Ransomwares seit 2016. Insgesamt wurden 15 Samples aus unterschiedlichen Familien ausgewählt, die in Tabelle 7 aufgelistet sind. Diese stammen von VirusShare (o. D.) und MalwareBazaar (o. D.), zwei bekannten Plattformen, die Malware-Samples für Forschungs- und Analysezwecke bereitstellen. Es wurden ausschließlich Samples ausgewählt, die einer Portable Executable in Form einer .exe-Datei entsprechen.

Ransomware Familien (Datum)		
Akira (08.24)	DarkSide (05.21)	CL0P (02.19)
BlackBasta (02.23)	Conti (05.21)	GandCarb (07.18)
BlackCat / ALPHV (03.22)	Revil / Sodinokibi (04.21)	Cerber (05.17)
LockBit (08.21)	Maze (06.20)	WannaCry (04.17)
BlackMatter (08.21)	Ryuk (03.20)	TeslaCrypt (02.16)

Tabelle 7: Ausgewählte Familien je Sample mit zusätzlicher Angabe des Datums der ersten Einreichung auf VirusTotal

Die Samples wurden in der Testumgebung ausgeführt, um die Funktionstüchtigkeit zu überprüfen und die erwarteten Eigenschaften einer Crypto-Ransomware festzustellen. Falls dies nicht der Fall war, wurde nach einem anderen Sample derselben Ransomware-Familie gesucht, bis ein funktionsfähiges Exemplar identifiziert werden konnte.

5.3 Testablauf

Vor dem Start eines Ransomware-Samples wird der File-Integrity-Monitor ausgeführt und die folgenden Benutzerverzeichnisse zur Überwachung übergeben: Desktop, Downloads, Dokumente, Bilder und Videos. Dabei nimmt die Baseline insgesamt 60 Dateien auf, darunter Dateiformate wie pdf, xlsx, docx, zip, ods, txt, png, jpg, mp4, sql und accdb. Von den erfassten 60 Dateien sind zwölf Decoys, die sich nach dem Starten des FIM in dem Benutzerverzeichnis befinden und nach dem im Abschnitt 4.4.4 beschriebenen Muster verteilt sind. Für zwei Dateien konnte kein Chi-Square-Wert der ersten 256 Bytes berechnet werden, weil diese kleiner als 256 Bytes sind. Analog dazu konnten für neun Dateien kein Dateityp bestimmt werden.

Mithilfe eines Python-Skripts wurde in jedem Testdurchlauf ein Sample ohne administrative Rechte gestartet und die Startzeit protokolliert. Der Prozess wurde so lange ausgeführt, bis dieser sich selbständig beendet. Im Falle von Prozessen, die auch nach der Verschlüsselung weiterhin ausgeführt werden, wurde eine maximale Laufzeit von ca. 15 Minuten definiert, nach

der der Prozess manuell terminiert wird, falls keine weiteren Aktivitäten festgestellt werden konnten.

Nach der Ausführung bzw. Terminierung eines Samples werden die überwachten Verzeichnisse sowie die Log- und Datenbankdatei über das File Transfer Protocol an die VM-3 übermittelt und die virtuelle Maschine in den ursprünglichen, nicht kompromittierten Zustand zurückgesetzt. Dies wird mittels der Snapshot-Funktion der Virtualisierungssoftware umgesetzt, die es ermöglicht, einen Schnappschuss des Zustandes einer VM zu erstellen, sodass dieser zu einem späteren Zeitpunkt wiederhergestellt werden kann.

5.4 Ergebnisse

Die Auswertung der Testdurchläufe ergab, dass bei 14 von 15 Samples eine Benachrichtigung an den Benutzer in Form einer Toast- als auch E-Mail-Benachrichtigung erfolgreich angezeigt bzw. zugestellt werden konnte. Die Anforderung NFA-5 erfüllt wurde, die eine Erkennung von mindestens 90 % der getesteten Samples erforderte. Durchschnittlich verschlüsselten die Schadprogramme rund 86 % der in der Baseline enthaltenen Dateien, was für die Evaluation als ausreichend betrachtet wird.

Die BlackCat-Familie konnte nicht erkannt werden, da der FIM unmittelbar nach dem Start des Samples ein Fehler in der Laufzeitumgebung verzeichnet. Ein ähnliches Verhalten wurde bei der BlackBasta-Familie festgestellt. Der FIM erkannte zwar das Sample, jedoch beendete sich der FIM-Prozess nach der Verschlüsselung von 31 Dateien mit einer AccessViolationException, die den ungültigen Zugriff auf einen geschützten Speicherbereich meint. Aufgrund unzureichender Daten wird BlackCat in den nachfolgenden Ergebnissen nicht weiter betrachtet. Für die Untersuchung von BlackBasta hingegen liegen genügend verwertbare Daten vor.

In beiden o. g. Fällen deutet die Ursache auf die gezielte Termination des FIM-Prozesses, sobald die Ransomware eine Datei nicht verschlüsseln konnte, weil der FIM über einen exklusiven Zugriff dieser Datei verfügt. Der FIM fordert beispielsweise für die lokale Datenbankdatei und der Log-Datei einen exklusiven Zugriff an, um diese vor einer Verschlüsselung zu bewahren. Zudem ist zur Vermeidung eines inkonsistenten Entropiewertes kurzzeitig ein exklusiver Zugriff auf die modifizierte Datei nötig, um zu verhindern, dass während der Berechnung

andere Prozesse Schreibvorgänge durchführen. Es empfiehlt sich eine Laufzeitanalyse des Codes durchzuführen, um das Kernproblem des Fehlers festzustellen und eine Lösung dafür zu finden. Da der Fehler auf eine Manipulation des Prozessspeichers hindeutet, kann unter Windows der Versuch unternommen werden, die Sicherheitsfunktion namens Protected Process Light (PPL) zu integrieren, die kritische Prozesse vor einer Termination und der Manipulation des Speichers schützt (Kaspersky, 2018).

Im Rahmen der Evaluation erfolgt eine Analyse der Häufigkeit der im Bewertungsmodell beschriebenen Ereignisse, wie in den nachfolgenden Diagrammen dargestellt.

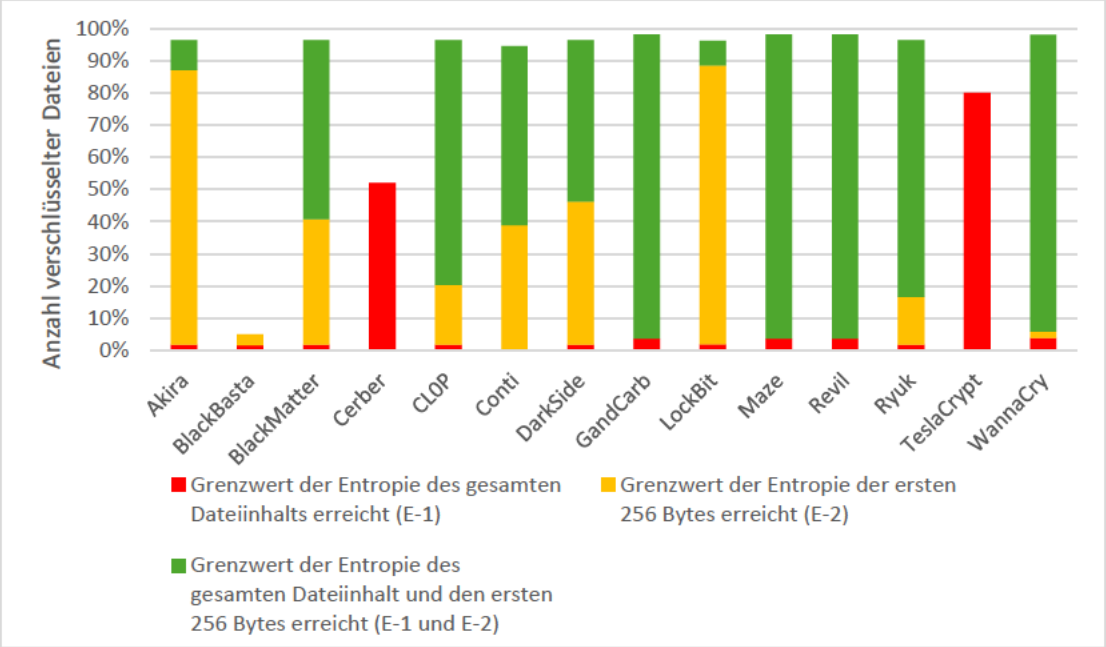

Abbildung 16: Ergebnisse der erkannten Dateien auf Basis der Entropie je Sample

Abbildung 16 gibt einen Überblick über die Anzahl der Dateien, die der FIM mittels der Entropie erkannt hat, entweder anhand der Entropie der ersten 256 Bytes (E-2), des gesamten Dateiinhalts (E-1) oder der Kombination (E-1 und E-2). Zur besseren Verständlichkeit des Diagramms wird das BlackMatter-Sample als Beispiel herangezogen. Die Säule erreicht eine Höhe von ca. 96 % und verdeutlicht, dass rund 96 % der Dateien, die BlackMatter verschlüsselte, von dem FIM erkannt wurden. Davon unterschritten knapp 39 % (gelber Anteil) ausschließlich

den eingestellten Grenzwert für den Entropiewert der ersten 256 Bytes, etwa 2 % (roter Anteil) nur den Entropiewert des gesamten Dateiinhalts und ca. 55 % (grüner Anteil) sowohl den Entropiewert für die ersten 256 Bytes als auch den des gesamten Inhalts.

In gesamter Betrachtung ist vor allem positiv hervorzuheben, dass die Kombination der Ereignisse E-1 und E-2 (grüner Anteil) am häufigsten vorkommen, da nach dem Bewertungsmodell für diese Fälle die meisten Punkte pro Datei vergeben werden. Allerdings stechen Samples wie Akira und LockBit vor allem mit einem hohen Anteil des Ereignisses E-2 heraus. Der Grund hierfür ist, dass LockBit ausschließlich die ersten 4096 Bytes und Akira die erste Hälfte einer Datei verschlüsselt, wodurch nur der Grenzwert der ersten 256 Bytes, aber nicht der des gesamten Dateiinhalts unterschritten wurde. Die Samples BlackMatter, Conti und DarkSide weisen einen geringeren Anteil des Ereignisses E-2 auf, weil nur Dateien abschnittsweise verschlüsselt werden, die größer als 100 MB sind.

Bei dem BlackBasta-Sample sind starke Abweichungen zu beobachten, da im Vergleich zu den anderen Samples nur 5 % der verschlüsselten Dateien vom FIM erkannt wurden. Über den gesamten Datenstrom wendet das Sample ein Muster an, bei dem ein 64-Byte-Block verschlüsselt wird und der nächste 128-Byte-Block im Klartext verbleibt. Die Verschlüsselung solcher kleinen Abschnitte reduziert den Entropiewert nicht ausreichend, um unter den Grenzwert zu fallen, was eine Schwäche des Bewertungsmodells darstellt.

TeslaCrypt und Cerber fallen vor allem nur durch das Auftreten von E-1 auf. Bei TeslaCrypt ist die Bytesequenz der ersten 364 Bytes aller verschlüsselten Dateien identisch, was eine Art Signatur der Familie darstellt. Dies führte stets zu denselben Entropiewerten der ersten 256 Bytes, welche über den eingestellten Grenzwert lagen und deshalb keine Datei anhand dieses Kriteriums erkannt werden konnte. Cerber weist ein ähnliches Verhalten auf, da erst ab dem 1.793. Byte mit der Verschlüsselung begonnen wird. Dies hat den zusätzlichen Effekt, dass der Chi-Square-Wert für alle Bytes einer Datei höher ausfallen und über dem Grenzwert liegen kann, wie es sich teilweise im Falle von Cerber bestätigte.

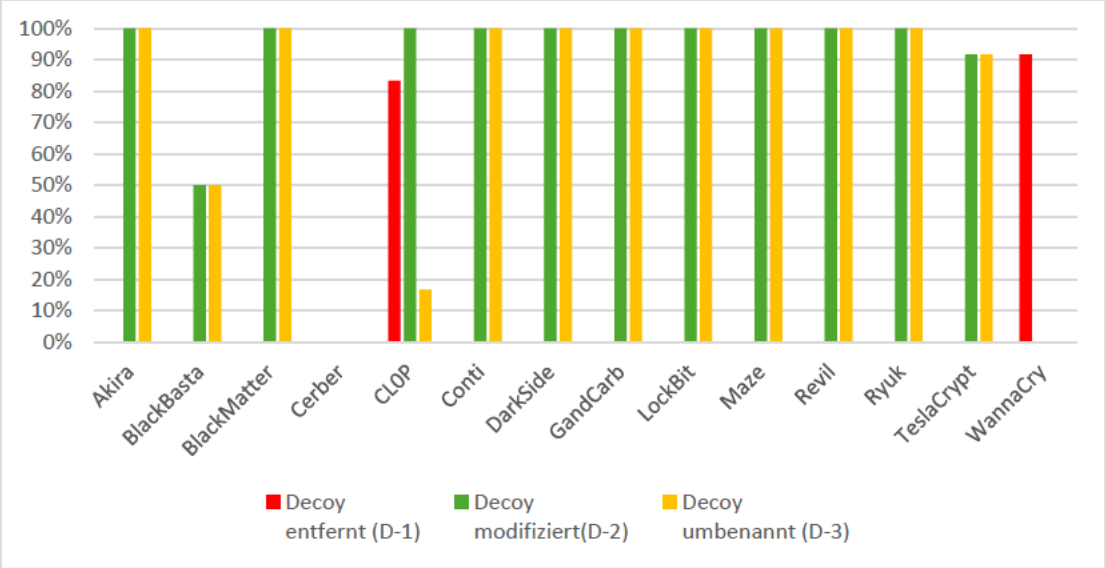

Abbildung 17: Ergebnisse der manipulierten Decoys je Sample

Die Abbildung 17 zeigt die Ereignisse in Bezug auf die Modifikationen der Decoys, die vom FIM erkannt wurden. Es lässt sich beobachten, dass zehn Samples alle im Dateisystem versteckten Decoys verschlüsselten. Sowohl WannaCry als auch TeslaCrypt ignorierten die Decoy-Datenbankdatei mit der Endung .sqlite, was allerdings kein Problem hinsichtlich der Erkennung darstellt.

Lediglich das Sample der Cerber-Familie zeigt sich als Ausreißer, welches die Decoys nicht verschlüsselte, da der Angriff ausschließlich auf die Dateien in dem Desktop- und Dokumente-Verzeichnis ausgerichtet war, in denen sich keine Decoys befanden. Dieses Ergebnis legt nahe, dass die Decoys weitreichender im Dateisystem verteilt werden sollten, um die Wahrscheinlichkeit einer Verschlüsselung durch Ransomware zu erhöhen, wenn nur bestimmte Verzeichnisse kompromittiert werden. Um herauszufinden, wie sich das Sample bezüglich versteckter Dateien verhält, wurde ein weiterer Testdurchlauf durchgeführt, bei dem festgestellt wurde, dass Cerber nur versteckte Dateien verschlüsselt, die sich nicht in einem versteckten Verzeichnis befinden.

Die strategische Platzierung der Decoys führte dazu, dass 13 Samples – im Kontext der überwachten Verzeichnisse – mit der Verschlüsselung der Decoys begonnen haben, wobei jedoch nicht direkt alle der zwölf Decoys betroffen waren.

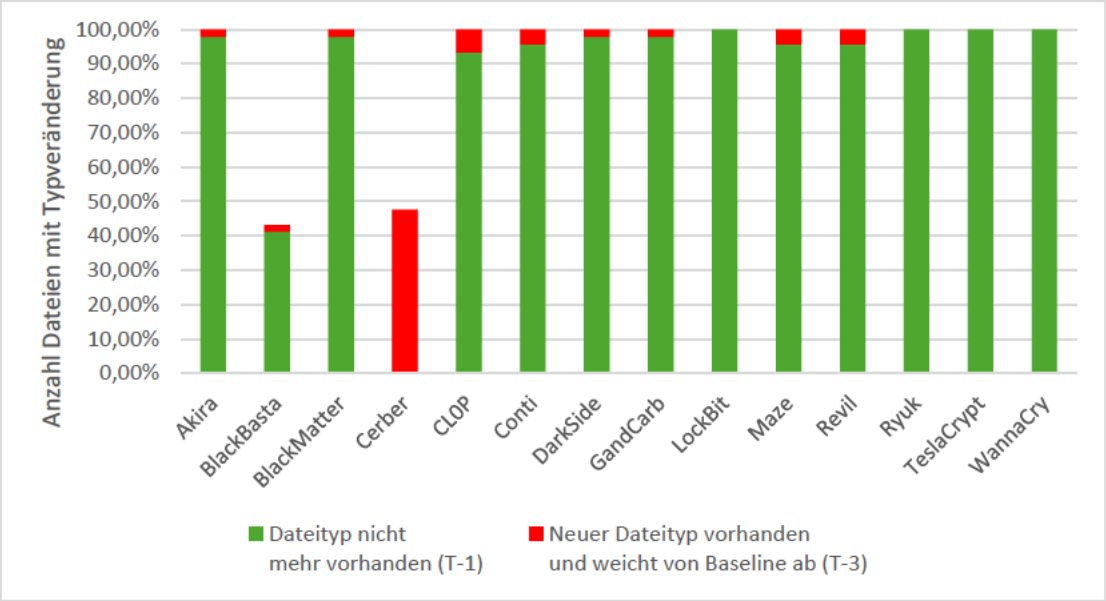

Abbildung 18: Ergebnisse der erkannten Dateien auf Basis des Dateityps je Sample

Das Diagramm in Abbildung 18 zeigt die Veränderungen des Dateityps nach der Verschlüsselung, die der FIM erkannt hat. Die Höhe der Säulen symbolisiert die Anzahl der Dateien, denen ein Typ in der Baseline zugeordnet werden konnte und nach der Verschlüsselung entweder von diesem abweicht (T-3, roter Anteil) oder dieser nicht mehr vorhanden ist (T-1, grüner Anteil).

Bemerkenswerterweise stellte der FIM bei zwölf Samples fest, dass jeweils mind. 95 % der verschlüsselten Dateien, die in der Baseline einen Dateityp hinterlegt hatten, nach der Verschlüsselung keinem Typ mehr zugeordnet werden konnten (T-1).

Ähnlich wie bei den Ergebnissen der Erkennung anhand der Entropie ist auch hier BlackBasta als Ausreißer auffällig. Allerdings ist positiv hervorzuheben, dass ungefähr 41 % der verschlüsselten Dateien nach der Verschlüsselung keinen Dateityp mehr aufweisen (T-1), während anhand der Entropie nur 5 % als verschlüsselt erkannt wurden. Damit zeigt sich dieser Ansatz als effektiver, aber nicht optimal bei Samples, die in sehr kleinen Abschnitten verschlüsseln.

Cerber zeigt im Vergleich zu den anderen Samples eine höhere Anzahl der Dateien, die nach der Verschlüsselung einen anderen Dateityp aufweisen. Dies lässt sich dadurch erklären, dass einige Dateien mit mehreren Dateitypen übereinstimmen. Beispielsweise basieren die Dateien von Microsoft-Office auf das Open Packaging Conventions Format, dessen Signatur in den ersten Bytes zu finden ist. Die eigentliche Erkennung als z. B. Word-Dokument erfolgt anhand der Signatur, die sich nicht im Header, sondern in späteren Abschnitten der Datei auffinden, welche jedoch von Cerber verschlüsselt wurden. Die anderen Samples weisen einen solchen Zusammenhang allerdings nicht auf und sind auf das zufällige Vorhandensein einer Signatur zurückzuführen.

Das T-2 Ereignis, welches die Hinzufügung eines neuen Dateityps beschreibt, wobei in der Baseline kein Typ vorhanden war, wurde nicht in das Diagramm aufgenommen, da es kein einziges Mal aufgetreten ist. Dies ist unter anderem darauf zurückzuführen, dass nur wenige Dateien in der Baseline vorhanden sind, die keinen Typ besitzen. Für größere Mengen dieser Art von Dateien ist eine Korrelation mit dem Ereignis T-3 zu erwarten, weil beide den Fall einer zufällig vorhandenen Signatur abdecken. Da T-2 im Vergleich zu T-1 deutlich seltener auftreten und diese nur marginal die Punkteanzahl im Bewertungsmodell beeinflussen, kann der Ausschluss von T-2 und T-3 aus dem Modell in Erwägung gezogen werden.

5.5 Diskussion

5.5.1 Beantwortung der Evaluationsfragen

Die zum Beginn der Evaluation formulierten Leitfragen werden in diesem Abschnitt erneut aufgegriffen und auf Basis der in der Evaluation gewonnenen Erkenntnisse beantwortet.

(1) Wie viele der getesteten Ransomware-Samples wurden erfolgreich erkannt?

Von 15 getesteten Ransomwares konnten 14 anhand der bösartigen Verschlüsselung erkannt werden.

(2) Weshalb wurde ein Ransomware-Sample möglicherweise nicht erkannt, falls dies der Fall ist?

Die Ausführung zweier Samples resultierte in die Beendigung des FIM durch einen Fehler. Dies wurde auf den exklusiven Zugriff der Dateien zurückgeführt, der für die korrekte Funktionsweise des FIM notwendig ist. Durch den exklusiven FileHandle wird verhindert, dass eine Ransomware beispielsweise die Baseline oder die Log-Datei verschlüsselt, allerdings umgingen beide Samples dies, indem sie den Prozess gezielt manipulieren und zum Absturz bringen. Eines der Samples führte unmittelbar zum Absturz, da dieses bereits sehr früh im Angriffsprozess versuchte, eine Datei mit exklusiven Zugriff zu verschlüsseln, weshalb durch den FIM keine Erkennung erfolgte. Das andere Sample hingegen führte den Absturz erst nach einer gewissen Verzögerung herbei, da das Sample erst zu einem späteren Zeitpunkt eine gesperrte Datei zu verschlüsseln versuchte. Dadurch konnte der FIM genügend auffällige Dateimodifikationen monitoren und eine Benachrichtigung aussenden.

(3) Wie effektiv zeigten sich die eingesetzten Merkmale bei der Erkennung von Crypto-Ransomware in Bezug auf das eingesetzte Bewertungsmodell?

Ursprünglich wurde der Entropiewert der ersten 256 Bytes in dem FIM integriert, damit mehr Dateien mit einer bereits hohen Entropie anhand dieses Wertes bewertet werden können (siehe Abschnitt 4.4.4). In der Evaluation zeigte sich, dass dieses Merkmal als Ergänzung zum Entropiewert des gesamten Dateiinhalts aufgrund der teilweisen Verschlüsselung fungiert. Während bei den Samples bis 2021 überwiegend vollständige Verschlüsselungen zu beobachten sind, wenden neuere Samples eine teilweise Verschlüsselung an. Die Auswertung zeigt zwei Schwachstellen zur Umgehung beider Entropiewerte: Entweder werden die Daten in sehr kleinen Abschnitten (kleiner als 256 Bytes) verschlüsselt, oder die Verschlüsselung startet nicht ab dem 1. Byte, sondern versetzt[5]. Die Verwendung eines speziellen Encodings (z. B. Base64) zur

[5] Wie stark die Verschlüsselung versetzt werden muss, hängt unter anderem von der Dateigröße ab.

Umgehung der Entropie (siehe Abschnitt 3.2.2) wurde bei keinem Sample festgestellt, ist jedoch weiterhin eine potenzielle Schwachstelle, die ausgenutzt werden kann.

Da der Dateityp ebenfalls auf dem Dateiinhalt basiert, sind Zusammenhänge mit der Entropie als Merkmal zu beobachten. In den meisten Fällen fehlte die Signatur des Dateityps, allerdings ist im Falle einer abschnittsweisen Verschlüsselung unvorhersehbar, ob die Dateityp-Signatur verschlüsselt wird. Das Einfügen eines Dateityps in die verschlüsselten Daten seitens der Ransomware wurde nicht festgestellt, wäre jedoch eine mögliche Umgehungsstrategie.

Zur Vermeidung unbeabsichtigter Änderungen durch den Benutzer und der damit verbundenen Benutzertransparenz wurden die Decoys versteckt in dem Dateisystem hinterlegt. Zwar bietet dies der Ransomware einen Angriffspunkt zur Umgehung, indem versteckte Dateien ignoriert werden, jedoch zeigte die Evaluation mit der Verschlüsselung der Decoys bei 13 Samples ein gegenteiliges Ergebnis.

Zusammenfassend lässt sich aus der Evaluation ableiten, dass sich die genutzten Merkmale überwiegend als effektiv erwiesen, auch wenn vereinzelt Ausnahmen zu identifizieren waren. Allerdings gilt zu beachten, dass die Evaluation in einer kontrollierten Umgebung mit optimalen Bedingungen durchgeführt wurde, die nicht zwingend einer realen Umgebung entsprechen muss.

5.5.2 Anzahl der getesteten Samples

Diese Arbeit beschränkt sich auf 15 populäre Ransomware-Samples aus disjunkten Familien. Während diese Anzahl das Testen der grundlegenden Funktionalitäten des FIM ermöglicht und bereits Einschränkungen aufgezeigt hat, könnte eine vielfältigere Anzahl möglicherweise weitere Schwächen aufzeigen, beispielsweise weitere Umgehungstechniken und eine damit verbundene geringere Erkennungsrate.

Im Vergleich dazu umfassen einige der genannten Arbeiten aus Kapitel 3 bis zu tausenden von Samples, allerdings verteilen diese sich oftmals auf eine kleine Menge von Familien. Damit ist die Wahrscheinlichkeit erhöht, dass mehrere ausgewählte Samples einer Familie eine identische Funktionalität aufweisen, jedoch durch die angewendeten Verschleierungsmethoden unterschiedlich aussehen.

5.5.3 Limitationen des File-Integrity-Monitors

Der entwickelte File-Integrity-Monitor weist einige Limitationen auf. Darunter ist dieser nicht für Verzeichnisse mit sich häufig ändernden Dateien ausgelegt, da das System auf einer Baseline basiert, die bei jeder (gutartigen) Änderung aktualisiert werden müsste. Beispielsweise sind unter Windows im AppData-Verzeichnis programmspezifische Dateien enthalten, wobei zahlreiche Anwendungen hochfrequent temporäre Dateien oder Caching-Dateien modifizieren.

Zusätzlich sollte berücksichtigt werden, dass der Benutzer dafür zuständig ist, welche Verzeichnisse überwacht werden. Eine Erkennung anhand der Entropie oder des Dateityps ist nur dann gegeben, wenn die Ransomware die Dateien des zu überwachenden Ordners verschlüsselt. Darüber hinaus sind die Merkmale abhängig von dem Aufbau einer Datei. Wenn beispielsweise ausschließlich Dateien überwacht werden, die bereits verschlüsselt sind und daher eine hohe Entropie besitzen, kann auch in diesem Fall eine Erkennung nur eingeschränkt erfolgen.

Der FIM ist bei einem Zugriff auf eine modifizierte Datei darauf angewiesen zu warten, wenn ein anderer Prozess die Datei liest oder bearbeitet. Theoretisch könnte die Ransomware eine Einsicht in den Dateiinhalt unterbinden, indem sie die Blockierung der Datei so weit hinauszögert, bis der FIM keinen weiteren Versuch zur Erlangung des FileHandle durchführt.

Die verwendete ReadDirectoryChangesW Funktion zum Monitoring von Veränderungen in einem Verzeichnis könnte beispielsweise durch den Einsatz eines API-Hooks umgangen werden (siehe Abschnitt 3.3.2). Im Rahmen der Evaluation waren jedoch keine Anzeichen einer Manipulation zu beobachten.

Ransomwares, die die Dateien mithilfe des File-to-File-Musters verschlüsseln (siehe Abschnitt 3.2.1), unterliegen einer weiteren Limitation in Bezug auf der Berechnung der Entropie und des Dateityps. Der FIM setzt voraus, dass die durch die Ransomware neu erzeugten Dateien denselben Dateinamen wie die Originaldateien tragen. Wenn die Ransomware jedoch zufällige oder verschlüsselte Dateinamen verwendet, ist nicht mehr eindeutig, zu welcher ursprünglichen Datei der verschlüsselte Inhalt gehört, sodass die Ermittlung der korrekten Werte aus der Baseline nicht möglich ist. Alternativ kann in diesen Fällen jedoch eine Überwachung von massenweise Löschungen kritischer Dateien erfolgen.

Eine weitere Einschränkung der Anwendung ist, dass diese nicht vor Crypto-Ransomware warnt, die eine vollständige Verschlüsselung des Datenträgers auf Blockebene durchführt. Ein exemplarisches Beispiel hierfür ist die Ransomware Petya, die das MBR manipuliert, das System neu startet und anstatt das Betriebssystem zu laden, ein Schadprogramm ausführt, welches den Datenträger verschlüsselt (Carlin et al., 2018).

6 Schlussbetrachtung

6.1 Fazit

Ransomwares haben sich schon länger als einer der größten Gefahren in der Bedrohungslandschaft etabliert. Die weitreichende Verschlüsselung von Daten in Kombination mit der Datenexfiltration ist zu einem Standardverfahren der Angreifer geworden, die systemweite Ausfälle hervorrufen und dadurch nicht nur zu wirtschaftlichen Schäden führen, sondern auch Auswirkungen auf die Gesellschaft haben können. Angesichts der Tatsache, dass selbst umfassende Präventionsmaßnahmen einen Ransomware-Angriff nicht immer verhindern können, spielt die Erkennung solcher Angriffe eine maßgebliche Rolle, um durch frühzeitige Maßnahmen den potenziellen Schaden zu reduzieren.

In Kapitel 3 wurden verschiedene Techniken zur Erkennung von Crypto-Ransomware aufgezeigt, die sich in die statische und dynamische Analyse einordnen lassen. Während statische Analysetechniken beispielsweise die in Ransomware-Dateien vorhandenen API-Calls oder Signaturen zur Identifikation nutzen, verwenden dynamische Analysetechniken Informationen wie Netzwerkaktivitäten oder I/O-Zugriffsmuster, um Ransomware zur Laufzeit zu erkennen. Obwohl der große Teil der in Kapitel 3 betrachteten Arbeiten hohe Genauigkeitswerte (F1-Score) von mind. 97 % erreichen, sind sie jeweils mit spezifischen Limitierungen verbunden. Teilweise sind die Techniken anfällig für Verschleierungsmethoden, können auf unterschiedliche Weise umgangen werden oder der Ansatz ist von einem spezifischen Kontext abhängig, wie einem bestimmten Netzwerkprotokoll, einer Datei im PE-Format oder dem Vorhandensein eines Domain-Generation-Algorithm seitens der Ransomware. Techniken, die auf Ebene des Dateisystems operieren, setzen direkt an den zentralen Auswirkungen an, die durch die Verschlüsselung entstehen, und sind damit nicht von einem spezifischen Kontext abhängig.

Der im Rahmen dieser Arbeit entwickelte File-Integrity-Monitor (FIM) ist ein für das Windows-Betriebssystem ausgelegtes Sicherheitstool, das die Integrität von konfigurierten Verzeichnissen vor der unbefugten Verschlüsselung durch Crypto-Ransomware überwacht und den Benutzer bei dieser Art von Integritätsverletzung benachrichtigt. Der FIM ist damit nicht

für die Prävention dieser Angriffe konzipiert, sondern für die Phase, in der das System bereits kompromittiert ist und die kryptographische Verschlüsselung der Dateien gestartet wurde.

Um eine Verschlüsselung durch Ransomware zu erkennen, wird zunächst über die in den konfigurierten Verzeichnissen enthaltenen Dateien eine Baseline erstellt, die als nicht kompromittierte Referenz angenommen wird. In dieser wird der Dateityp und die Entropie in Form des Chi-Square-Wertes sowohl über den gesamten Dateiinhalt als auch über die ersten 256 Bytes ermittelt. Diese Werte werden ebenfalls für die modifizierte Variante einer Datei ermittelt und mit den in der Baseline befindlichen Werten verglichen. Ein entworfenes Bewertungsmodell legt Regeln bei einer Änderung des Dateityps und Grenzwerte für die Entropie fest und bewertet die einzelnen Aktivitäten. Zusätzlich zur Entropie und dem Dateityp als Merkmal wurden statische Decoys in dem Benutzerverzeichnis versteckt, wobei ein modifizierender Zugriff als schädliche Aktivität gewertet wird.

Zu Beginn wurden Anforderungen des FIM definiert, die erfolgreich umgesetzt wurden. Die Umsetzung der Funktionalitäten erfolgte in einer graphischen Benutzeroberfläche, wobei die wesentlichen Funktionen die Folgenden sind:

(1) Der Nutzer kann Verzeichnisse definieren, deren Inhalte überwacht und jegliche Aktivitäten wie das Modifizieren, Hinzufügen oder Löschen von Dateien oder Unterverzeichnissen protokolliert werden, die für den Nutzer in der GUI einsehbar sind
(2) Der Nutzer kann eine Baseline über die in den konfigurierten Verzeichnissen enthaltenen Dateien erstellen, diese aktualisieren und in der GUI einsehen
(3) Der Nutzer kann das Monitoring, bei dem die konfigurierten Verzeichnisse auf Crypto-Ransomware-Aktivitäten überwacht werden, aktivieren oder deaktivieren und wird bei einer Verschlüsselung durch Crypto-Ransomware per E-Mail und Toast-Notification alarmiert

In der Evaluation konnte der FIM eine Verschlüsselung auf Basis des Bewertungsmodells bei 14 von 15 kryptographischen Ransomware-Samples aus verschiedenen Familien erkennen. Die partielle Verschlüsselung wurde als Herausforderung für die Erkennung einer Verschlüsselung mittels der Entropie, wie sie von Autoren benannt wurde, bestätigt. Zwar konnte durch den Entropiewert der ersten 256 Bytes dieser Aspekt kompensiert werden, jedoch zeigten sich in

Einzelfällen weiterhin Einschränkungen, wenn die Verschlüsselung versetzt beginnt oder sehr kleine Abschnitte verschlüsselt werden.

Bei zwei Samples konnte in der Evaluation jeweils beobachtet werden, dass knapp über die Hälfte der Dateien auch nach der Verschlüsselung denselben Dateityp aufweisen, weil die für die Identifikation des Dateityps relevanten Bytes aufgrund der partiellen Verschlüsselung nicht betroffen waren. In dieser Hinsicht bietet es sich an ein Merkmal zu ermitteln, das die partielle Verschlüsselung stärker berücksichtigt.

Die Decoys wurden in dem Benutzerverzeichnis platziert, wobei sich dieser Ort überwiegend als geeignet erwies, da alle bis auf ein Sample die Decoys verschlüsselten. Das abweichende Sample konzentrierte sich ausschließlich auf die Verschlüsselung zweier Verzeichnisse, in denen keine Decoys vorhanden waren.

Auch wenn die Robustheit des FIM bezüglich einer Manipulation durch Ransomware kein zentraler Aspekt dieser Arbeit war, wurde im Zuge der Evaluation bei zwei Samples festgestellt, dass der FIM durch diese manipuliert und in einen Fehlerzustand übergegangen ist. Der FIM fordert während der Ausführung exklusive Zugriffe auf Dateien an und sperrt diese somit für andere Prozesse. Dadurch werden unter anderem Dateien, die zur ordnungsgemäßen Funktionsweise des FIM nötig sind, nicht durch Ransomware verschlüsselt. Beide Samples umgehen dies, indem sie die Anwendung gezielt manipulieren, was zu einem Absturz führt.

Zusammenfassend sind bei der Entropie, dem Dateityp und den Decoys als Merkmal zur Erkennung von Ransomware jeweils die oben genannten individuellen Schwächen zu beobachten, die in der Gesamtheit jedoch eine solide Grundlage liefern, da die Defizite eines Merkmals durch die anderen ausgeglichen werden, sodass der File-Integrity-Monitor einen Beitrag zur Erkennung von Ransomware-Aktivitäten leistet.

6.2 Ausblick

In zukünftigen Arbeiten kann der Aspekt der Robustheit näher aufgegriffen werden. Wie bereits im Abschnitt 3.2.4 beschrieben, bietet Windows beispielsweise die Sicherheitsfunktion namens Protected Process Light (PPL) an, mit der kritische Prozesse vor einer Termination und der Manipulation des Speichers geschützt werden können (Kaspersky, 2018). Die Funktion

wird von verschiedenen Anbietern von Sicherheitssoftware genutzt, darunter auch von Antivirenprogrammen (Korkin, 2021).

Eine Möglichkeit, um die Robustheit des Dateisystemmonitorings zu erhöhen, ist die Migration des Monitorings in den Kernelmodus. Dadurch erfolgt das Abfangen der Dateisystemaktivitäten auf einer niedrigeren Ebene, die für Programme im Benutzermodus transparent und damit schwieriger zu manipulieren sind. Zudem wird auf dieser Ebene kein FileHandle benötigt und es stehen weitere Informationen zur Verfügung, etwa die ID eines Prozesses, der auf eine Datei zugegriffen hat, oder die ID einer Datei. Derartige Informationen sind für den Einsatz von weiteren Detektionstechniken, wie einem I/O-Zugriffsmuster eines Prozesses, äußerst hilfreich, damit diese im Kontext des File-Integrity-Monitors nutzbar gemacht werden können (Ahmed et al., 2021).

Um dem Problem der partiellen Verschlüsselung in Bezug auf der Entropie weiter entgegenzuwirken, können Untersuchungen stattfinden, die sich nicht nur auf das Unterschreiten des Entropiewertes, sondern auch auf das Verhältnis zwischen dem alten und neuen Entropiewert konzentrieren, da auch eine teilweise Verschlüsselung zu einer Erhöhung der Entropie führt.

Die Erzeugung von digitalen Fingerabdrücken auf den statischen Decoys können eine Umgehungsmöglichkeit aus Sicht der Ransomware darstellen, wodurch diese gezielt enttarnt und somit nicht verschlüsselt werden könnten. Als Gegenmaßnahme kann die Implementierung dynamischer Decoys erfolgen, sodass diese an unterschiedlichen Stellen im Dateisystem platziert und auf generativer Weise erzeugt werden (Ganfure et al., 2023).

Da in der Evaluation keine Tests zu False Positives stattgefunden haben, ist es empfehlenswert diesen Aspekt in zukünftigen Arbeiten näher zu untersuchen, um die Zuverlässigkeit des FIM bei gutartigen Änderungen zu bewerten und gegebenenfalls Optimierungen vorzunehmen.

Außerdem wäre es denkbar weitere Features in den File-Integrity-Monitor zu integrieren, etwa das Eindämmen der Ransomware bei einer Detektion, um den Schaden unmittelbar zu begrenzen. Hierfür sollte zunächst untersucht werden, welche Möglichkeiten zur Eindämmung existieren und ob Maßnahmen wie das Terminieren des Ransomware-Prozesses eine wirksame Option darstellen, um den Verschlüsselungsvorgang abzubrechen, oder die Ransomware Mechanismen zur Persistenz anwendet.

Literaturverzeichnis

Aboud, M., & Mariyappn, K. (2021). Investigation of Modern Ransomware Key Generation Methods: A Review. *2022 International Conference on Computer Communication and Informatics (ICCCI)*, 1-5. https://doi.org/10.1109/iccci50826.2021.9402680

Ahmed, M., Kim, H., Camtepe, S., & Nepal, S. (30. 09 2021). Peeler: Profiling Kernel-Level Events to Detect. In *Lecture Notes in Computer Science* (S. 240-260). https://doi.org/10.1007/978-3-030-88418-5_12

Ahmed, Y. A., Koçer, B., Huda, S., Al-Rimy, B. A., & Hassan, M. M. (2020). A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection. *Journal of Network and Computer Applications, 167*, 102753. https://doi.org/10.1016/j.jnca.2020.102753

Almashhadani, A. O., Carlin, D., Kaiiali, M., & Sezer, S. (29. Juli 2022). MFMCNS: a multi-feature and multi-classifier network-based system for ransomworm detection. *Computers & Security, 121*, 102860. https://doi.org/10.1016/j.cose.2022.102860

Al-Rimy, B. A., Maarof, M. A., & Shaid, S. Z. (2019). Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. *Future Generation Computer Systems, 101*, 476-491. https://doi.org/10.1016/j.future.2019.06.005

Alzahrani, A., Alshehri, A., Alshahrani, H., Alharthi, R., Fu, H., Liu, A., & Zhu, Y. (2018). RanDroid: Structural Similarity Approach for Detecting Ransomware Applications in Android Platform. *IEEE International Conference on Electro/Information Technology (EIT)*, 0892-0897. https://doi.org/10.1109/eit.2018.8500161

Andronio, N., & Zanero, S. (2015). HelDroid: Dissecting and Detecting Mobile Ransomware. *Lecture notes in computer science*, 382-404. https://doi.org/10.1007/978-3-319-26362-5_18

Aryal, K., Gupta, M., Abdelsalam, M., Kunwar, P., & Thuraisingham, B. (2024). A survey on Adversarial Attacks for Malware analysis. *IEEE Access, 13*, 428-459. https://doi.org/https://doi.org/10.1109/ACCESS.2024.3519524

Bajpai, P., & Enbody, R. (2020). An Empirical Study of Key Generation in Cryptographic Ransomware. *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 10*, 1-8. https://doi.org/10.1109/cybersecurity49315.2020.9138878

Baker, K. (5. Oktober 2023). *How Does Ransomware Spread? 10 Most Common Infection Methods*. Abgerufen am 2. März 2025 von CrowdStrike. https://www.crowdstrike.com/en-us/cybersecurity-101/ransomware/how-ransomware-spreads/

Baldwin, J., & Dehghantanha, A. (2018). Leveraging Support Vector Machine for Opcode Density Based Detection of Crypto-Ransomware. *Advances in information security*, 107-136. https://doi.org/10.1007/978-3-319-73951-9_6

Beaman, C., Barkworth, A., Akande, T., Hakak, S., & Khan, M. (27. September 2021). Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & Security, 111*, 102490. https://doi.org/https://doi.org/10.1016/j.cose.2021.102490

Becker, L. (13. Juni 2017). *MacRansom und MacSpy: Mac-Malware kommt als Dienstleistung*. Abgerufen am 3. Februar 2025 von Heise. https://www.heise.de/news/MacRansom-und-MacSpy-Mac-Malware-kommt-als-Dienstleistung-3742502.html

Begovic, K., Al-Ali, A., & Malluhi, Q. (16. Juni 2023). Cryptographic ransomware encryption detection: Survey. *Computers & Security, 132*, 103349. https://doi.org/10.1016/j.cose.2023.103349

Berrueta, E., Morato, D., Magana, E., & Izal, M. (2019). A survey on Detection Techniques for Cryptographic Ransomware. *IEEE Access, 7*, 144925-144944. https://doi.org/10.1109/ACCESS.2019.2945839

Berrueta, E., Morato, D., Magaña, E., & Izal, M. (30. Juli 2022). Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic. *Expert Systems with Applications, 209*, 118299. https://doi.org/10.1016/j.eswa.2022.118299

Bisson, D. (19. Januar 2017). *Got Outdated Software? RIG Exploit Kit and Cerber Ransomware Hope You Say 'Yes'*. Abgerufen am 28. Februar 2025 von Tripwire. https://www.tripwire.com/state-of-security/got-outdated-software-rig-exploit-kit-cerber-ransomware-hope-say-yes

Briegleb, V. (13. Mai 2017). *Heise*. Abgerufen am 21. Januar 2025 von https://www.heise.de/news/Ransomware-WannaCry-befaellt-Rechner-der-Deutschen-Bahn-3713426.html

BSI. (2022). *Ransomware Bedrohungslage 2022.* Abgerufen am 14. Februar 2025 von https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Ransomware.pdf?__blob=publicationFile&v=5

BSI. (12. November 2024). *Die Lage der IT-Sicherheit in Deutschland 2024.* Abgerufen am 14. März 2025 von https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2024.pdf?__blob=publicationFile&v=5

BSI. (o. D.). *Ransomware – Fakten und Abwehrstrategien*. Abgerufen am 27. Januar 2025 von Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Cyber-Sicherheitslage/Analysen-und-Prognosen/Ransomware-Angriffe/ransomware-angriffe_node.html

BSI. (o. D.). *Ransomware – Vorsicht vor Erpressersoftware*. Abgerufen am 19. Februar 2025 von Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/dok/10366332

Carlin, D., O'Kane, P., & Sezer, S. (2018). Dynamic Opcode Analysis of Ransomware. *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 1-4. https://doi.org/10.1109/cybersecpods.2018.8560667

Cebere, B. C., Flueren, J. L., SebastiÃ¡n, S., Plohmann, D., & Rossow, C. (29. September 2024). Down to earth! Guidelines for DGA-based Malware Detection. *RAID '24: Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, 147-165. https://doi.org/10.1145/3678890.3678913

Cen, M., Jiang, F., Qin, X., Jiang, Q., & Doss, R. (2023). Ransomware early detection: A survey. *Computer Networks, 239*, 110138. https://doi.org/10.1016/j.comnet.2023.110138

Cesario, N., Lewis, D., Rosales, C., Antolini, F., Stojanovic, R., & Vandenberg, L. (2024). Ransomware Detection Using Opcode Sequences and Machine Learning: A Novel Approach with t-SNE and Support Vector Machines. *TechRxiv*, PrePrint. https://doi.org/10.36227/techrxiv.172963142.20817264/v1

Coglio, F., Lekssays, A., Carminati, B., & Ferrari, E. (2023). Early-Stage ransomware detection based on pre-attack internal API calls. In L. Barolli, *Advanced Information Networking and Applications* (S. 417-429). Springer International Publishing. https://doi.org/https://doi.org/10.1007/978-3-031-28451-9_36

Cusack, G., Michel, O., & Keller, E. (2018). Machine Learning-Based Detection of Ransomware Using SDN. *SDN-NFV Sec'18: Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, 1-6. https://doi.org/10.1145/3180465.3180467

CyberOne. (28. April 2020). *Ransomware and the CIA Triad: Considerations for Evolving Attack Methods*. Abgerufen am 29. Januar 2025 von https://www.cyberonesecurity.com/ransomware-and-the-cia-triad-considerations-for-evolving-attack-methods/

Dargahi, T., Dehghantanha, A., Nikkhah Bahrami, P., Bianchi, G., & Benedetto, L. (7. August 2019). A Cyber-Kill-Chain based taxonomy of crypto-ransomware features. *Journal*

of Computer Virology and Hacking Techniques, 15(4), 277-305. https://doi.org/10.1007/s11416-019-00338-7

Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2022a). Comparison of entropy calculation methods for ransomware encrypted file identification. *Entropy, 24*(10), 1503. https://doi.org/10.3390/e24101503

Davies, S., Macfarlane, R., & Buchanan, W. (2021). Differential area analysis for ransomware attack detection within mixed file datasets. *Computers & Security, 108*, 102377. https://doi.org/10.1016/j.cose.2021.102377

Davies, S., Macfarlane, R., & Buchanan, W. (2022b). NapierOne: A modern mixed file data set alternative to Govdocs1. *Forensic Science International: Digital Investigation, 40*, 301330. https://doi.org/https://doi.org/10.1016/j.fsidi.2021.301330

Deng, X., Cen, M., Jiang, M., & Lu, M. (2023). Ransomware early detection using deep reinforcement learning on portable executable header. *Cluster Computing, 27*(2), 1867-1881. https://doi.org/10.1007/s10586-023-04043-5

Denham, B., & Thompson, D. (2023). Analysis of Decoy Strategies for Detecting Ransomware. In *2023 IEEE Conference on Communications and Network Security (CNS)* (S. 1-6).

Dolesi, K., Steinbach, E., Velasquez, A., Whitaker, L., Baranov, M., & Atherton, L. (2024). A Machine Learning Approach to Ransomware Detection Using Opcode Features and K-Nearest Neighbors on Windows. *TechRxiv*, PrePrint. https://doi.org/10.36227/techrxiv.172926410.04244699/v1

Edwards, S. (30. August 2023). *10 Techniken zur Malware-Erkennung*. Abgerufen am 28. Januar 2025 von CrowdStrike. https://www.crowdstrike.com/de-de/cybersecurity-101/malware/malware-detection/

Enginsight. (o. D.). *File Integrity Monitoring – Daten vor Manipulation schützen*. Abgerufen am 02. Juli 2025 von Enginsight. https://enginsight.com/de/glossar/file-integrity-monitoring/

Ganfure, G. O., Wu, C.-F., Chang, Y.-H., & Shih, W.-K. (2023). RTRAP: Trapping and Containing ransomware with Machine Learning. *IEEE Transactions on Information Forensics and Security, 18*, 1433-1448. https://doi.org/https://doi.org/10.1109/tifs.2023.3240025

Genç, Z. A., Lenzini, G., & Sgandurra, D. (2019). On Deception-Based protection against cryptographic ransomware. In R. Perdisci, C. Maurice, G. Giacinto, & M. Almgren, *Detection of Intrusions and Malware, and Vulnerability Assessment* (S. 219-239). Springer International Publishing. https://doi.org/10.1007/978-3-030-22038-9_11

Genç, Z., Lenzini, G., & Ryan, P. (2018). No Random, no ransom: A key to stop cryptographic ransomware. In *Lecture notes in computer science* (S. 234-255). https://doi.org/10.1007/978-3-319-93411-2_11

Gibert, D., Mateu, C., & Planes, J. (2. Januar 2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications, 153*, 102526. https://doi.org/10.1016/j.jnca.2019.102526

Gómez-Hernández, J., Álvarez-González, L., & García-Teodoro, P. (5. Dezember 2018). R-Locker: Thwarting ransomware action through a honeyfile-based approach. In *Computers & Security* (S. 389-398). https://doi.org/10.1016/j.cose.2017.11.019

Grabmair, M. (04. August 2016). *Neue iOS-Ransomware: So hebeln Sie die Betrüger aus*. Abgerufen am 04. Februar 2025 von MacLife. https://www.maclife.de/news/neue-ios-ransomware-hebeln-betrueger-10080759.html

Hasarat, D. (28. März 2024). *Dispelling the Myth: Is F1-Score Actually Better Than Classification Accuracy?* Abgerufen am 21. April 2025 von IBM Community. https://community.ibm.com/community/user/blogs/danish-hasarat/2024/03/28/dispelling-the-myth-is-f1-score-actually-better-th

horsicq. (23. März 2025). *Program for determining types of files for Windows, Linux and MacOS.* Abgerufen am 23. März 2025 von GitHub. https://github.com/horsicq/Detect-It-Easy

hzqst. (2. Dezember 2022). *VmwareHardenedLoader*. Abgerufen am 22. Februar 2025 von GitHub. https://github.com/hzqst/VmwareHardenedLoader

Imtiaz, K. (März. April 2024). *What is File Integrity Monitoring?* Abgerufen am 2. April 2025 von CrowdStrike. https://www.crowdstrike.com/en-us/cybersecurity-101/exposure-management/file-integrity-monitoring/

Jung, S., & Won, Y. (2018). Ransomware detection method based on context-aware entropy analysis. *Soft Computing, 22*(20), 6731-6740. https://doi.org/10.1007/s00500-018-3257-z

Kaspersky. (19. Juli 2018). *Beschreibung der Technologie Protected Process Light (PPL) für Windows*. Abgerufen am 28. April 2025 von Kaspersky. https://support.kaspersky.com/de/common/windows/13905

Kharraz, A., Arshad, S., Mulliner, C., Robertson, W., & Kirda, E. (10. August 2016). UNVEIL: a large-scale, automated approach to detecting ransomware. *USENIX Security Symposium*, 757-772. https://atc.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_kharraz.pdf

Kivva, A. (26. Februar 2024). *The mobile malware threat landscape in 2023*. Abgerufen am 2. April 2025 von Securelist. https://securelist.com/mobile-malware-report-2023/111964/

Kok, S., Abdullah, A., & Jhanjhi, N. (2020). Early detection of crypto-ransomware using pre-encryption detection algorithm. *Journal of King Saud University - Computer and Information Sciences, 34*(5), 1984-1999. https://doi.org/10.1016/j.jksuci.2020.06.012

Kok, S., Abdullah, A., Jhanjhi, N., & Supramaniam, M. (2019). Prevention of Crypto-Ransomware Using a Pre-Encryption Detection Algorithm. *Computers 2019, 8*(4), 79. https://doi.org/10.3390/computers8040079

Kolodenker, E., Koch, W., Stringhini, G., & Egele, M. (31. März 2017). PayBreak: Defense Against Cryptographic Ransomware. *Proceedings of the 2022 ACM on Asia*

Conference on Computer and Communications Security. https://doi.org/10.1145/3052973.3053035

Korkin, I. (2021). Protected Process Light is not Protected: MemoryRanger Fills The Gap Again. *2021 IEEE Security and Privacy Workshops (SPW)*, 298-308. https://doi.org/10.1109/spw53761.2021.00050

Kosinski, M. (4. Juni 2024). *Was ist Ransomware?* Abgerufen am 20. Januar 2025 von https://www.ibm.com/de-de/topics/ransomware

Kowalczyk, H., Zieliński, P., & Nowak, A. (9. Mai 2024). Dissecting MacOS Ransomware: A Comparative Analysis and Mitigation Strategies. *Research Square (Research Square)*. https://doi.org/10.21203/rs.3.rs-4385485/v1

Lemmou, Y., Lanet, J.-L., & Souidi, E. M. (30. Dezember 2020). A behavioural in-depth analysis of ransomware infection. *IET Information Security, 15*(1), 38-58. https://doi.org/10.1049/ise2.12004

Lenaerts-Bergmans, B. (19. Juli 2023). *Command and Control (C&C) Attacks Explained*. Abgerufen am 24. Februar 2025 von CrowdStrike. https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/command-and-control-cac-attack/

Liska, A., & Gallo, T. (2016). *Defending Against Digital Extortion.*

MacColl, J., Hüsch, P., Mott, G., Sullivan, J., Nurse, J., Turner, S., & Pattnaik, N. (2024). The scourge of ransomware: Victim insights on harms to individuals, organisations and society. *Royal United Services Institute for Defence and Security Studies (RUSI) Occasional Paper*. https://static.rusi.org/ransomware-harms-op-january-2024.pdf

Malik, P., Nautiyal, L., & Ram, M. (2022). *Machine Learning for Cyber Security.* De Gruyter.

MalwareBazaar. (o. D.). Abgerufen am 24. März 2025 von https://bazaar.abuse.ch/

Malwarebytes. (o. D.). *Malvertising*. Abgerufen am 26. Februar 2025 von Malwarebytes. https://www.malwarebytes.com/de/malvertising

Manavi, F., & Hamzeh, A. (2020). A New Method for Ransomware Detection Based on PE Header Using Convolutional Neural Networks. *17th International ISC Conference on*

Information Security and Cryptology (ISCISC), 82-87. https://doi.org/10.1109/iscisc51277.2020.9261903

McIntosh, T., Susnjak, T., Liu, T., Xu, D., Watters, P., Liu, D., Hao, Y., Ng, A., & Halgamuge, M. (2024). Ransomware Reloaded: Re-examining its trend, research and mitigation in the era of data exfiltration. *ACM Computing Surveys*. https://doi.org/10.1145/3691340

Medhat, M., Gaber, S., & Abdelbaki, N. (2018). A New Static-Based Framework for Ransomware Detection. *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 710-715. https://doi.org/10.1109/dasc/picom/datacom/cyberscitec.2018.00124

Microsoft. (3. November 2022). *Stopping C2 communications in human-operated ransomware through network protection*. Abgerufen am 6. März 2025 von Microsoft Threat Intelligence. https://www.microsoft.com/en-us/security/blog/2022/11/03/stopping-c2-communications-in-human-operated-ransomware-through-network-protection/

Microsoft. (27. April 2024). *TemplateStudio*. GitHub. https://github.com/microsoft/TemplateStudio

Modi, J., Traore, I., Ghaleb, A., Ganame, K., & Ahmed, S. (2020). Detecting ransomware in encrypted web traffic. In A. Benzekri, M. Barbeau, G. Gong, R. Laborde, & J. Garcia-Alfaro, *Foundations and Practice of Security* (S. 345-353). Springer International Publishing. https://doi.org/10.1007/978-3-030-45371-8_22

Moore, C. (2016). Detecting Ransomware with Honeypot Techniques. *2016 Cybersecurity and Cyberforensics Conference (CCC)*, 77-81. https://doi.org/10.1109/ccc.2016.14

Neprash, H., McGlave, C., Cross, D., Virnig, B., Puskarich, M., Huling, J., Rozenshtein, A., & Nikpay, S. (29. Dezember 2022). Trends in ransomware attacks on US hospitals, clinics, and other health care delivery organizations, 2016-2021. *JAMA Health Forum, 3*(12), e224873. https://doi.org/10.1001/jamahealthforum.2022.4873

Oelmaier, F., Knebelsberger, U., & Naefe, A. (2023). *Krisenfall Ransomware: Strategien für Wiederaufbau, Forensik und Kommunikation.* SpringerNature. https://doi.org/10.1007/978-3-658-41614-0

Oz, H., Aris, A., Levi, A., & Uluagac, A. (31. Januar 2022). A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions. *54*(11s), 1-37. https://doi.org/10.1145/3514229

PE Format. (20. Februar 2025). Abgerufen am 15. März 2025 von Microsoft. https://learn.microsoft.com/en-us/windows/win32/debug/pe-format

Ploszek, R., Švec, P., & Debnár, P. (2021). Analysis of encryption schemes in modern ransomware. *Rad Hrvatske akademije znanosti i umjetnosti Matematičke znanosti, 25(60)*, 1-13. https://doi.org/10.21857/mnlqgc58gy

Pontello, M. (01. Juni 2025). *TrID - File Identifier*. Abgerufen am 3. Juni 2025 von Marco Pontello's Home Page. https://mark0.net/soft-trid-e.html

Pornasdoro, A. (17. Dezember 2014). *Your Browser is (not) Locked*. Abgerufen am 18. März 2025 von Microsoft. https://www.microsoft.com/en-us/security/blog/2014/12/17/your-browser-is-not-locked/

Rezaei, T., Manavi, F., & Hamzeh, A. (2021). A PE header-based method for malware detection using clustering and deep embedding techniques. *Journal of Information Security and Applications, 60*, 102876. https://doi.org/10.1016/j.jisa.2021.102876

Robles, S., Alimboyao, A., Santos, J., Policarpio, M., Morales, N., & Chavez, I. (10. November 2023). *Cerber Ransomware Exploits Atlassian Confluence Vulnerability CVE-2023-22518*. Abgerufen am 1. März 2025 von TrendMicro. https://www.trendmicro.com/de_de/research/23/k/cerber-ransomware-exploits-cve-2023-22518.html

Salehi, S., Shahriari, H., Ahmadian, M. M., & Tazik, L. (2018). A Novel Approach for Detecting DGA-based Ransomwares. *2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, 1-7. https://doi.org/10.1109/iscisc.2018.8546941

Scaife, N., Carter, H., Traynor, P., & Butler, K. (2016). CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 303-312. https://doi.org/10.1109/ICDCS.2016.46

SentinelOne. (12. August 2021). *What Is A Malware Signature and How Does It Work?* Abgerufen am 14. Februar 2025 von SentinelOne. https://www.sentinelone.com/blog/what-is-a-malware-file-signature-and-how-does-it-work/

SentinelOne. (o. D.). *Cerber Ransomware: In-Depth Analysis, Detection, and Mitigation*. Abgerufen am 7. Februar 2025 von https://www.sentinelone.com/anthology/cerber-ransomware/

Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., Goldstein, T. A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., & Taylor, G. (2019). Adversarial training for free! *Advances in neural information processing systems, 32*. https://doi.org/10.48550/arxiv.1904.12843

Shaukat, S., & Ribeiro, V. (2018). RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning. *10th International Conference on Communication Systems & Networks (COMSNETS)*, 356-363. https://doi.org/10.1109/COMSNETS.2018.8328219

Sheen, S., & Yadav, A. (2018). Ransomware detection by mining API call usage. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. https://doi.org/10.1109/icacci.2018.8554938

Sheen, S., Asmitha, K., & Venkatesan, S. (2022). R-Sentry: Deception based ransomware detection using file access patterns. *Computers & Electrical Engineering, 103*, 108346. https://doi.org/10.1016/j.compeleceng.2022.108346

Shishkova, T., & Kivva, A. (21. Februar 2022). *Mobile malware evolution 2021*. Abgerufen am 15. April 2025 von Securelist. https://securelist.com/mobile-malware-evolution-2021/105876/

Sihwail, R., Omar, K., & Ariffin, K. (30. September 2018). A survey on malware analysis techniques: static, dynamic, hybrid and memory analysis. *International Journal on Advanced Science Engineering and Information Technology, 8*(4-2), 1662-1671. https://doi.org/10.18517/ijaseit.8.4-2.6827

Sikorski, M., & Honig, A. (2012). *Practical malware analysis: The Hands-On Guide to Dissecting Malicious Software.* No Starch Press.

Sophos. (22. April 2024). *https://assets.sophos.com/X24WTUEQ/at/9brgj5n44hqvgsp5f5bqcps/sophos-state-of-ransomware-2024-wp.pdf.* Abgerufen am 10. Februar 2025 von https://assets.sophos.com/X24WTUEQ/at/9brgj5n44hqvgsp5f5bqcps/sophos-state-of-ransomware-2024-wp.pdf

Sophos. (30. April 2024). *Root causes of ransomware attacks in organizations worldwide as of February 2024*. Abgerufen am 27. Februar 2025 von Statista. https://www.statista.com/statistics/1410445/cause-ransomware-attacks-global/

Starke, G., & Hruschka , P. (o. D.). *arc42*. Abgerufen am 2. Juli 2025 von https://docs.arc42.org/

Tanenbaum, A., & Bos, H. (2016). *Moderne Betriebssysteme.* Pearson Studium.

Tang, F., Ma, B., Li, J., Zhang, F., Su, J., & Ma, J. (2020). RansomSpector: An introspection-based approach to detect crypto ransomware. *Computers & Security, 97*, 101997. https://doi.org/10.1016/j.cose.2020.101997

Trend Micro. (6. März 2024). *Operating systems with the highest share of ransomware attacks detected worldwide from 2019 to 2023*. Abgerufen am 2. Februar 2025 von Statista. https://www.statista.com/statistics/1498850/most-targeted-operating-systems-with-ransomware/

Van Boven, L. S., Kusters, R. W., Tin, D., Rao, M., Dameff, C., & Barten, D. G. (15. Juni 2023). Hacking Acute Care: A qualitative study on the health care impacts of ransomware attacks against hospitals. *Annals of Emergency Medicine, 83*(1), 46-56. https://doi.org/10.1016/j.annemergmed.2023.04.025

Verfürden, M., & Tyborski, R. (07. November 2022). *Handelsblatt*. Abgerufen am 21. Januar 2025 von https://www.handelsblatt.com/unternehmen/industrie/autozulieferer-cyberangriff-auf-continental-ransomware-gruppe-erbeutet-offenbar-40-terabyte-daten/28792818.html

VirusShare. (o. D.). Abgerufen am 24. März 2025 von https://virusshare.com/

VirusTotal. (9. April 2025). *GitHub*. Abgerufen am 18. April 2025 von The pattern matching swiss knife. https://github.com/VirusTotal/yara

Williams, D. (17. Februar 2024). *Data Exfiltration – What you Need to Know*. Abgerufen am 21. Januar 2025 von BlackFog. https://www.blackfog.com/what-you-need-to-know-about-data-exfiltration/

Zalesskiy, J. (28. März 2024). *Basic Malware Packers: What are They and How to Analyze Them in ANY.RUN*. Abgerufen am 24. März 2025 von Any.Run. https://any.run/cybersecurity-blog/malware-packers-explained/

Zhang, B., Xiao, W., Xiao, X., Sangaiah, A. K., Zhang, W., & Zhang, J. (2019). Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes. *Future Generation Computer Systems, 110*, 708-720. https://doi.org/10.1016/j.future.2019.09.025

Zlatkovski, D., Mileva, A., Bogatinova, K., & Ampov, I. (2018). A New Real-Time File Integrity Monitoring System for Windows-based Environments. *ICT Innovations 2018.* https://www.researchgate.net/publication/331258764_A_New_Real-Time_File_Integrity_Monitoring_System_for_Windows-based_Environments

A Anhang

Die dieser Arbeit beiliegenden CD enthält die folgenden ergänzenden Materialien:

(1) Die vorliegende Arbeit als PDF-Dokument
(2) Quellcode des File-Integrity-Monitors
(3) Daten zur Bestimmung des Grenzwertes der Chi-Square-Werte
(4) Ergebnisse der Evaluation des File-Integrity-Monitors
(5) Auflistung der verwendeten Samples mit relevanten Details

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	[redacted]
Ort	Datum	Unterschrift im Original