



universität  
wien

# MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Sicherung von Angriffsspuren bei Ransomwareangriffen  
auf KMUs“

verfasst von / submitted by

Louis Christ

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of  
Master of Science (MSc)

Wien, 2022 / Vienna 2022

Studienkennzahl lt. Studienblatt /  
degree programme code as it appears on  
the student record sheet:

A 066 926

Studienrichtung lt. Studienblatt /  
degree programme as it appears on  
the student record sheet:

Masterstudium Wirtschaftsinformatik UG2002

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. Dr. Gerald Quirchmayr

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
1.1	Ransomware . . . . .	6
1.2	Digitale Forensik . . . . .	6
<b>2</b>	<b>State of the Art in Literatur und Praxis</b>	<b>9</b>
2.1	Literaturanalyse - Status Quo der Analyse von Ransomware-Angriffen . . . . .	9
2.1.1	Forensic Process . . . . .	9
2.1.2	Forensic Technology . . . . .	12
2.1.3	Live Response . . . . .	12
2.1.4	Post Mortem Analyse . . . . .	13
2.1.5	Sleuth Kit . . . . .	13
2.1.6	Autopsy - Digital Forensic . . . . .	14
2.2	Auswertung und Aufbereitung . . . . .	14
2.3	Implikationen für das geplante Modell . . . . .	15
<b>3</b>	<b>Methodischer Ansatz</b>	<b>17</b>
3.1	Leitlinie 1: Design als Artefakt . . . . .	18
3.2	Leitlinie 2: Problemrelevanz . . . . .	19
3.3	Leitlinie 3: Bewertung des Designs . . . . .	20
3.3.1	Usability . . . . .	21
3.3.2	Funktional . . . . .	21
3.4	Leitlinie 4: Beiträge zur Forschung . . . . .	23
3.4.1	Das Design-Artefakt . . . . .	23
3.4.2	Grundlagen . . . . .	23
3.4.3	Bewertung des Designs . . . . .	23
3.5	Leitlinie 5: Strenge der Forschung . . . . .	23
3.6	Leitlinie 6: Design als Suchprozess . . . . .	24
3.7	Leitlinie 7: Kommunikation der Forschung . . . . .	25
<b>4</b>	<b>Vorgehensmodell zur Ransomware-Analyse</b>	<b>26</b>
4.1	BPMN - Business Process Modeling Notation . . . . .	26
4.2	Das Vorgehensmodell im Detail . . . . .	27
4.3	Umsetzbarkeit in Code . . . . .	30
<b>5</b>	<b>Prototypische Implementierung des Ransomware-Analyse Tools</b>	<b>31</b>
5.1	Technology Stack . . . . .	31
5.1.1	NextJS . . . . .	32
5.1.2	Prisma . . . . .	32
5.1.3	MongoDB . . . . .	35
5.2	Systemarchitektur des Prototyps . . . . .	35
5.3	Implementierung der automatischen Analyse . . . . .	37
5.4	Implementierung einer API-Schnittstelle . . . . .	41

<b>6</b>	<b>Test des Prototypen</b>	<b>43</b>
6.1	Usability Tests . . . . .	43
6.1.1	Kann die Analyse von einem Menschen mit guten IT-Kenntnissen durchgeführt werden? . . . . .	43
6.1.2	Wie wird die Benutzerfreundlichkeit eingeschätzt? . . . . .	44
6.1.3	Eignet sich der Prototyp für kleine und mittelständische Unternehmen? . . . . .	44
6.1.4	Ist der Prototyp auf unterschiedlichen Betriebssystemen lauffähig? . . . . .	44
6.1.5	Welche Bereiche kann das System? . . . . .	45
6.2	Hypothesenbasierte funktionale Tests . . . . .	45
6.2.1	Welcher Typ von Ransomware wird bei der automatischen Analyse erkannt? . . . . .	45
6.2.2	Wie hoch ist die Trefferquote bei der automatischen Analyse? . . . . .	46
6.2.3	Ist der Prototyp für die Zukunft erweiterbar? . . . . .	47
6.2.4	Wie hoch ist die algorithmische Komplexität? . . . . .	48
6.3	System Limitations . . . . .	49
6.3.1	Direkte Abhängigkeit der Storage Complexity vom Umfang und Struktur der eingegebenen Daten . . . . .	49
6.3.2	Grenzen der Detektion . . . . .	49
<b>7</b>	<b>Diskussion der Ergebnisse</b>	<b>51</b>
7.1	Leitlinie 1: Design als Artefakt . . . . .	51
7.2	Leitlinie 2: Problemrelevanz . . . . .	51
7.3	Leitlinie 3: Bewertung des Designs . . . . .	53
7.4	Leitlinie 4: Beiträge zur Forschung . . . . .	53
7.5	Leitlinie 5: Strenge der Forschung . . . . .	53
7.6	Leitlinie 6: Design als Suchprozess . . . . .	53
7.7	Leitlinie 7: Kommunikation der Forschung . . . . .	54
<b>8</b>	<b>Conclusio</b>	<b>55</b>
<b>A</b>	<b>Anhang</b>	<b>59</b>
A.1	User Information . . . . .	59
A.2	Installation Guide . . . . .	59

## Abbildungsverzeichnis

1	SAP-Modell . . . . .	11
2	Einfluss von Menschen in den verschiedenen Phasen einer forensischen Untersuchung [18] . . . . .	11
3	Der Generate und Test Zyklus nach [24] . . . . .	25
4	BPMN 1.2 Elemente [17] . . . . .	27
5	Vorgehensmodell . . . . .	29

6	Komponenten Diagramm . . . . .	31
7	Prisma im Vergleich zu Alternativen [14] . . . . .	33
8	Prisma Migrate Workflow [13] . . . . .	34
9	Datenbankeintrag der WannaCry-Ransomware . . . . .	35
10	Systemarchitektur des Prototyps aus technischer Sicht . . . . .	36
11	Systemarchitektur des Prototyps aus funktionaler Sicht . . . . .	37
12	Ablauf der automatischen Analyse . . . . .	39
13	Laufzeitanalyse . . . . .	49
14	Ergebnis Analyse mit <i>.encrypted</i> . . . . .	50
15	QR-Code Vercel . . . . .	59
16	QR-Code GitHub . . . . .	59
17	QR-Code Node.js . . . . .	60

## Tabellenverzeichnis

1	Design-Science Leitlinien [24] . . . . .	18
2	Design Evaluierungsmethoden [24] . . . . .	22
3	Betriebssysteme & Browser . . . . .	45
4	Getestete Ransomware . . . . .	46
5	Laufzeitanalyse . . . . .	48
6	Zielkriterien . . . . .	52
7	Beschränkungen . . . . .	52

## Glossar

**API** Ein Application Programming Interface stellt einem anderen Programm eine Schnittstelle bereit um Daten auszutauschen. 7, 8, 31, 32, 37, 41, 47

**APT** Advanced Persistent Threat ist ein zielgerichteter Angriff der häufig darauf abzielt Daten aus dem Zielsystem zu stehlen [29]. 48

**BPMN** Business Process Model and Notation. 26, 27, 53

**CSV** Comma-separated value, eine Datei bei der die Werte durch Kommas getrennt sind. 38, 45, 46, 48–50

**DDoS** Distributed Denial of Service ist ein Angriff bei dem viele kompromittierte Hosts Pakete an ein Opfer senden um es lahmzulegen [25]. 48

**HTML** Hypertext Markup Language wird verwendet um Websites im Internet zu entwickeln. 12, 13, 32

**JSON** JavaScript Object Notation ist ein Dateiformat zum Datenaustausch. 31, 35, 42

**KMU** Klein- und Mittelbetriebe. 19, 51, 55

**MIME** Gibt an, um welchen Medientyp es sich bei einer Datei handelt [28]. 14

**NIST** National Institute of Standards and Technology. 13, 14

**ORM** Object-Relation Mapper ist eine Technik, mit der ein objektorientiertes Programm seine Objekte in einer Datenbank ablegen kann [34]. 32

**PDF** Portable Document Format. 9

**PHP** Hypertext Preprocessor ist eine Open Source-Skriptsprache zum erstellen von Webanwendungen [10]. 32, 35, 36

**RAM** Random-Access Memory. 12, 16, 52

**SQL** Structured Query Language. 31, 32, 35, 36

Die im Bereich der digitalen Forensik verwendete Terminologie stützt sich auf folgende Quelle [15]. Diese Zusammenstellung dient z.B. auch als Grundlage für die Basisausbildung im Studiengang Cybersecurity der Universität der Bundeswehr in München.

# 1 Einleitung

Die fortschreitende Digitalisierung im privaten Sektor sowie in der Industrie hat nicht nur Vorteile. Durch sie nehmen auch Cyberangriffe jedes Jahr zu [1], wodurch die Digitale Forensik immer mehr an Bedeutung gewinnt. Immer mehr Daten werden nicht mehr in Papierform abgelegt sondern auf digitalen Datenträgern, somit steigt auch die Abhängigkeit von diesen Systemen. Werden die Daten durch einen Ransomware-Angriff verschlüsselt und damit für den User unbrauchbar gemacht, ist das im privaten Sektor sehr ärgerlich. In der Industrie hingegen kann ein solcher Angriff Fertigungsstraßen oder sogar ganze Unternehmen arbeitsunfähig machen und wertvolle Daten vernichten.[30]

## 1.1 Ransomware

Ransomware ist eine Art Datendiebstahl, bei dem die Daten auf dem infizierten System verschlüsselt und damit unbrauchbar gemacht werden. Sie gelangt meistens durch eine E-Mail auf das System des Opfers, indem ein Anhang oder ein Link zu einer Website versteckt sind. Ist die Ransomware auf dem System, beginnt sie die Daten des Nutzers zu verschlüsseln und fordert anschließend ein Lösegeld. Dieses wird meist in einer Kryptowährung gefordert, damit der Täter unerkannt bleibt. Im schlimmsten Fall verbreitet sich die Ransomware durch das Netzwerk und verschlüsselt noch andere Systeme. Es kann bei der Verschlüsselung um jegliche Arten von Benutzerdaten gehen. Angefangen bei Dokumenten oder Musik bis hin zu sensiblen und wichtigen Daten, die Unternehmen zum Arbeiten benötigen. So unter anderem wichtige Patientendaten in einem Krankenhaus oder die Konstruktionspläne eines Architekten. Eine Ransomware ist schwer wieder zu entfernen. Selbst wenn das Lösegeld gezahlt wird, gibt es keine Garantie, dass die Daten auch wirklich wieder entschlüsselt werden. Einer der bekanntesten Angriffe war WannaCry im Jahr 2017, welche Krankenhäuser, Universitäten, Unternehmen und Regierungsorganisationen angriff und ein Lösegeld forderte. Insgesamt hatte WannaCry mehr als 2.000.000 Opfer weltweit.[31]

Um sich vor Ransomware zu schützen sollte ein Viren-Scanner auf dem Computer installiert sein und auf dem aktuellen Stand gehalten werden. Es sollten niemals Links geöffnet werden, die unseriös sind. Es ist sinnvoll, in regelmäßigen Abständen Backups durchzuführen um im Falle eines Angriffs möglichst wenig Daten zu verlieren. Weiterhin kann der Zugriff auf Programme beschränkt werden, sodass keine gefährliche Software auf dem Rechner ausgeführt wird. E-Mails können vom Mail-Server vorher gescannt und eine Firewall kann eingerichtet werden.[31]

## 1.2 Digitale Forensik

Digitale Forensik beschreibt den Wissenschaftlichen Prozess, der sich mit der Analyse der Digitalen Spuren in IT-Systemen beschäftigt. Ziel dabei ist es, die

Täter eines Cyberangriffes festzustellen, indem digitale Beweismittel gesammelt und durch die richtigen Analyseschritte aufbereitet werden. Damit können sie anschließend vor Gericht verwenden zu können. [30] In dieser Arbeit soll es gezielt um Ransomware-Forensik gehen.

Digitale Forensik ist ein sehr komplexes Thema und benötigt in der Regel einen IT-Forensiker mit ausreichend Erfahrung, um eine fundierte Analyse durchzuführen. Für kleine und mittelständische Unternehmen stellt es oft ein personales und finanzielles Problem dar, eine Analyse nach einem Angriff durchzuführen. [30]

Egal wie der Betroffene beeinträchtigt wurde, er stellt sich unweigerlich die Frage, wie der Angriff abgelaufen ist. Diese Erkenntnisse sind nicht nur für das Opfer von Interesse, auch für zukünftige Angriffe sind Daten über den Ablauf des Angriffs von essentieller Bedeutung. Sie können dabei helfen, können präventiv gegen ähnliche Angriffe vorgehen, indem Lücken aufgedeckt und diese anschließend in der Sicherheitsinfrastruktur geschlossen werden. Weiterhin können bei einer Untersuchung auch Sicherheitsschwachstellen gefunden werden, die nicht mit dem Angriff in Verbindung stehen. Diese können dann aktiv behoben werden um einen weiteren Angriff zu vermeiden.[11] Im besten Fall können sie dabei behilflich sein, die verschlüsselten Daten wiederherzustellen.[12]

Dieses Problem soll im Rahmen dieser Arbeit angegangen werden. Es soll ein Tool entwickelt werden, welches dabei hilft, folgende Fragen zu beantworten:

- Wie wurde der Angriff gestartet?
- Wie ist der Angreifer auf das System gekommen?
- Welche Dateien sind betroffen?
- Wann hat der Angriff stattgefunden?

Das entwickelte Tool soll bei der Analyse unterstützen, diese jedoch nicht selbst durchführen. Zur Umsetzung der Arbeit wird das Tool Autopsy - Digital Forensics als Basis verwendet, da es eines der bekanntesten kostenlosen Tools zur Analyse einer Festplatte oder eines Festplatten-Images nach einem Angriff ist. Es scannt den Datenträger und liefert Informationen, wie zum Beispiel, ob sich die Dateieigenschaften von Dateien verändert haben oder wann eine Datei verändert worden ist wurde. Weiterhin werden E-Mails und Bilder gescannt. Zur Analyse des Systems wird es heruntergefahren, eine Kopie des Datenträgers erstellt und anschließend wird diese mit Autopsy unter Anleitung des entwickelten Tools durchgeführt.

Entwickelt wird das Tool mit dem React-Framework NextJS. Der größte Vorteil von NextJS, ist die Möglichkeit mit einem Framework sowohl Frontend als auch Backend zu entwickeln. NextJS unterstützt nicht nur das Erstellen von Routen zum Frontend sondern auch zu APIs im Backend. Jede Datei, die im *api*-Ordner abgelegt wird, wird automatisch ein API-Endpoint.[4] So kann

z.B. eine API-Schnittstelle erstellt werden um die Datenbank der Ransomware-Typen mit weiteren Einträgen zu befüllen, wodurch das Tool auch in Zukunft einfach erweitert werden kann. NextJS kann durch Prisma erweitert werden, welches die Kommunikation mit der Datenbank übernimmt. Es ermöglicht dem Entwickler Abfragen zu schreiben, die *Type-Safe* sind und gibt in einer Prisma Schema Datei die Objekte vor, welche dann zu Datenbank-Einträgen werden. [14] Als Datenbank soll die MongoDB Cloud zum Einsatz kommen, in der die verschiedenen Ransomware-Typen gespeichert werden. Gehostet wird NextJS durch den Anbieter Vercel.

## 2 State of the Art in Literatur und Praxis

Hat es einen Ransomware-Angriff gegeben, stellt sich häufig die Frage, wie, wo und wann ein solcher Angriff stattgefunden hat. Weiterhin von Bedeutung ist auch, wer den Angriff durchgeführt hat und was dagegen getan werden kann, dass sich ein solcher nicht wiederholt. Diese Informationen sind zum einen nützlich um in Zukunft weitere solcher Angriffe entgegen zu wirken, zum anderen werden die forensischen Daten vor Gericht benötigt um den Täter ausfindig zu machen, anzuklagen und Schadensersatzforderungen geltend zu machen.[2] Wobei hier zu beachten ist, dass digitale Forensik aufgrund fehlender personeller und finanzieller Mittel häufig nicht eingesetzt wird.[30] Doch was sind überhaupt digitale Spuren?

”Digitale Spuren sind alle Daten, die durch die Benutzung von Computern gespeichert oder übertragen werden und die die Theorie über einen Tathergang unterstützen oder ablehnen, oder die ein kritisches Element, wie ein Motiv oder Alibi, betreffen.” [30]

Diese digitalen Spuren sind dabei weniger offensichtlich als bei einem Einbruch in ein Haus, wo es sich beispielsweise um ein zerbrochenes Fenster handeln kann. Bei einem digitalen Angriff kann es sich um Dateien, Dokumente, wie z.B. ein PDF, E-Mails, Bilder, Videos bis hin zu einzelnen Bits handeln, die der Täter genutzt hat, um in das System einzudringen. [30]

### 2.1 Literaturanalyse - Status Quo der Analyse von Ransomware-Angriffen

Dieses Kapitel befasst sich mit dem aktuellen Status der Digitalen Forensik in Bezug auf Ransomware-Angriffen in der Literatur. Dazu wird im folgenden Abschnitt ein Überblick über die daraus gewonnenen Erkenntnisse gegeben. Zunächst wird aufgezeigt, was ein Forensischer Prozess ist, wie dieser abläuft und letztendlich eine Auswertung der gefundenen Ergebnisse.

In den Unterpunkten 2.1.1 bis 2.1.4 werden grundlegende Vorgehensweisen und Prozesse beschrieben, exemplarisch ausgewählte Tools werden in 2.1.5 und 2.1.6 dargestellt, soweit diese als für die Masterarbeit besonders relevant eingeschätzt werden.

#### 2.1.1 Forensic Process

Bei digitaler Forensik handelt es sich nicht um eine einzelne Aufgabe oder einen einzelnen Prozess, es geht viel mehr um eine Gruppe von Aufgaben und Prozessen. Diese sind nach [19] oft durch die Implementierungsdetails definiert.

Weiterhin unterscheidet [19] zwischen Untersuchungsverfahren, die von traditionellen Forensikern entwickelt worden sind und jenen, die von Technologen entwickelt wurden. Die von den traditionellen Forensikern entwickelten Verfahren konzentrieren sich dabei verstärkt auf die Handhabung der Beweise, wobei die von den Technologen entwickelten Verfahren sich stärker mit den technischen Details der Beweiserfassung beschäftigen.[19] Pollit und Whiteledge [32] definieren den Forensischen Prozess wie folgt:

”Digitale Forensik ist die Wissenschaft der Sammlung, Erhaltung, Untersuchung, Analyse und Präsentation relevanter digitaler Beweise zur Verwendung in Gerichtsverfahren”.

Die digitale Forensik kann zur Aufklärung der Fragen nach dem Was, Wo, Wann und Wie beitragen. Weiterhin kann aufgedeckt werden, was genau auf dem System geschehen ist und welche Werkzeuge verwendet wurden, insbesondere, was gegen eine Wiederholung der Tat getan werden kann. [21] Generell gibt es sechs allgemeingültige Anforderungen an einen forensischen Prozess, diese sind besonders für Juristen von hoher Relevanz. Die sechs Anforderungen lauten wie folgt:

- Akzeptanz
- Glaubwürdigkeit
- Wiederholbarkeit
- Integrität
- Ursache und Auswirkungen
- Dokumentation

Akzeptanz bedeutet in diesem Fall, dass die verwendeten Methoden und Schritte in der Fachwelt beschrieben und allgemein akzeptiert sind. Generell können auch neue Methoden verwendet werden, diese sollten jedoch nachgewiesen korrekt sein. Weiterhin muss die Glaubwürdigkeit der Methoden gegeben und die Funktionalität muss somit nachweisbar sein. Die eingesetzten Hilfsmittel müssen bei erneuter Anwendung oder der Anwendung Dritter immer wieder das gleiche Ergebnis liefern und die Integrität der Spuren muss gewahrt sein. Die Beweise dürfen durch die Untersuchung also nicht verändert werden. Die Auswahl der Methoden muss am Ende der Untersuchung dazu führen, dass nachvollziehbare Verbindungen zwischen Ereignissen und Beweisen entstehen. Abschließend muss jeder Ermittlungsschritt dokumentierbar sein.[22] Für die Untersuchung eignet sich ganz besonders Open-Source-Software, wie z.B. das Tool Autopsy - Digital Forensics”, welches auch im Rahmen dieser Masterarbeit verwendet wird. Hierbei kann die korrekte Funktionsweise zusätzlich im Quellcode überprüft werden. [21]

Nachdem ein Vorfall entdeckt worden ist, kann der folgende Ablauf nach Dole [21] in drei Phasen unterteilt werden. Diese drei Phasen werden durch das SAP-Modell beschrieben, welches in Grafik 1 zu sehen ist und aus den Phasen Secure, Analyse und Present besteht.

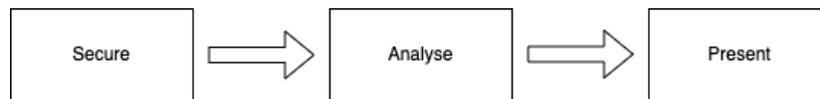


Abbildung 1: SAP-Modell

In der ersten Phase, der Sicherungsphase, werden zunächst alle Beweise gesichert. Wenn nämlich Informationen in dieser Phase nicht erfasst werden, sind sie in der Analysephase verloren. Von großer Bedeutung für die Ermittler ist, wer mit den Beweisen in Kontakt gekommen ist, da jeder Kontakt mit den Beweisen diese verändern kann. In 2 wird dargestellt, in welchen Phasen der Untersuchung welche Menschen mit den Beweisen in Kontakt kommen. Darunter sind zum Beispiel das Opfer, der Ersthelfer, die Ermittler oder Polizisten.[18]

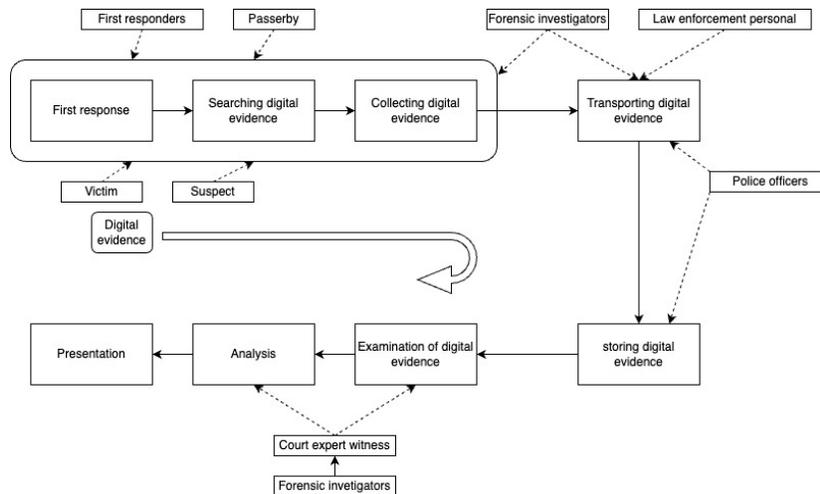


Abbildung 2: Einfluss von Menschen in den verschiedenen Phasen einer forensischen Untersuchung [18]

Weiterhin sollte die Integrität der Daten gewahrt bleiben. Dies kann durch die Verwendung von Hashwerten oder das Vier-Augen-Prinzip geschehen. In Phase zwei werden die gesicherten Daten anschließend ausgewertet.[21] Diese Auswertung lässt sich nach Meier [30] nochmal in 4 Phasen unterteilen, beginnend mit der Identifikation, gefolgt von Individualisierung, Assoziation und Rekonstruktion. Die Identifikation versucht die Frage zu beantworten: **Was ist es?** und bezeichnet nach Meier [30] die Wahrnehmung einer Spur als solche. Als Ausgangspunkt für die Identifikation dient beispielsweise eine Festplatte. Ei-

nem Experten ist es durch sein Wissen möglich, eine Hypothese aufzustellen, was genau passiert ist. Diese Hypothese kann lauten: Ein Computer hat eine bestimmte Website aufgerufen und dabei einen bestimmten Browser verwendet. Der Ermittler kann also den Cache des Browsers auf der Festplatte ausfindig machen und diesen als Spur identifizieren. Im nächsten Schritt, der Individualisierung, kann der Ermittler mit Hilfe eines Werkzeugs den Cache auswerten und sich die Eigenschaften der Dateien zu den Inhalten der besuchten Website ansehen. Hat der Ermittler genug individuelle Eigenschaften identifiziert, folgt die Assoziation. Hierbei kann er durch die individuellen Eigenschaften wie z.B. HTML-Dateien oder Mediendateien, die auf der Website vorkommen, seine Hypothese annehmen und somit auf einen Kontakt zwischen dem Computer und der Website schließen. Abschließend folgt der Schritt der Rekonstruktion. Hierbei ordnet der Ermittler die Ergebnisse der Assoziation in Raum und Zeit ein, um somit die Fragen nach dem wo, wie und wann zu beantworten.[30] Die gesammelten Erkenntnisse werden dann in Phase drei des SAP-Modells präsentiert. Wichtig ist hierbei die nachvollziehbare Dokumentation der gezogenen Schlüsse.[21]

### 2.1.2 Forensic Technology

Grundlegend gibt es zwei Ansätze, wie die Forensische Analyse durchgeführt werden kann. Eine Möglichkeit ist es, den Rechner herunterzufahren und die Festplatte in aller Ruhe zu untersuchen, diese Methode wird *Post Mortem Analyse* genannt. Die andere Option ist die sogenannte *Live Response*, bei der der laufende Rechner untersucht wird.[21] Im folgenden werden beide Methoden genauer erläutert, wobei die *Post Mortem Analyse* eher im Vordergrund steht, da die im Rahmen der Masterarbeit entwickelte Software eben diese unterstützt. Weiterhin wird versucht, die Frage zu beantworten, ob Open-Source-Software ausreichend ist, um einen Analyseprozess vollständig durchzuführen.

### 2.1.3 Live Response

Bei dieser Analysemethode ist es von besonderer Bedeutung, schnell zu handeln, da der große Vorteil der Live Response darin liegt, auch flüchtige Beweise, wie z.B. Daten im RAM zu sichern. Sollte der Angreifer noch aktiv sein, ist es auch möglich, Daten aus dem Netzwerkverkehr sicherzustellen. Der zweite wichtige Aspekt der Live Response ist, das System bei der laufenden Untersuchung möglichst wenig zu verändern. Der Forensiker sollte aus diesem Grund jede ausgeführte Aktion auf dem Rechner nachvollziehbar dokumentieren.[20] Um die Integrität der Daten sicherzustellen, sollten die Tools von einer vertrauenswürdigen Quelle gestartet werden, wie z.B. einer CD oder einem USB-Stick, da Malware auch die Befehle, die zur Ausgabe genutzt werden, beeinflussen kann. Das Ausführen der Befehle sollte automatisiert stattfinden. Zum Sammeln der Beweise sind eine Vielzahl an Befehlen notwendig. Das manuelle Aufrufen der Befehle kostet eine Menge Zeit, die bei dieser Analysemethode oft nicht vorhanden ist, ebenso können schnell Fehler auftreten. Die benötigten Werkzeuge

sollten somit automatisiert und immer mit den gleichen Parametern aufgerufen werden. Um diesen Prozess zu erleichtern, gibt es für die verschiedenen Betriebssysteme verschiedene Toolkits. Eines der bekanntesten für Windows ist das *Windows Forensik Toolchest*, welches verschiedene Programme mit vorkonfigurierten Optionen aufruft. Nach dem Durchlauf erzeugt es eine Ausgabe im HTML-Format. Für Linux kann beispielsweise das Tool *Linux-Live-Response* verwendet werden, welches flüchtige Daten auf einem externen Speichermedium sichert. Ist der Bericht fertig generiert, kann offline und ohne Zeitdruck nach verdächtigen Spuren gesucht werden. [21]

#### 2.1.4 Post Mortem Analyse

Um die Post Mortem Analyse durchzuführen muss zunächst eine bitgenaue 1:1-Kopie des Datenträgers erstellt werden. Eine 1:1-Kopie unterscheidet sich erheblich von einem einfachen Backup, da alle Zeitstempel erhalten bleiben und auch nicht belegter Speicherplatz ebenfalls gesichert wird. Um zu verhindern, dass beim Mounten des Backup-Datenträgers die Quelle verändert wird, können Hardware Write-Blocker eingesetzt werden. Um sicher zu gehen, dass das Erzeugen der 1:1-Kopie erfolgreich war, kann vor und nach dem Sichern eine kryptografische Prüfsumme erzeugt werden. Stimmt diese auf dem Live-System und auf der Sicherung überein, kann der Forensiker sicher sein, dass das Sichern erfolgreich war. Wurde die Kopie des Datenträgers erzeugt, kann dieser analysiert werden, ohne dass der Ermittler ständig Gefahr läuft, Daten zu verlieren.[21] Einer der größten Nachteile dieser Analysemethode ist jedoch, dass wichtige Informationen, wie z.B. offene Verbindungen, laufende Prozesse oder eingeloggte User verloren gehen, wenn der Computer ausgeschaltet wird. [20]

Um die Analyse durchzuführen gibt es eine Vielzahl an Tools, die verschiedene Bereiche eines Systems analysieren. Meist werden diese Tools über die Kommandozeile gesteuert, es gibt jedoch auch Tools mit einer grafischen Oberfläche wie z.B. Autopsy - Digital Forensic, auf die später noch genauer eingegangen wird.

#### 2.1.5 Sleuth Kit

Das Sleuth kit ist eine Sammlung verschiedener Tools, die auf der Kommandozeile ausgeführt werden. Das Hauptaugenmerk dieser Tools liegt auf der Analyse von Datenträgern, wobei die gängigen Dateisysteme wie z.B. NTFS, FAT, ExFAT und viele weitere unterstützt werden. Weiterhin können die Tools sowohl auf Windows als auch auf Unix Systemen verwendet werden.[8] Nach [21] sind typische Werkzeuge von Forensikern beispielsweise **find** oder **cat**. Das Tool **find** gibt dem Anwender die Möglichkeit, ein Dateisystem nach unterschiedlichen Kriterien zu durchsuchen. Der Forensiker kann unter anderem Hashes der Dateien im System erstellen, diese Hashes können dann mit Datenbanken wie der NIST National Software Reference Library verglichen werden.[21] Ein typi-

scher Computer enthält zwischen 10.000 und 100.000 Dateien, es ist also sehr aufwändig, sich diese Dateien alle im einzelnen anzusehen.[23] Um die Untersuchung zu beschleunigen, kann der Ermittler durch Datenbanken wie die der NIST viele schon bekannte Dateien identifizieren und muss diese nicht mehr einzeln betrachten. Bereits im September 2004 hat die Datenbank der NIST 28 Millionen Dateisignaturen enthalten und wurde seitdem jedes Quartal aktualisiert. Auf diese Weise wird ein Großteil des Aufwands bei der Feststellung, welche Dateien auf Computern oder Dateisystemen, die im Rahmen von strafrechtlichen Ermittlungen beschlagnahmt wurden und als Beweismittel wichtig sind, verringert.[3] Mit dem Tool **cat** können Schlüsselwörter in einem Image gefunden werden.[21] Diese Tools und noch viele weitere sind ein Bestandteil des Sleuth Kit, wobei hier zu erwähnen ist, dass es nicht das eine richtige Tool oder eine Reihe von Tools gibt. Der Schlüssel liegt darin, das richtige Tool für die jeweilige Aufgabe parat zu haben.[23]

### 2.1.6 Autopsy - Digital Forensic

Autopsy baut auf dem eben beschriebenen Sleuth Kit auf und bietet eine grafische Oberfläche. Der Forensiker wird mit Hilfe der grafischen Oberfläche durch den Analyseprozess geführt. Zunächst können verschiedene Analyse-Modi gewählt werden, diese werden dann auf ein zuvor gewähltes Festplatten-Image angewendet. Zu den Analyse-Modi gehören die **File-Analyse**, bei der Dateien und Verzeichnisse eines Dateisystems angezeigt werden, **Keyword-Search**, bei der nach verschiedenen Stichwörtern gesucht werden kann, **File Type**, wobei Dateien anhand ihres Dateityps sortiert werden können. Spannend ist hierbei besonders, dass Dateien angezeigt werden können, bei denen der MIME-Type nicht mit der Dateierweiterung übereinstimmt. Das kann bei der Erkennung einer Ransomware von großer Hilfe sein, da erkennbar ist, ob eine Datei mit der Endung **.jpg** eigentlich eine **.exe** Datei ist und somit dazu verwendet wurde, um die Ransomware auszuführen. Weiterhin kann Autopsy mit dem Modul Image-Details detaillierte Informationen zu Bildern oder Details zu Verzeichniseinträgen mit dem Modul **Meta Data** geben. Autopsy generiert während der Analyse Prüfsummen zu allen erzeugten Dateien, um die Integrität zu wahren. [21] Dolle kommt in [21] zu dem Entschluss, dass Open-Source-Software zwar ausreichend ist um eine Analyse vollständig durchzuführen, jedoch muss auf eine Vielzahl verschiedener Werkzeuge zurückgegriffen werden. Weiterhin ist der Funktionsumfang und Bedienkomfort nicht auf dem Level von kommerziellen Anwendungen wie z.B. Encase.

## 2.2 Auswertung und Aufbereitung

Das Auffinden elektronischer Beweise ist nicht besonders schwer, die Auswertung und Analyse jedoch schon. Es erfordert jahrelange Erfahrung und eine fundierte Ausbildung um eine cyberforensische Untersuchung durchzuführen und die Ergebnisse richtig zu interpretieren. Es kann für einen Fall schlimme Folgen haben, wenn Beweise falsch interpretiert werden oder es versäumt wird, diese überhaupt

als Beweis wahrzunehmen. Die Folgen reichen von finanziellen Schäden bis hin zur rechtlichen Haftung für das Unternehmen und die berufliche Haftung für den Ermittler.[23] Um die Beweiskette nachzuweisen muss dem Ermittler jede Einzelheit über den Umgang mit den Beweismitteln bekannt sein und zwar in jedem Schritt. Die Frage nach den sechs Ws muss angewendet werden. Diese Fragen (aus [18]) lauten wie folgt:

- Um **wen** handelt es sich?
- **Was** ist passiert?
- **Wann** hat es stattgefunden?
- **Wo** hat es stattgefunden?
- **Warum** geschah es?
- **Wie** ist es passiert?

Wie die Ergebnisse der Untersuchung letztendlich präsentiert werden, hängt von den Umständen ab. Hier macht es einen Unterschied, ob eine Untersuchung von einem Unternehmen in Auftrag gegeben worden, oder ob sie Teil eines Strafverfolgungsverfahrens ist. Es kann also erforderlich sein, dass der cyberforensische Ermittler die Ergebnisse vor Gericht oder dem Vorstand eines Unternehmens präsentiert. Unabhängig vom Rahmen der Präsentation sollte der Abschlussbericht des Ermittlers als urheberrechtlich geschützt und vertraulich betrachtet werden. Nur Personen, die die entsprechenden Berechtigungen haben, sollten darauf Zugriff haben. Die Inhalte und Details des Berichts können dabei je nach Organisation und Abteilung unterschiedlich sein. Jedoch sollte der Bericht in jedem Fall einen klaren Zeitplan der Abläufe aufzeigen. Ebenfalls enthalten sollte er eine fundierte Dokumentation der Schritte, Maßnahmen und Erkenntnisse und vor allem der Schlussfolgerungen, die der cyberforensische Ermittler getroffen hat, um auf das Ergebnis zu kommen.[23] Abschließend sollte erwähnt werden, ob die Hypothese, die der Ermittler zu Beginn aufgestellt hat, angenommen oder abgelehnt werden muss.[27] Letztendlich sollte der Ermittler eine angemessene Anzahl an Kopien des Abschlussberichts anfertigen. [23]

### 2.3 Implikationen für das geplante Modell

Hauptergebnis der Literaturanalyse ist, dass durch die Sammlung des Wissens eine Wissensbasis über bereits vorhandene Tools gewonnen werden konnte. Hierbei kann zwischen Open-Source-Software und kommerziellen Programmen unterschieden werden. Open-Source-Software ist ausreichend, um eine Analyse vollständig durchzuführen und bietet den Vorteil, dass der Quellcode einsehbar ist, verfügt jedoch nicht über den Funktionsumfang der kommerziellen Programme. [21] Diese Erkenntnis ist hilfreich, da auch für die Umsetzung der Masterarbeit ein Open-Source-Tool verwendet wird.

Abgesehen davon konnte festgestellt werden, dass es verschiedene Prozessmodelle für den Ablauf einer forensischen Untersuchung gibt. Im Kern ähneln diese sich sehr stark, unterteilen die verschiedenen Prozesse jedoch häufig unterschiedlich oder benennen die einzelnen Schritte anders. Im Vordergrund dieser Masterarbeit steht jedoch die Analyse der gesammelten Beweise. Zur Analyse konnte eine grundlegende Wissensbasis geschaffen werden. Unter anderem wurde festgestellt, worauf bei einer forensischen Analyse zu achten ist. Es sollte z.B. gut nachvollziehbar sein, wer wann Zugriff auf die Beweise hatte und welche Interaktionen durchgeführt worden sind, da jede Interaktion und jeder Kontakt mit den Beweisen diese verändern kann. Weiterhin kann zwischen zwei Analysemethoden gewählt werden, der **Live Response** und der **Post Mortem Analyse**. Beide Methoden haben Vor- und Nachteile. Bei der Live Response ist der Zugriff auf flüchtige Daten wie z.B. den RAM, in dem wichtige Informationen zu den laufenden Prozessen stehen können, möglich. Nachteil ist jedoch, dass die Sicherung der Beweise zeitnah geschehen muss, ohne dass das System in der Zwischenzeit heruntergefahren worden ist wurde, da dabei wichtige Informationen verloren gehen können. Die Post Mortem Analyse hat den Nachteil, dass kein Zugriff auf die flüchtigen Informationen aus dem RAM möglich ist, jedoch kann die Analyse der Beweise in aller Ruhe durchgeführt werden, da hier eine Kopie der Festplatte analysiert wird und das System bei der Analyse nicht hochgefahren ist. Im Rahmen dieser Masterarbeit wird die Post Mortem Analyse im Vordergrund stehen.[21] Weiterhin konnten aus [23] einige Richtlinien und Hinweise aus der Literatur abgeleitet werden:

- Analysten sollten Kopien von Dateien untersuchen, nicht die Originaldateien. Der Analytiker sollte mehrere Kopien der gewünschten Dateien oder Dateisysteme erstellen, in der Regel eine Master- und eine Arbeitskopie
- Eine Bit-genaue Kopie sollte durchgeführt werden, wenn Beweise für die Strafverfolgung oder für Disziplinarmaßnahmen benötigt werden oder wenn es wichtig ist, die Dateizeiten zu erhalten
- Analysten sollten sich bei der Identifizierung von Dateitypen auf den Dateihheader und nicht auf Dateiendungen verlassen. Da Benutzer einer Datei jede beliebige Dateiendung zuweisen können, sollten Analysten nicht davon ausgehen, dass diese korrekt sind.
- Unternehmen sollten sich der technischen und logistischen Komplexität der Analyse bewusst sein

Diese Erkenntnisse werden bei der Ausarbeitung und Entwicklung des Projekts im Rahmen der Masterarbeit von großer Hilfe sein.

Für das in der Masterarbeit zu entwickelnde Modell ergeben sich daraus folgende Anforderungen und Gestaltungsmöglichkeiten:

- Da Open-Source-Software ausreichend ist, um eine vollständige Analyse durchzuführen, kann diese für die Masterarbeit verwendet werden.

- Der Anwender hat wenig Erfahrung und benötigt viel Zeit für die Analyse, weshalb eine Live Response Analyse ungeeignet ist. Eine Post Mortem Analyse sollte zum Einsatz kommen.
- Zu Beginn des Prozesses muss eine Bit-genau Kopie erstellt werden. Dem Nutzer muss erklärt werden, wie er diese Kopie erstellt.
- Es wird eine Gruppe von Aufgaben geben, die der Nutzer durchlaufen muss, um eine Analyse durchzuführen.
- Dem Nutzer muss klar sein, dass die Beweise nicht verändert werden sollten, um ein möglichst genaues Ergebnis zu erhalten.
- Es handelt sich um einen komplexen Prozess, der eine fundierte Ausbildung benötigt, um diesen professionell durchführen zu können. Das Vorgehen muss daher sehr detailliert und für Laien verständlich beschrieben werden.

### 3 Methodischer Ansatz

Informationssysteme werden genutzt um die Effektivität und die Effizienz von Unternehmen zu verbessern. Welche Fähigkeiten das Informationssystem hat, hängt dabei von den Merkmalen des Unternehmens und den Entwicklungs- und Implementierungsmethoden ab. Der Großteil der Arbeit von IT Managern befasst sich mit dem Design der zielgerichteten Organisation von Ressourcen zur Erreichung eines bestimmten Ziels. Design ist dabei ein Prozess, eine Abfolge von Expertentätigkeiten, die ein innovatives Produkt hervorbringt. Dieses Produkt wird dann bewertet, um die Qualität als auch den Designprozess zu verbessern. Dieser Prozess wird mehrfach wiederholt. Die Ausrichtung der Forschungsaktivitäten auf die Bedürfnisse des Unternehmens gewährleistet die Relevanz der Forschung, wobei das Ziel der Design Science der Nutzer ist. Das Grundprinzip ist dabei, Wissen und Verständnis eines Designproblems durch den Bau eines Produkts oder Prototypen zu erlangen. Dabei sollen die sieben Leitlinien helfen, auf die im Folgenden eingegangen wird. Eine Übersicht über die Leitlinien findet sich zudem in Tabelle 1. [24]

<b>Leitlinie</b>	<b>Beschreibung</b>
Leitlinie 1: Design als Produkt	Designwissenschaftliche Forschung muss ein praktikables Produkt in Form eines Konstrukts, eines Modells, einer Methode oder einer Instanzierung hervorbringen.
Leitlinie 2: Problemrelevanz	Ziel der designwissenschaftlichen Forschung ist es, technologiebasierte Lösungen für wichtige und relevante Unternehmensprobleme zu entwickeln.
Leitlinie 3: Bewertung des Entwurfs	Der Nutzen, die Qualität und die Wirksamkeit eines Design-Artefakts müssen durch gut durchgeführte Evaluierungsmethoden rigoros nachgewiesen werden.
Leitlinie 4: Beitrag der Forschung	Wirksame designwissenschaftliche Forschung muss klare und nachprüfbare Beiträge in den Bereichen Design-Artefakt, Design-Grundlagen und/oder Design-Methoden liefern.
Leitlinie 5: Strenge der Forschung	Design-Science stützt sich auf die Anwendung strenger Methoden sowohl bei der Konstruktion als auch bei der Bewertung des Design-Artefakts.
Leitlinie 6: Design als Suche	Die Suche nach einem wirksamen Artefakt erfordert den Einsatz der verfügbaren Mittel, um die gewünschten Ziele zu erreichen und gleichzeitig den Gesetzen der Problemumgebung gerecht zu werden.
Leitlinie 7: Kommunikation der Forschung	Designwissenschaftliche Forschung muss sowohl für technologieorientierte als auch für managementorientierte Zielgruppen präsentiert werden.

Tabelle 1: Design-Science Leitlinien [24]

### 3.1 Leitlinie 1: Design als Artefakt

Das Ergebnis der Design-Science im Bereich der Informationstechnologien ist per Definition ein zielgerichtetes IT-Artefakt, das zur Lösung eines wichtigen organisatorischen Problems geschaffen wurde. Es muss effektiv beschrieben werden, um seine Implementierung und Anwendung in einem geeigneten Bereich zu

ermöglichen. Viel mehr noch ist die Anpassung an eine Organisation für die erfolgreiche Entwicklung und Implementierung eines Informationssystems entscheidend. Die Instantiierung eines Artefakts zeigt die Machbarkeit sowohl des Entwurfsprozesses als auch des entworfenen Produkts. [24]

In diesem Fall ist das Problem, dass KMUs oft nicht die notwendigen Ressourcen haben um einen IT-Forensiker zu bezahlen, der nach einem Cyber-Angriff eine professionelle Analyse durchzuführen kann. Das zu entwickelnde Artefakt soll daher einen Menschen mit guten IT-Kenntnissen dabei unterstützen, eine Analyse einer Ransomware-Attacke selbstständig durchzuführen. Um dieses Problem zu lösen, soll ein Leitfaden entwickelt werden, der in Form einer Website aufgerufen werden kann und die einzelnen Schritte, die zur Analyse durchgeführt werden müssen, beschreibt. Womöglich sollen Prozesse automatisiert werden um die Analyse einfach zu gestalten. Außerdem soll diese allgemein gehalten werden, sodass sie sich in möglichst vielen Unternehmen und auf den gängigen Betriebssystemen anwenden lässt. Bei diesem Projekt ist es daher nicht möglich, die Entwicklung an eine einzelne Organisation oder ein Unternehmen anzupassen

### 3.2 Leitlinie 2: Problemrelevanz

Ziel der Forschung im Bereich der Informationssysteme ist es, Wissen und Verständnis zu erlangen, das die Entwicklung und Umsetzung technologiegestützter Lösungen für bisher ungelöste und wichtige Geschäftsprobleme ermöglicht. Formal kann ein Problem definiert werden, als die Unterschiede zwischen einem Zielzustand und dem aktuellen Zustand eines Systems. Die Problemlösung kann als ein Suchprozess (siehe Leitlinie 6) definiert werden, bei dem Maßnahmen zur Verringerung oder Beseitigung der Unterschiede eingesetzt werden. Diese Definitionen setzen eine Umgebung voraus, die einem System sowohl Zielkriterien als auch Beschränkungen auferlegt. Die Relevanz der Design Science liegt in der Beziehung zu einer bestimmten Gruppe von Menschen. [24]

Durch die zunehmenden Cyberangriffe gewinnt auch die IT-Forensik immer mehr an Bedeutung. [30] Ziel dieser Arbeit ist es, Menschen mit guten IT-Kenntnissen eine selbstständige Analyse eines von Ransomware befallenen Systems zu ermöglichen. Die Art der Analyse ist dabei keine Live Response sondern eine Post Mortem Analyse, sie wird also an einem heruntergefahrenen System durchgeführt. Der durchführenden Person soll geholfen werden zu ermitteln, wann der Angriff stattgefunden hat, welche Ransomware genau eingesetzt worden ist und wie diese in das System gelangt ist. Eine Beschränkung des Systems ist, dass die Analyse nicht voll automatisch ablaufen kann. Weiterhin ist die Analyse nicht durchführbar für Menschen mit schlechten IT-Kenntnissen und ausschließlich für Ransomware. Die Analyse-Software muss in der Lage sein, auf möglichst vielen Betriebssystemen zu laufen. Da die Analyse auch für finanziell schlecht aufgestellte Unternehmen durchführbar sein soll, darf nur kostenlos zur Verfügung stehende Software eingesetzt werden. Es ergeben sich hiermit folgende **Zielkriterien**:

- Die Analyse ist eine Post Mortem Analyse
- Sie muss für Menschen mit guten IT-Kenntnissen durchführbar sein
- Die Analyse-Software soll auf möglichst vielen Betriebssystemen laufen
- Es darf keine gebührenpflichtige Software verwendet werden
- Die Datenbank der Ransomware-Typen soll erweiterbar sein, um neue Ransomware zu erkennen

Weiterhin konnten folgende **Beschränkungen** ermittelt werden:

- Es kann nicht an einem laufenden System gearbeitet werden
- Die Analyse ist für Menschen mit schlechten IT-Kenntnissen nur sehr schwer durchführbar
- Die Analyse muss mit kostenloser Software durchführbar sein
- Die Software kann nicht bei der Analyse anderer Cyberangriffe eingesetzt werden und beschränkt sich somit ausschließlich auf Ransomware

### 3.3 Leitlinie 3: Bewertung des Designs

Evaluierungsmethoden sind entscheidend, um den Nutzen, die Qualität und die Wirksamkeit eines Design-Artefakts oder eines Prototypen nachzuweisen. Das Unternehmensumfeld, in dem das Produkt eingesetzt werden soll, legt die Anforderungen fest, auf denen deren Bewertung beruht. Die Evaluierung ist somit ein wesentlicher Bestandteil des Forschungsprozesses. Das IT-Artefakt kann in Hinblick auf verschiedene Kriterien bewertet werden. Darunter fallen beispielsweise Funktionalität, Vollständigkeit, Konsistenz, Genauigkeit, Leistung, Zuverlässigkeit, Benutzerfreundlichkeit, Eignung für die Organisation und andere relevante Qualitätsmerkmale. Da es sich beim Entwurf um eine iterative und inkrementelle Tätigkeit handelt, liefert die Evaluierungsphase der Konstruktionsphase wichtige Rückmeldungen über die Qualität des Entwurfsprozesses und des zu entwickelnden Produkts. Das Design-Artefakt kann als vollständig und effektiv bezeichnet werden, wenn es die Anforderungen des Problems erfüllt und Einschränkungen berücksichtigt werden. Weiterhin zu berücksichtigen ist, wenn sich die verfügbare Technologie oder das organisatorische Umfeld ändern, dass die Annahmen früherer Forschungsarbeiten ungültig werden. Nach [24] gibt es verschiedene Methoden ein Design zu evaluieren, diese sind in Tabelle 2 zusammengefasst. Die Auswahl der Evaluationsmethoden muss auf das entworfene Artefakt und die gewählten Evaluationsmetriken abgestimmt sein. Die Güte und Wirksamkeit eines Artefakts kann durch gut gewählte Evaluationsmethoden nachgewiesen werden.[24]

Das Artefakt wird für kein spezielles Unternehmensumfeld entwickelt, sondern soll in möglichst vielen kleinen und mittelständischen Unternehmen zum Einsatz

kommen können. Dabei soll es für möglichst viele verschiedene Ransomware-Typen funktionieren, die in der Datenbank hinterlegt sind. Diese Datenbank sollte erweiterbar sein, um zukünftig noch weitere Arten von Ransomware erkennen zu können. Der Analyseprozess ist nicht vollständig automatisierbar, da Autopsy nicht die notwendigen Exporte anbietet und so Daten für ein eigens entwickeltes Tool fehlen. Der Prozess soll soweit wie möglich automatisiert werden, benötigt jedoch aufgrund der Komplexität des Themas weiterhin einen Menschen mit guten IT-Kenntnissen, der diesen durchführt. Zur Evaluierung der Funktion, Benutzerfreundlichkeit, Vollständigkeit, Konsistenz, Genauigkeit, Leistung und Zuverlässigkeit soll die Testing-Evaluierungsmethode wie in Tabelle 2 unter Punkt 4 verwendet werden. White Box-Tests sollen auf verschiedenen Systemen mit verschiedenen Ransomware-Typen ausgeführt werden, anschließend muss das Tool auf die infizierten Systeme angewendet und eine Evaluierung vorgenommen werden. Die Evaluierung erfolgt nach folgenden Kriterien:

### **3.3.1 Usability**

- Kann die Analyse von einem Menschen mit guten IT-Kenntnissen durchgeführt werden?
- Wie wird die Benutzerfreundlichkeit eingeschätzt?
- Eignet sich der Prototyp für kleine und mittelständische Unternehmen?
- Ist der Prototyp auf unterschiedlichen Betriebssystemen lauffähig?
- Welche Bereiche kann das System?

### **3.3.2 Funktional**

- Welcher Typ von Ransomware wird bei der automatischen Analyse erkannt?
- Wie hoch ist die Trefferquote bei der automatischen Analyse?
- Ist der Prototyp für die Zukunft erweiterbar?
- Wie hoch ist die algorithmische Komplexität?
- Wie hoch ist die Storage Complexity?
- Gibt es Ransomware die das System nicht erkennt?

<b>Methode</b>	<b>Beschreibung</b>
1. Beobachtung	Fallstudie: Vertiefung des Artefakts im Unternehmensumfeld
	Feldstudie: Überwachung der Verwendung von Artefakten in mehreren Projekten
2. Analytisch	Statische Analyse: Untersuchung der Struktur eines Artefakts auf statische Eigenschaften (z. B. Komplexität)
	Architektur-Analyse: Untersuchung der Einpassung von Artefakten in die technische IS-Architektur
	Optimierung: Demonstration inhärenter optimaler Eigenschaften des Artefakts oder Bereitstellung von Optimalitätsgrenzen für das Verhalten des Artefakts
	Dynamische Analyse: Untersuchung von Artefakten im Gebrauch auf dynamische Eigenschaften (z. B. Leistung)
3. Experimentel	Kontrolliertes Experiment: Untersuchung eines Artefakts in einer kontrollierten Umgebung auf seine Eigenschaften (z. B. Benutzerfreundlichkeit)
	Simulation - Ausführen eines Artefakts mit künstlichen Daten
4. Testing	Funktionale (Black Box) Tests: Ausführen von Artefaktschnittstellen zur Entdeckung von Fehlern und zur Identifizierung von Mängeln
	Strukturelle (White Box) Tests: Durchführung von Abdeckungsprüfungen einiger Metriken (z. B. Ausführungspfade) in der Artefaktimplementierung
5. Beschreibend	Fundierte Argumentation: Verwendung von Informationen aus der Wissensbasis, um ein überzeugendes Argument für den Nutzen des Artefakts zu entwickeln
	Szenarien: Entwickeln Sie detaillierte Szenarien rund um das Artefakt, um seinen Nutzen zu demonstrieren.

Tabelle 2: Design Evaluierungsmethoden [24]

### **3.4 Leitlinie 4: Beiträge zur Forschung**

Die Frage, die bei der Bewertung jeder Art von Forschung am wichtigsten ist, lautet: Was sind die neuen und interessanten Beiträge? Design-Science birgt das Potenzial für drei Arten von Forschungsbeiträgen, die auf der Neuartigkeit, der Allgemeinheit und der Bedeutung des entworfenen Artefakts basieren. Eine oder mehrere dieser Eigenschaften müssen in einem Forschungsprojekt zu finden sein. [24]

#### **3.4.1 Das Design-Artefakt**

Meist ist der Beitrag der Design-Science das Artefakt oder der Prototyp selbst. Dieser muss die Lösung eines bisher ungelösten Problems ermöglichen. [24]

#### **3.4.2 Grundlagen**

Die kreative Entwicklung von neuartigen, angemessen bewerteten Konstrukten, Modellen, Methoden oder Instanzen, sowie die Erweiterung und Verbesserung der bestehenden Grundlagen auf Wissensbasis der Design-Science sind ebenfalls wichtige Beiträge zur Forschung. [24]

#### **3.4.3 Bewertung des Designs**

Schließlich ist die kreative Entwicklung und Anwendung von Bewertungsmethoden (z.B., experimentell, analytisch, beobachtend, testen und beschreiben) und neue Bewertungs-Metriken ein entscheidender Bestandteil der Design-Science. [24]

In diesem Fall soll ein Prototyp entwickelt werden, welcher unterstützend zu der Analyse einer Ransomware-Attacke beiträgt. Es handelt sich dabei nicht um ein bisher ungelöstes Problem, jedoch konnte ein ähnlicher Leitfaden während der Recherche nicht gefunden werden. Die spannende Neuheit an diesem Prototyp soll somit die Vereinfachung der Analyse sein. Erreicht werden soll das, indem einem Menschen mit guten IT-Kenntnissen, der aber kein ausgebildeter IT-Forensiker ist, ein Leitfaden an die Hand gegeben wird, der ihm/ihr die Analyse ermöglicht, ohne dafür einen IT-Forensiker zu benötigen.

### **3.5 Leitlinie 5: Strenge der Forschung**

Die Anwendung strenger Methoden ist bei Design-Science sowohl bei der Konstruktion, als auch bei der Bewertung des Artefakts oder Prototypen erforderlich. Dabei bezieht sich Strenge auf die Art und Weise wie die Forschung durchgeführt wird. Besonders in Hinblick auf die Anwendbarkeit und die Verallgemeinerbarkeit des Artefakts muss die Strenge bewertet werden. Diese ergibt sich dabei aus der effektiven Nutzung der Wissensbasis, d. h. der theoretischen Grundlagen und Forschungsmethoden. Der Erfolg der Forschung beruht dabei maßgeblich auf der Auswahl geeigneter Techniken und der Auswahl der Mittel,

zur Bewertung dieser durch den Forscher. Die Anwendung dieser Techniken hilft eine Theorie oder ein Artefakt zu entwickeln. Diese Artefakte sind oft Teil eines Mensch-Maschine-Problemlösungssystems und müssen in einer geeigneten Umgebung von geeigneten Probanden erprobt werden. Das Hauptziel der Forschung ist jedoch nicht zu theorisieren oder zu beweisen wieso ein Artefakt funktioniert, sondern darin, zu ermitteln, wie gut ein Artefakt funktioniert.[24]

Um zu bewerten, wie gut das entwickelte Artefakt funktioniert werden im Vorfeld verschiedene Bewertungskriterien festgelegt, die auch unter Kapitel 3.3 zu finden sind. Dabei wird auf verschiedene Aspekte eingegangen. Um diese Kriterien in einer geeigneten Umgebung zu testen, bietet sich ein White Box-Test wie in Tabelle 2 unter Punkt 4 beschrieben, an. Durch den Test können die verschiedenen Kriterien bewertet und anschließend in einem iterativen Prozess angepasst und optimiert werden, bis das Artefakt möglichst gut funktioniert.

### **3.6 Leitlinie 6: Design als Suchprozess**

Bei einem realistischen Problem mit einem Informationssystem ist die Suche nach einem optimalen Ergebnis oft nicht zu bewältigen. Der Designprozess ist daher nach [24] ein Kreislauf, in dem immer wieder eine neue Lösung generiert und dann wieder getestet wird. Dieser Kreislauf wird in Grafik 3 gezeigt. Design-Science ist ein iterativer Prozess, bei dem eine effektive Lösung gesucht wird unter Einhaltung der in der Umwelt bestehenden Gesetze. Zunächst werden die Mittel, Ziele und Gesetze abstrahiert und geeignet dargestellt, um dann in weiteren Iterationen den Umfang des Entwurfsproblems zu erweitern und so einen Fortschritt zu erzielen. Je mehr die Maßnahmen, Mittel, Ziele und Gesetzmäßigkeiten verfeinert werden, desto mehr steigt auch der Wert und die Relevanz des Design-Artefakts. Letztendlich sollte das Artefakt für eine bestimmte Klasse von Problemen gut funktionieren. Wie gut ein Artefakt funktioniert, wird durch den Vergleich zu anderen Designern deutlich, die eine Lösung für dieselbe Problemsituation erstellt haben. Es ist wichtig zu verstehen, warum ein Artefakt funktioniert, jedoch steht zunächst im Vordergrund festzustellen, dass es in einer bestimmten Umgebung funktioniert, auch wenn nicht vollständig erklärt werden kann warum.[24]

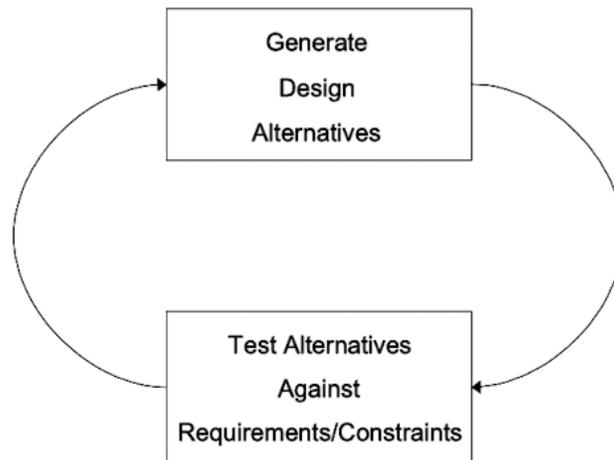


Abbildung 3: Der Generate und Test Zyklus nach [24]

Um zu zeigen, dass das Artefakt funktioniert, muss es immer wieder in einer Testumgebung eingesetzt werden. Dabei müssen auftretende Probleme in einem iterativen Prozess behoben werden, damit sich der Prototyp einem optimalen Ergebnis annähert. Der Prototyp muss am Ende der Entwicklungsphase gut für die Bewertungskriterien aus Kapitel 3.3 funktionieren.

### 3.7 Leitlinie 7: Kommunikation der Forschung

Die aus der Forschung gewonnenen Ergebnisse müssen sowohl für technologieorientierte als auch für managementorientierte Zielgruppen präsentiert werden. Dabei ist das technologieorientierte Publikum an Details interessiert, die benötigt werden, um das Artefakt zu konstruieren, da es den Praktikern nur so möglich ist, die Vorteile des Artefakts zu nutzen. Forscher wiederum benötigen eine kumulative Wissensbasis, um auf dem vorhandenen Wissen Erweiterungen oder Bewertungen aufzubauen. Das Management orientierte Publikum hingegen benötigt genügend Details zur Entscheidung, ob Ressourcen zur Konstruktion oder zum Erwerb des Artefakts im spezifischen Kontext eingesetzt werden sollen. Der Fokus der gesamten Präsentation sollte auf der Neuartigkeit und Wirksamkeit des im Artefakt realisierten Lösungsansatzes liegen. [24]

## 4 Vorgehensmodell zur Ransomware-Analyse

Bei der Analyse mit dem in dieser Masterarbeit entwickelten Tool muss sich an einen vorgegebenen Ablauf gehalten werden. Dieser muss zunächst modelliert werden. Das Modell sollte eine effiziente Darstellung des Ablaufs ermöglichen und gleichzeitig übersichtlich und für den Leser leicht verständlich und nachvollziehbar sein, die Semantik trotzdem stark genug, um die Bedeutung und Eigenschaften des Prozesses grafisch darzustellen. Diese Eigenschaften bringt ein BPMN-Diagramm mit sich und ist daher optimal für die Darstellung des Ablaufs geeignet.

### 4.1 BPMN - Business Process Modeling Notation

Ein Geschäftsprozess besteht aus einer Reihe von Aktivitäten, die in einer bestimmten Reihenfolge abgearbeitet werden müssen, um ein bestimmtes Ziel zu erreichen. Die BPMN wurde im Jahr 2004 entwickelt und ist heute der Standard für die grafische Darstellung von Geschäftsprozessen jeder Art. Sie wird verwendet um Geschäftsprozesse zu visualisieren und anschließend zu optimieren. Ziel bei der Entwicklung von BPMN war es, eine Notation zu schaffen, die für die verschiedenen Geschäftsanwender leicht verständlich ist. Angefangen bei den Entwicklern der ersten Entwürfe der Prozesse, über die technischen Entwickler, welche den Prozess implementieren müssen, bis hin zu den Mitarbeitern, die die modellierten und implementierten Prozesse verstehen und durchführen müssen. Ein BPMN-Diagramm besteht aus verschiedenen Elementen, die in Grafik 4 zu sehen sind. Angefangen bei Events, welche den Start des Events oder das Ende beschreiben oder z.B. was bei einem Fehler in einem Prozess passiert. Gateways geben beispielsweise die Möglichkeit Aufgaben parallel ablaufen zu lassen. Activities stellen die verschiedenen Aufgaben für den Anwender dar und werden mit den Connection Objects miteinander verbunden. Swimlanes werden verwendet um die Modellierungselemente zu gruppieren und Artefakte um zusätzliche Informationen in dem Prozess bereitzustellen, welche den Fluss aber nicht beeinflussen.[17]

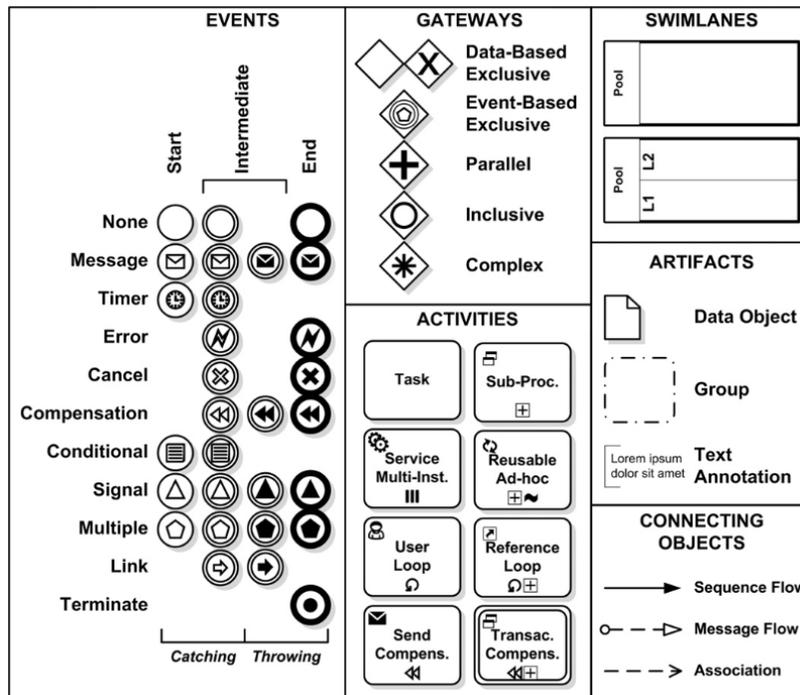


Abbildung 4: BPMN 1.2 Elemente [17]

Durch die hier beschriebenen Eigenschaften und Komponenten ist BPMN geeignet, den Ablauf der Ransomware-Analyse übersichtlich und mit ausreichend Details darzustellen.

## 4.2 Das Vorgehensmodell im Detail

Zu Beginn des Prozesses muss zunächst erkannt werden, dass ein Angriff stattfindet oder stattgefunden hat. Dies geschieht meist dadurch, dass Dateien nicht mehr geöffnet werden können oder es erscheint auf dem Bildschirm eine Lösegeldforderung. Wurde der Angriff erkannt, muss das System heruntergefahren werden. Dies sollte so schnell wie möglich passieren, da manche Ransomware sich über das Netzwerk auf andere Computer oder Server verteilen kann und diese auch verschlüsselt. Sollte das Herunterfahren aus irgend einem Grund nicht möglich sein, so muss versucht werden, den Angriff zu isolieren. Ist das System heruntergefahren oder der Angriff isoliert, muss eine 1:1-Bit-genaue Kopie des Systems erstellt werden. Diese Kopie wird benötigt um die Integrität der Daten zu wahren. Der IT-Forensiker muss dadurch nicht mit der originalen Festplatte arbeiten, sondern kann auf die Kopie zurückgreifen. Das Original kann in einem Tresor verwahrt werden und bleibt unverändert. Zum Durchführen des Backups wird AOMEI Partition Assistant verwendet. Es ist für Privatanwender

kostenlos, muss jedoch für kommerzielle Nutzung lizenziert werden. Ist die Kopie abgeschlossen kann sie mit Autopsy gescannt werden, auch dazu findet sich eine genau Beschreibung im Tool. Autopsy verfügt über eine Vielzahl von Analysemöglichkeiten. Es besteht die Möglichkeit, nach den letzten Aktivitäten auf dem Computer zu suchen. Weiterhin kann die *Extension Mismatcher Detection* eingesetzt werden, welche überprüft ob eine Dateiendung auch zum Dateityp passt, ob also ein .jpg auch wirklich ein Bild ist oder nicht doch ein ausführbares Programm. Außerdem enthalten ist ein Android Analyse Tool, welches nach Daten vom System und Drittanbieter-Apps suchen kann. Nicht alle diese Module sind für einen Ransomware-Angriff relevant, benötigen aber eine Menge Zeit, während andere zwingend durchgeführt werden müssen, um den Tathergang rekonstruieren zu können. Ist der Name der Ransomware schon bekannt durch die Dateiendungen der verschlüsselten Dateien oder die Lösegeldforderung, kann direkt nach Zeitpunkt und Ursache des Angriffs gesucht werden. Ist der Name nicht bekannt, stellt das Tool eine automatische Suche bereit, bei der eine Text-Datei mit allen Dateien auf dem Rechner aus Autopsy exportiert wird. Basierend auf dieser wird dann ein Scann durchgeführt und mit den bekannten Ransomware-Typen aus der Datenbank abgeglichen. Dieser automatische Scann berücksichtigt die Dateiendungen der Dateien auf dem System und sucht nach dem Namen der Lösegeldforderung. Anschließend können verschiedene Filter in Autopsy verwendet werden um herauszufinden, zu welchem Zeitpunkt welche Programme ausgeführt worden sind oder wann Dateien modifiziert wurden. Dies geschieht direkt in Autopsy über die *Timeline*-Funktion, welche optimal geeignet ist, den Ablauf aller Aktionen auf dem Rechner grafisch darzustellen. Diese Timeline lässt sich zudem mit unterschiedlichen Filtern und Suchkriterien auf das Wesentliche reduzieren um dem Nutzer möglichst schnell ersichtlich zu machen, wann der Angriff stattgefunden hat. Ist der Angriffszeitpunkt bekannt, können wieder andere Filter angewendet werden, um einzusehen, wann zum Beispiel eine E-Mail empfangen oder geöffnet wurde, eine bestimmte Website aufgerufen oder wann Programme installiert oder ausgeführt worden sind. Diese Informationen können dem Nutzer Aufschluss darüber geben, wie der Angreifer in das System gelangt ist. Letztendlich werden die Ergebnisse aus der Analyse aufbereitet, dokumentiert und eventuell präsentiert. Die erarbeiteten Informationen können für die Prävention weiterer Angriffe oder die Entschlüsselung der befallenen Daten von Nutzen sein. Alleine wenn der Name der Ransomware bekannt ist, kann nach einem Tool zur Entschlüsselung gesucht werden. In dem Analyse-Tool, das im Rahmen dieser Arbeit entwickelt wird, werden die bekannten Tools zur Entschlüsselung hinterlegt um dem Nutzer einen möglichst schnellen Zugriff darauf gewähren zu können. Kann ein Täter ausfindig gemacht werden, können die Daten auch vor Gericht Verwendung finden. Somit endet der Prozess.

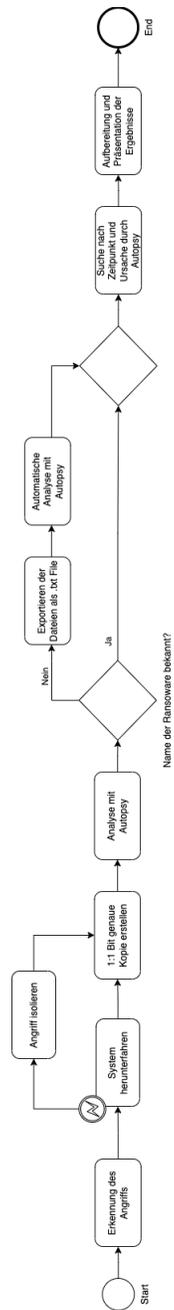


Abbildung 5: Vorgehensmodell

### 4.3 Umsetzbarkeit in Code

Da der Prozess durch das zu entwickelnde Tool nicht vollständig automatisiert, sondern den Anwender bei der Durchführung des Prozesses unterstützen soll, kann der Prozess nicht vollständig in Code umgesetzt werden. Die Erkennung des Angriffs muss unweigerlich durch den Nutzer stattfinden. Der Nutzer muss anschließend individuell entscheiden, ob das System heruntergefahren werden kann oder ob der Angriff isoliert wird. Anschließend kann das Tool zum Einsatz kommen. Eine Erklärung, wie und mit welchem Tool eine Kopie des Datenträgers erstellt werden kann, wird in dem Ransomware-Analyse-Tool bereitgestellt. Ebenso eine genaue Beschreibung, wie die Analyse mit Autopsy durchzuführen ist. Um dem Nutzer einen Teil der Arbeit abzunehmen, kann ein Modul implementiert werden, welches die Dateien auf der Festplatte automatisch analysiert, um so den Typ der Ransomware herauszufinden. Weiterhin sollte das Tool, wenn vorhanden, Informationen darüber bereitstellen, ob es eine Möglichkeit gibt, die Daten mit einem Entschlüsselungs-Tool wiederherzustellen. Die Suche nach dem Zeitpunkt und der Ursache des Angriffs ist nicht automatisierbar, da Autopsy nicht die nötigen Exporte oder Schnittstellen anbietet, um Zugriff auf die benötigten Daten zu geben. An dieser Stelle kann dem Anwender geholfen werden, indem beschrieben wird, wie genau die benötigten Informationen aus Autopsy herausgefiltert werden können. Abschließend muss der Nutzer die Daten selbstständig aufbereiten und gegebenenfalls präsentieren.

## 5 Prototypische Implementierung des Ransomware-Analyse Tools

Da es wichtig war, den Prototypen auf möglichst vielen Plattformen lauffähig zu machen, wurde er in Form einer Website implementiert. Eine aufwändige Installation ist somit nicht notwendig, es wird lediglich ein Browser zum Aufrufen der Website benötigt. Für die Entwicklung wurde kein Content-Management-System wie Wordpress verwendet, sondern auf NextJS zurückgegriffen. Es bietet eine sehr hohe Performance, außerdem gibt es die Möglichkeit, API-Schnittstellen bereitzustellen um die Datenbank durch weitere Ransomware-Typen zu erweitern oder diese zu verändern.

### 5.1 Technology Stack

Zur Umsetzung des Ransomware-Analyse-Tools sind eine Reihe verschiedener Technologien notwendig. Zum einen NextJS, welches ein React-Framework ist und zur Entwicklung der Website verwendet wurde. Es gibt dem Entwickler die Möglichkeit, sowohl Frontend als auch Backend mit einer Programmiersprache zu entwickeln. So werden Dateien, die in dem api-Ordner liegen, automatisch zu einem API-Endpoint. Ein API-Endpoint kann genutzt werden um beispielsweise über eine Schnittstelle noch weitere Ransomware-Typen in die Datenbank einzutragen. Um mit der Datenbank zu kommunizieren wird Prisma genutzt. Prisma wird für das Management der Daten in der Datenbank und die Abfrage dieser Daten verwendet, wodurch die Abfragen *Type-Safe* werden. Der Entwickler kann sich damit mehr auf die Funktionalität als auf komplizierte SQL-Abfragen konzentrieren. Sowohl NextJS, als auch Prisma laufen auf einem Server von Vercel. Prisma unterstützt die Kommunikation von SQL- als auch NoSQL-Datenbanken, daher wurde hier auf die MongoDB Cloud Datenbank gesetzt. Diese besteht nicht aus Tabellen wie eine herkömmliche SQL-Datenbank, sondern aus sogenannten Collections, welche wiederum aus JSON ähnlichen Dokumenten bestehen. Der Browser schickt somit, wie in Grafik 6 gezeigt, eine Anfrage an den Server von Vercel, auf dem das Programm, geschrieben in NextJS, läuft. Bei Bedarf werden durch Prisma weitere Daten von der MongoDB-Datenbank abgefragt, und letztendlich wird im Browser eine Website mit den abgefragten Daten dargestellt.

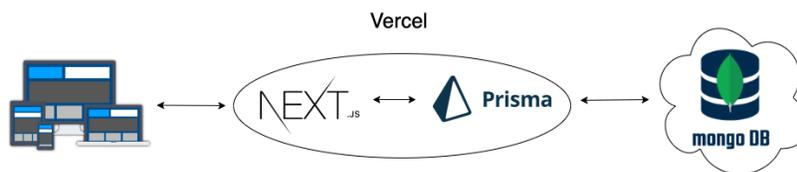


Abbildung 6: Komponenten Diagramm

### 5.1.1 NextJS

NextJS ist ein Framework, welches auf React basiert. Es wird dabei mit einer minimalen Konfiguration ausgeliefert und kann mit verschiedenen Paketen, wie z.B. Prisma, erweitert werden. NextJS unterstützt statisches, clientseitiges und serverseitiges Rendern von Anwendungen. Weiterhin versieht NextJS Seitenverzeichnisse und Seiten, aufgrund ihres Namens, mit einem Routing. Die einzelnen Seiten sind dabei React-Komponenten. Beim Aufrufen einer Seite gibt es die Möglichkeit des statischen Rendering, bei der die Daten zur Erstellungszeit abgefragt werden und des serverseitigen Rendering bei dem die Daten bei jeder Anfrage abgefragt und gerendert werden. [5] Weiterhin kommt NextJS mit einer Vielzahl an Funktionen, die die Performance einer Website steigern und die Entwicklung erleichtern. Angefangen bei der automatischen Bildoptimierung, welche die Bilder immer in der perfekten Größe für das jeweilige Gerät bereitstellt [7] bis hin zum Fast Refresh, was beim Entwickeln der Anwendung die geänderte React-Komponente sofort aktualisiert, damit der Entwickler ein direktes Feedback erhält.[6] Der größte Vorteil von NextJS ist jedoch die Möglichkeit, mit einem Framework sowohl Frontend als auch Backend zu entwickeln. NextJS unterstützt nicht nur das Erstellen von Routen zum Frontend sondern auch zu APIs im Backend. Jede Datei, die im /api Ordner abgelegt wird, wird automatisch ein API-Endpoint.[4] Es ist also nicht notwendig, ein Frontend mit HTML und ein Backend z.B. mit PHP zu bauen, alles funktioniert mit NextJS.

### 5.1.2 Prisma

Prisma ist ein Open-Source ORM welches aus dem Prisma Client, Prisma Migrate und Prisma Studio besteht.[13] Das Schreiben und Debuggen von SQL-Abfragen kann viel Entwicklungszeit in Anspruch nehmen. Prisma soll hier Abhilfe schaffen und es Entwicklern ermöglichen, typischere Abfragen zu schreiben, die JavaScript-Objecte zurück geben. Das Ziel von Prisma ist es, die Arbeit mit Datenbanken zu erleichtern und dadurch zu erreichen, dass Entwickler in Objekten denken können und Abfragen immer Type-safe sind. Weiterhin gibt es eine **Single source of truth** für die Datenbank und die Objekte in der Applikation und automatische Vervollständigung für den Code. Mit Prisma versucht man eine gute Mischung zwischen Kontrolle über den Code und Produktivität zu finden, im Vergleich zu anderen Möglichkeiten der Implementierung. Dies wird in Grafik 7 gezeigt. [14]

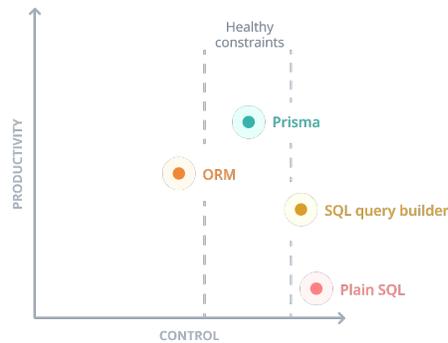


Abbildung 7: Prisma im Vergleich zu Alternativen [14]

Zu Beginn wird die Prisma Schema Datei angelegt. In dieser werden die Datenmodelle, die in der Anwendung benötigt werden, der Connection-String und der Typ der Datenbank definiert. Die in der Schema Datei definierten Datenmodelle ergeben später eine Tabelle in einer relationalen Datenbank oder eine Collection in MongoDB. Weiterhin bilden sie eine Grundlage der Queries, die mittels dem Prisma Client an die Datenbank gesendet werden können. [13] Im folgenden ein Ausschnitt aus dem Prisma Schema des Ransomware-Analyse-Tools:

```

1 datasource db {
2   provider = "mongodb"
3   url      = env("DATABASE_URL")
4 }
5
6 generator client {
7   provider = "prisma-client-js"
8 }
9
10 model Type {
11   id          String @id @default(auto()) @map("_id") @db.ObjectId
12   name        String
13   dateiendung String?
14   loesegeldforderung String?
15   merkmale    String?
16   verschluesselung String?
17   decryptor   String?
18   link        String?
19 }

```

Listing 1: Prisma Schema

Zu Beginn wird der Typ der Datenbank festgelegt, in diesem Fall *MongoDB* und der Connection-String, der in einer Environment-Variable gespeichert ist. Anschließend wird der Generator für den Prisma Client definiert, der später dazu verwendet wird, Anfragen an die Datenbank zu senden. Zuletzt findet sich ein Modell mit dem Namen *Type*, das aufgebaut ist aus einer *id*, die automatisch

generiert wird, einem Namen für die Ransomware, einer Dateiendung, einer Lösegeldforderung, weitere Merkmale, die Art der Verschlüsselung, der Name eines Decryptors und ein Download-Link zum Decryptor. Alle Einträge sind vom Typ String wobei das ? kennzeichnet, dass die Variable optional ist. Nur die ID und der Name sind somit zwingend erforderlich.

Daten aus dem hier definierten Modell *Type* können dann mit dem folgenden Befehl abgerufen werden. Dieser Befehl gibt alle Datensätze aus der Tabelle zurück:

```
1 const rans = await prisma.type.findMany();
```

Um aus dem Datenbankschema die Tabellen zu erzeugen wird Prisma Migrate verwendet. Nachdem Änderungen am Prisma Schema lokal vorgenommen worden sind, wird

```
1 prisma migrate dev
```

im Terminal ausgeführt, welches die Tabellen oder Collections in der Datenbank erzeugt und die Daten im NextJS Code durch den Prisma Client verwendbar macht. Dieser Prozess wird in Grafik 8 visualisiert.

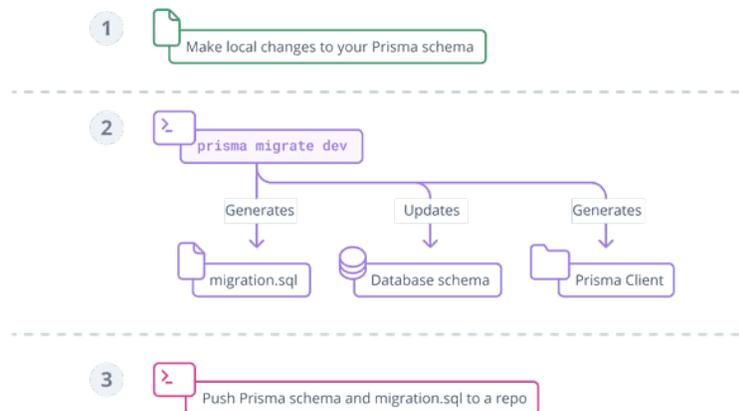


Abbildung 8: Prisma Migrate Workflow [13]

Die letzte Komponente, die Prisma mit sich bringt, ist Prisma Studio. Prisma Studio ist ein Grafisches User Interface welches die Ansicht und Bearbeitung der Daten in der Datenbank ermöglicht. [13]

### 5.1.3 MongoDB

MongoDB ist anders als herkömmliche SQL-Datenbanken nicht aufgebaut aus Tabellen sondern aus Dokumenten und ist damit eine sogenannte NoSQL-Datenbank, aktuell sogar eine der beliebtesten, wenn nicht sogar die beliebteste NoSQL-Datenbank, die in vielen wissenschaftlichen Projekten eingesetzt wird.[26] Daten werden ähnlich wie in einem JSON-Dokument abgespeichert, dadurch kann die Datenstruktur einfach verändert werden und verschiedene Dokumente können sehr unterschiedlich sein. [9] Prisma ist in der Lage, Abfragen auf einer MongoDB Datenbank auszuführen und auch Daten in diese hinein zu schreiben. Ein Beispiel für einen solchen Datenbankeintrag sieht man hier in Grafik 9.

```
_id: ObjectId('62971934faf4b31a4dc855c2')
name: "WannaCry"
dateiendung: ".wcry
              .wncry
              .WNCRY
              .WCRY"
loesegeldforderung: "@Please_Read_Me@.txt"
merkmale: ""
verschlueselung: ""
decryptor: "Trend Micro File Decryptor"
link: "https://success.trendmicro.com/solution/1114221"
```

Abbildung 9: Datenbankeintrag der WannaCry-Ransomware

Dieser Eintrag zeigt das bekannte WannaCry Virus. Jeder Eintrag ist mit einer eindeutigen ID versehen, weiterhin mit Namen und Dateieendungen, die die verschlüsselten Dateien haben können. Der Name der Lösegeldforderung und falls vorhanden, Merkmale zum Virus, der Art der Verschlüsselung und ein Name und Link zum Decryptor. In der Ransomware-Analyse-Software helfen diese Daten beim automatischen Suchen nach dem Namen der Ransomware und helfen dem Nutzer, wenn möglich, ein Tool zur Entschlüsselung zu finden.

## 5.2 Systemarchitektur des Prototyps

In Abbildung 10 wird die Systemarchitektur des Prototypen aus technischer Sicht gezeigt. Hier wird nochmal die Stärke von NextJS deutlich. Statt das Frontend, die externe Schicht mit HTML und CSS zu entwickeln, die Backend-Funktionalität mit PHP und die Abfragen an die Datenbank mit SQL, kann fast alles mit NextJS gelöst werden. Einzig das Design des User Interfaces bildet hier eine Ausnahme, dieses erfordert HTML und CSS, Funktionen im Frontend wiederum können mit NextJS implementiert werden. Soll beispielsweise ein Element auf der Website ein- oder ausgeblendet oder eine Abfrage an die Datenbank gesendet werden, können diese Funktionalitäten mit NextJS implementiert werden. Die logische Schicht und die Datenschicht können vollständig mit NextJS implementiert werden. Funktionen im Backend der Website sowie die API-Schnittstelle werden mit NextJS in Kombination mit Prisma entwi-

ckelt. Sendet das Frontend eine Anfrage an den Server, der bei Vercel gehostet ist, können die Daten von NextJS verarbeitet werden. Müssen Daten an die MongoDB-Datenbank gesendet oder abgefragt werden, so ist dies durch Prisma auch mit NextJS möglich. Es ist somit nicht notwendig, Abfragen in SQL oder der MongoDB eigenen Sprache, MongoDB Query Language (MQL), zu schreiben. Um beispielsweise im Backend alle Daten aus der Tabelle *Type* abzufragen, genügt eine Zeile Code die wie folgt aussieht:

```
1 const rans = await prisma.type.findMany();
```

Listing 2: Abfrage aller Einträge aus der Tabelle Type

Durch diese Systemarchitektur hat der Entwickler den Fokus auf der Programmierung mit einer Programmiersprache und muss sich nicht mit SQL-Abfragen, PHP und JavaScript auseinandersetzen.

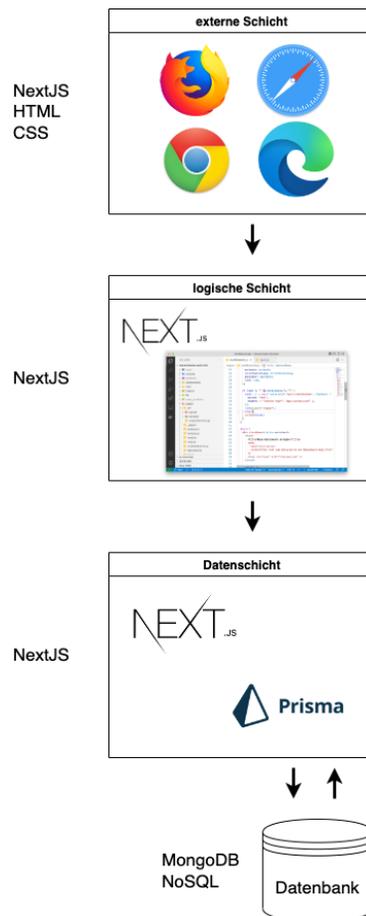


Abbildung 10: Systemarchitektur des Prototyps aus technischer Sicht

Spannend ist außerdem die Systemarchitektur aus funktionaler Sicht, diese wird in Abbildung 11 gezeigt. Der Nutzer hat im Analyse-Tool mehrere Möglichkeiten. Er kann die Automatische Analyse ausführen, diese besteht aus mehreren Funktionen. Zum einen werden die aktuell in der Datenbank hinterlegten Ransomware-Typen abgefragt, eine CSV-Datei kann importiert werden, welche anschließend automatisch untersucht wird. Am Ende der Analyse wird dem Nutzer ein Ergebnis ausgegeben. Der Nutzer kann zudem Ransomware in die Datenbank eintragen, entweder über die bereitgestellte API-Schnittstelle oder einfach über ein Formular auf der Website, oder aber Daten aus der Datenbank abfragen. Es ist möglich, alle aktuellen Datenbankeinträge anzuzeigen und diese durch einen Suchbegriff zu filtern.

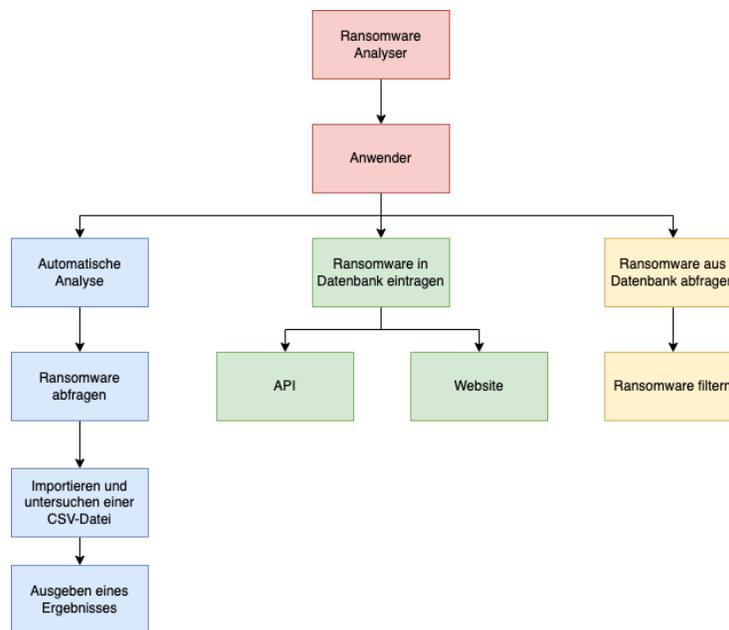


Abbildung 11: Systemarchitektur des Prototyps aus funktionaler Sicht

Aus Implementierungssicht besonders herausfordernd waren die Entwicklung der Analyse-Komponente und die Entwicklung einer API-Schnittstelle. Diese beiden Aspekte werden daher in den nachfolgenden Kapiteln detaillierter behandelt.

### 5.3 Implementierung der automatischen Analyse

Ist es für den Nutzer nicht möglich, den Namen der Ransomware aufgrund der Dateiendungen oder des Namens herauszufinden, wird eine automatische Analyse bereitgestellt. Diese gleicht die Dateien auf der Festplatte mit den bekannten Dateiendungen aus der Datenbank ab und versucht so, eine Annahme darüber

zu treffen, um welche Ransomware es sich handelt. Dazu stellt Autopsy eine Möglichkeit bereit, eine CSV-Datei zu exportieren, welche alle Dateien mit deren Dateiendung vom zu analysierenden System enthält. Der Pfad zu dieser Datei muss auf der Website angegeben werden, anschließend wird lokal vom Browser eine Analyse durchgeführt, diese Analyse findet also nicht auf dem Server statt. Um die Analyse durchzuführen wird zunächst beim Aufruf der Seite wie in Listing 3 gezeigt, eine Abfrage der verschiedenen Ransomware-Typen von der Datenbank vorgenommen. In Listing 4 anschließend an die Funktion, durch den Parameter *props* übergeben, die den Seitenaufbau durchführt.

```
1 export async function getServerSideProps() {
2   const rans = await prisma.type.findMany();
3   return {
4     props: {
5       rans,
6     },
7   };
8 }
```

Listing 3: Abfrage der Ransomware-Typen vom Server

```
1 export default function Home(props) {
```

Listing 4: Beginn des Seitenaufrufs

Zum Parsen der CSV-Datei wird das Paket *papaparse* verwendet. Auf der Website befindet sich ein Ablagebereich für die CSV-Datei des Nutzers. Wird die Datei dort per Drag and Drop abgelegt oder durch einen Klick der Pfad angegeben, wird die Funktion ausgeführt, die man in Listing 5 sieht. Zunächst wird dem Nutzer ein Text und eine Progress Bar eingeblendet. Anschließend wird *papaparse* die Datei übergeben und in einer geschachtelten for-Schleife analysiert. Die Funktionalität der Schleifen wird in Grafik 12 gezeigt.

Es wird für jeden Eintrag in der Datenbank jede mögliche Dateiendung und jede Lösegeldforderung aus der CSV-Datei des Nutzers überprüft. Da eine Ransomware mehrere Dateiendungen haben kann und jede einzeln geprüft werden muss, sind hier drei for-Schleifen ineinander geschachtelt. Weiterhin kann eine Ransomware unterschiedliche Namen für eine Lösegeldforderung haben, auch das wird geprüft. Um die Erkennung zu verbessern, wird mit höchster Priorität geprüft, ob sowohl die passende Dateiendung, als auch die Lösegeldforderung gefunden werden. Ist das der Fall, wird die Variable *editable* auf *false* gesetzt. Somit endet die for-Schleife und die Variable mit dem Namen der Ransomware (*ran*) wird nicht mehr verändert. Es gibt Ransomware, die keine Lösegeldforderung erstellt oder es ist keine in der Datenbank hinterlegt. In diesem Fall beruft sich das Tool nur auf die Dateiendung, das Ergebnis ist dadurch weniger präzise, da manche Ransomware-Typen gleiche Dateiendungen haben. Weiterhin dauert der Durchlauf der Schleifen länger, da nicht an einem bestimmten Punkt abgebrochen werden kann. Es wäre schließlich möglich, zu einem späteren Zeitpunkt noch ein passendes Paar aus Lösegeldforderung und Dateiendung zu finden. Wurde ein passendes Paar gefunden oder die Schleifen sind bis zum Ende

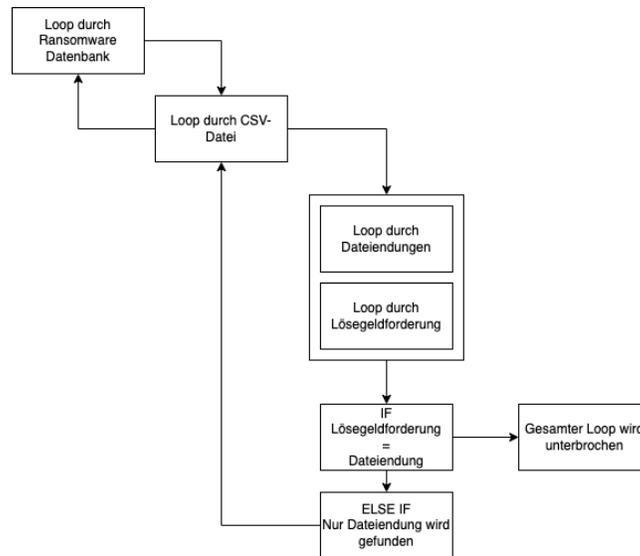


Abbildung 12: Ablauf der automatischen Analyse

durchgelaufen, wird dem Nutzer mitgeteilt, welche Ransomware gefunden wurde. Es wird außerdem ein Link bereitgestellt durch den mehr Informationen zur Ransomware gefunden werden können. Sollte am Ende der Schleife keine Ransomware erkannt worden sein, wird auch das dem Nutzer mitgeteilt.

```

1 const parseFile = (file) => {
2   setProg(10);
3   setIsVisible(true);
4   setParsedCsvData(
5     "Analyse gestartet, dies kann ein paar Minuten dauern. Das
6     Ergebnis wird Ihnen im Anschluss hier angezeigt. Bitte
7     verlassen oder aktualisieren Sie die Seite nicht."
8   );
9   Papa.parse(file, {
10    header: true,
11    complete: (results) => {
12      let mail = "Folgende Ransomware wurde gefunden: ";
13      let ran = "";
14      let list = "";
15      let fileEnd = "";
16      let editable = true;
17      let loesegeldforderung = "";
18      for (const i of props.rans) {
19        if (editable) {
20          for (const y of results.data) {
21            //Pruefen der Dateieindungen
22            for (let x = 0; x < i.dateieindung.split("\r\n").
length; x++) {
                if (
                    i.dateieindung.split("\r\n")[x].replaceAll(" ", "")

```

```

23 ) != "" &&
24     y["File Extension"] != null
25     ) {
26     try {
27         if (
28             y["File Extension"] ==
                i.dateiendung.split("\r\n")[x].replaceAll("
", "") &&
29                 !fileEnd.includes(i.name)
30             ) {
31                 fileEnd = i.name;
32                 if (i.loesegeldforderung == "") {
33                     list = list + i.name + " ";
34                 }
35             }
36         } catch (e) {
37             console.log(e);
38         }
39     }
40     //Pruefen der Loesegeldforderung
41     for (
42         let x = 0;
43         x < i.loesegeldforderung.split("\r\n").length;
44         x++
45     ) {
46         try {
47             if (
48                 y["Name"] ==
49                 i.loesegeldforderung
50                 .split("\r\n")
51                 [x].replaceAll(" ", "") &&
52                 !loesegeldforderung.includes(i.name) &&
53                 i.loesegeldforderung != ""
54             ) {
55                 loesegeldforderung = i.name;
56             }
57         } catch (e) {
58             console.log(e);
59         }
60     }
61
62     //Auswertung
63     if (loesegeldforderung == fileEnd && fileEnd != "")
64     {
65         ran = fileEnd;
66         editable = false;
67         console.log("Editable false!");
68     } else if (
69         i.loesegeldforderung == "" &&
70         editable &&
71         i.fileEnd != ""
72     ) {
73         ran = fileEnd;
74     }
75 }
76 }

```

```

77     }
78
79     setProg(100);
80     if (editable == false) {
81         setRanName(ran);
82     } else {
83         setRanName(list);
84     }
85     if (ran == "") {
86         setRanName("nichts gefunden");
87     }
88     setParsedCsvData(mail);
89 },
90 });
91 };

```

Listing 5: Automatische Analyse

## 5.4 Implementierung einer API-Schnittstelle

Um das Ransomware-Analyse-Tool in Zukunft leicht erweiterbar zu machen, wurde es mit einer API-Schnittstelle ausgestattet, über die einfach und schnell weitere Ransomware-Typen hinzugefügt werden können. Die Datei *createRansome.js* befindet sich in folgender Ordnerstruktur: *pages/api/createRansome.js*. Dadurch, dass die Datei im *api*-Ordner abgelegt wird, erzeugt NextJS automatisch einen API-Endpunkt, der über *Server-URL/api/createRansome* erreichbar ist. Wird ein POST-Request an diese Schnittstelle geschickt, wird ein neuer Eintrag in der Tabelle mit den verschiedenen Ransomware-Typen vorgenommen und somit wird die Ransomware auch in der automatischen Analyse gefunden. Der Body des Requests muss aussehen wie in Listing 6.

```

1 {
2   "name": "Test",
3   "dateiendung": ".test",
4   "loesegeldforderung": "test.txt",
5   "merkmale": "Das ist ein Test",
6   "verschluesselung": "256Bit",
7   "decryptor": "test",
8   "link": "test.de"
9 }

```

Listing 6: Beispiel JSON-Input für Schnittstelle

Der Schnittstelle muss ein Name für die Ransomware mitgegeben werden. Die anderen Parameter wie Dateiendung, Lösegeldforderung, Merkmale, Verschlüsselung, der Decryptor und der Link zum Decryptor sind optional.

Implementiert wurde das ganze wie im Code in Listing 7. Zunächst muss Prisma importiert werden, um auf die Datenbank zugreifen zu können. Anschließend wird eine Funktion aufgerufen, welche den Request-Body übergeben bekommt. In der Funktion wird zunächst geprüft, ob es sich wirklich um einen POST-Request handelt, andernfalls bekommt der User eine Fehlermeldung zurückgegeben. Handelt es sich um einen POST-Request, werden im nächsten Schritt die

einzelnen übergebenen Werte aus dem Request-Body in Variablen gespeichert, dies geschieht von Zeile 10 bis 19. Anschließend wird in Zeile 21 durch *await prisma.type.create()* ein neuer Datenbankeintrag erzeugt. Beim Aufruf der Funktion steht das *type* für den Name der Tabelle aus der Prisma Schema Datei und *create()* dafür, dass ein Eintrag erzeugt werden soll. Es gibt neben *create()* auch noch *delete*, *find*, *findMany* und *update*, welche zum Löschen, Suchen oder Modifizieren von Datensätzen verwendet werden können. In dem *create*-Befehl werden die einzutragenden Daten im JSON-Format mitgegeben. Hiermit ist der Datenbankeintrag erzeugt. Letztendlich wird in Zeile 33 eine Antwort auf den POST-Request gegeben und der Nutzer erhält die eingetragenen Daten inklusive der automatisch vom Server generierten ID zurück. Wie so ein Eintrag in der MongoDB-Datenbank aussieht, sieht man in Grafik 9.

```
1 import prisma from "../../lib/prisma";
2
3 export default async (req, res) => {
4   if (req.method !== "POST") {
5     return res.status(405).json({ message: "Method not allowed" });
6   }
7
8   console.log(JSON.stringify(req.body));
9
10  const data = req.body;
11  const {
12    name,
13    dateiendung,
14    loesegeldforderung,
15    merkmale,
16    verschluesselung,
17    decryptor,
18    link,
19  } = data;
20
21  const ransome = await prisma.type.create({
22    data: {
23      name: name,
24      dateiendung: dateiendung,
25      loesegeldforderung: loesegeldforderung,
26      merkmale: merkmale,
27      verschluesselung: verschluesselung,
28      decryptor: decryptor,
29      link: link,
30    },
31  });
32
33  res.json(ransome);
34  };
```

Listing 7: Implementierung API-Schnittstelle

## 6 Test des Prototypen

Um die Software auf die gegebenen funktionalen Evaluierungskriterien und jene, die mit der Benutzbarkeit zusammenhängen, zu testen, kamen sowohl offene Tests zum Einsatz als auch Tests basierend auf Hypothesen.

Es ist wichtig, die richtige Testtechnik anzuwenden um beim Testen der unterschiedlichen Qualitätsmerkmale die richtigen Ergebnisse zu erhalten. Im Vordergrund zum Testen der Funktionalität stehen die White Box-Tests, diese konzentrieren sich auf die Erkennung logischer Fehler im Code. Die Auswahl der Testfälle ergibt sich dabei aus der Implementierung einzelner Komponenten. Es werden Testfälle entworfen, bei denen die interne Logik des Programms und ihre Verzweigungen getestet werden. Der Programmierer muss dabei eine vollständige Kenntnis über die Struktur des Codes haben um die Testfälle zu entwerfen und er muss wissen, was das Programm tun soll und ob es von der geplanten Funktionalität abweicht. [33] Der Einfachheit halber wurden diese in hypothesenbasierte Tests zum Testen der Funktionalität und Usability Tests zum Testen der Einsetzbarkeit unterteilt.

### 6.1 Usability Tests

#### 6.1.1 Kann die Analyse von einem Menschen mit guten IT-Kenntnissen durchgeführt werden?

Um diese Frage zu beantworten, wurde ein offener Test durchgeführt, bei dem der Nutzer eine mit WannaCry infizierte Virtuelle Maschine bekommen hat und den Link zur Startseite des entwickelten Tools. Als Testperson wurden Geschäftsführer von kleinen Unternehmen herangezogen, die nach eigenen Angaben auch für die IT in ihrem Unternehmen zuständig sind. Sie sollten also über gute IT-Kenntnisse verfügen.

Ihre Aufgabe war es mit Hilfe des Tools die virtuelle Maschine zu analysieren und folgende Fragen zu beantworten:

- Wie wurde der Angriff gestartet?
- Wann hat der Angriff stattgefunden?
- Wie ist der Angreifer auf das System gekommen?
- Welche Dateien sind betroffen?

#### **Auswertung:**

Mit einer Pilotgruppe durchgeführte Tests haben ergeben, dass eine Analyse durch eine Person mit guten IT-Kenntnissen durchgeführt werden kann. Die Analyse benötigt jedoch erheblichen Zeitaufwand und eine detaillierte Beschreibung des Vorgehens. Nicht immer konnten der Angriffszeitpunkt oder der Tathergang rekonstruiert werden, jedoch haben alle Probanden herausgefunden, um welche Ransomware es sich auf dem infizierten System handelte. Weiterhin sind alle Testpersonen auf den Decryptor aufmerksam geworden, der im System

hinterlegt ist. Es wäre ihnen so zumindest möglich, die verschlüsselten Daten wiederherzustellen.

### **6.1.2 Wie wird die Benutzerfreundlichkeit eingeschätzt?**

Um die Benutzerfreundlichkeit zu bewerten, wurden die Probanden aus Kapitel 6.1.1 mit folgendem Ergebnis befragt:

- Für den Benutzer ist eine zu detaillierte Beschreibung mit vielen Fachbegriffen unvorteilhaft. Es führt zu Verwirrung, wenn z.B. Worte wie *Live Response* oder *Post Mortem Analyse* verwendet werden. Die Beschreibung sollte so einfach wie möglich gehalten und Vokabular verwendet werden, das auch für einen Laien gut verständlich ist.
- Die Mehrheit der Probanden hat nicht die Analyse mit Autopsy verwendet um den Typ der Ransomware ausfindig zu machen, sondern die Tabelle mit den Dateieindungen genutzt und kam so schon nach wenigen Minuten auf ein erstes Ergebnis
- Die Probanden waren mit der Benutzbarkeit zufrieden. Sie haben etwas Einarbeitungszeit benötigt, kamen jedoch mit der Menüführung und der Beschreibung des Vorgehens zurecht.
- Die Probanden haben wenig Wert auf den Tatzeitpunkt oder den Ablauf der Tat gelegt, wichtig war nur, die Daten wiederherzustellen um weiter arbeiten zu können.

### **6.1.3 Eignet sich der Prototyp für kleine und mittelständische Unternehmen?**

Ziel des Tests ist es herauszufinden, ob es einem Menschen mit guten IT-Kenntnissen und günstiger bis kostenloser Software möglich ist eine Analyse durchzuführen. Nicht jedes kleine und mittelständische Unternehmen verfügt über ein großes Budget oder gar einen ausgebildeten IT-Forensiker.[30] Daher ist es notwendig, die anfallenden Kosten so gering wie möglich zu halten. Dass die Analyse von Menschen mit guten IT-Kenntnissen durchgeführt werden kann, wurde bereits in Kapitel 6.1.1 gezeigt. Die entwickelte Software empfiehlt weiterhin die Verwendung der Open-Source-Software Autopsy - Digital Forensics, welche kostenlos ist. Die Analyse ist somit für kleine und mittelständische Unternehmen durchführbar, solange es im Unternehmen eine Person mit guten IT-Kenntnissen gibt.

### **6.1.4 Ist der Prototyp auf unterschiedlichen Betriebssystemen lauffähig?**

Um diese Frage zu beantworten wurde auf einen offenen Test zurückgegriffen. Der Prototyp wurde auf verschiedenen Betriebssystemen in verschiedenen Browsern aufgerufen und eine automatische Analyse durchgeführt. Die Ergebnisse finden sich in Tabelle 3.

Betriebssystem	Browser	Funktionalität
Windows 11	Chrome	✓
Windows 11	Firefox	✓
MacOS Ventura	Safari	✓
MacOS Ventura	Chrome	✓
MacOS Ventura	Firefox	✓
Linux	Firefox	✓
Linux	Chrome	✓
Android	Chrome	✓
iOS 16.1.1	Safari	✓

Tabelle 3: Betriebssysteme & Browser

Der Prototyp ist somit auf allen aktuellen Betriebssystemen in den meist genutzten Browsern lauffähig. Eine Einschränkung bietet jedoch Autopsy, was zur Analyse benötigt wird. Es ist nur für Desktop Betriebssysteme wie MacOS, Linux oder Windows verfügbar.

### 6.1.5 Welche Bereiche kann das System?

Wie in den vorherigen Tests gezeigt, ist das System ausgelegt für die Analyse von Ransomware-Angriffen. Es ist durch weitere Typen von Ransomware erweiterbar wie in Kapitel 6.2.3 gezeigt. Das System kann nicht, durch andere Typen von Malware erweitert werden. Die Beschreibung der Analyse lässt sich womöglich auf andere Malware übertragen, die automatische Analyse müsste jedoch mit erheblichem Aufwand angepasst werden, um andere Schadsoftware zu erkennen.

## 6.2 Hypothesenbasierte funktionale Tests

### 6.2.1 Welcher Typ von Ransomware wird bei der automatischen Analyse erkannt?

**Hypothese: Es wird angenommen, dass das System alle in der Datenbank abgelegten Typen von Ransomware erkennt. False Positivs können nur dort auftreten wo der Dateiname einem "legalen"**

**Dateinamen entspricht, wie z.B. readme.txt.**

Die automatische Analyse basiert auf den CSV-Dateien, die aus Autopsy exportiert werden können. Alle Dateien des Systems, inklusive ihrer Dateiendungen, sind in dieser Datei enthalten. Die Datenbank der entwickelten Software enthält über 660 verschiedene Typen von Ransomware. In diesem Test gab es zwei Fälle zu berücksichtigen. Zum einen Ransomware, bei der sowohl eine Lösegeldforderung, als auch eine Dateiendung hinterlegt sind, sowie bei der nur eine Dateiendung in der Datenbank hinterlegt ist.

Zum Testen wurde zum einen ein Export eines nicht infizierten Systems herangezogen. Dieser Export wurde anschließend verändert, indem zum einen die Dateiendung einer durch eine Ransomware verschlüsselte Datei eingefügt wurde. Bei Ransomware, die sowohl eine Dateiendung, als auch eine Lösegeldforderung in der Datenbank hinterlegt hat, wurde beides in der CSV-Datei präpariert. Anschließend wurde die automatische Analyse auf die in Tabelle 4 gezeigten Ransomware-Typen getestet. Im Test wurde jede Ransomware fehlerfrei erkannt. Weiterhin konnte festgestellt werden, dass die Analyse mit einem positiven Ergebnis abbricht, wenn eine passende Kombination aus Lösegeldforderung und Dateiendung gefunden wird. Ein Spezialfall bildet hierbei WannaCry, da hierbei ein Windows-System mit der Ransomware infiziert und basierend auf dieser Analyse, der Test durchgeführt wurde. Das Ergebnis des Tests ist positiv, da jede Ransomware im Test erkannt wurde und sich das Programm wie erwartet verhalten hat.

4rw5w	✓
Dusk	✓
M3TW0RM	✓
MM Locker	✓
Monkey	✓
Nefilim	✓
Ransomwared	✓
Rector	✓
ucyLocker	✓
UDLA	✓
WannaCry	✓

Tabelle 4: Getestete Ransomware

**Aufgrund der durchgeführten Tests kann angenommen werden, dass die aufgestellte Hypothese valide ist.**

### 6.2.2 Wie hoch ist die Trefferquote bei der automatischen Analyse?

**Hypothese:** Unter der Annahme, dass die Einträge der Datenbank vollständig sind, sollte es keine False Negativs geben. Als Ausgangshypothese gilt daher, dass nur dann False Negativs auftreten können, wenn keine entsprechenden Datenbankeinträge vorhanden sind. False Negativs sollten auf jene Fälle beschränkt bleiben, in denen identische Dateinamen oder Dateiendungen auch von "legalen" Programmen verwendet werden.

#### Fall 1: *False Negativ*

Um eine Ransomware zu erkennen, muss sie in der Datenbank abgelegt sein. Es

muss dort mindestens ein Name und die Dateierdung der verschlüsselten Dateien hinterlegt sein, im besten Fall ist auch der Dateiname der Lösegeldforderung hinterlegt. Sind diese Daten nicht vorhanden, ist die Ransomware der automatischen Analyse nicht bekannt und kann somit nicht identifiziert werden.

### Fall 2: *False Positiv*

Zum prüfen dieses Falls wurde zu Testzwecken ein neuer Datenbankeintrag in die Datenbank eingefügt. Dieser war aufgebaut wie in Listing 8 zu sehen.

```
1 {
2   "name": "Test",
3   "dateiendung": ".psd",
4   "loesegeldforderung": "readme.txt",
5   "merkmale": "Das ist ein Test",
6   "verschlueselung": "256Bit",
7 }
```

Listing 8: Demo-Input für False Positiv

Anschließend wurde eine Analyse durchgeführt von einem Windows-System auf dem sich Photoshop mit einer .psd-Datei befand. Auf dem Desktop wurde eine *readme.txt* Datei abgelegt, wie sie viele Programme erzeugen. Das Tool hat die Test-Ransomware gefunden und als Ergebnis ausgegeben. Verwendet eine Ransomware also eine Dateierdung eines "legalen"Programms und eine Lösegeldforderung mit einem Namen, der sich häufig findet, so kann es zu einem falsch positiven Ergebnis kommen.

**Es kann angenommen werden, dass die aufgestellte Hypothese valide ist.**

### 6.2.3 Ist der Prototyp für die Zukunft erweiterbar?

**Hypothese: Der Prototyp ist um weitere Ransomware erweiterbar, unter Berücksichtigung, dass das Ziel der Masterarbeit war, Ransomware und keine weitere Malware zu klassifizieren.**

Der Prototyp ist in sofern für die Zukunft erweiterbar, dass weitere Datensätze mit Ransomware-Typen über eine API-Schnittstelle in die Datenbank eingefügt werden können. Wie genau dieser Vorgang abläuft wird in Kapitel 5.4 erläutert. Um dies zu testen, wird in einem White Box-Test eine Anfrage mit Postman an den API-Endpunkt */api/createRansome* gesendet. Die zu übermittelnden Daten sind in Listing 6 einsehbar. Erwartet wird, dass ein neuer Eintrag auf der Seite */types* im Prototyp zu sehen ist. Weiterhin sollte der neue Eintrag in der automatischen Analyse berücksichtigt werden.

Der Test ist wie erwartet verlaufen. Der Eintrag ist sowohl auf der Website zu sehen, als auch in der automatischen Analyse berücksichtigt.

Eine Einschränkung des Prototypen besteht allerdings in der Erweiterbarkeit in Bezug auf Erkennung anderer Schadsoftware. Es ist nicht vorgesehen, dass

die Software erweitert werden kann, um zum Beispiel bei der Analyse von DDoS- oder APTs-Angriffen zu unterstützen oder diese zu erkennen. Die entwickelte Software ist ausschließlich für Ransomware-Angriffe geeignet.

Aufgabe und Ziel dieser Masterarbeit war die Unterstützung der Klassifikation von Ransomware-Angriffen. Das Konzept ist auch auf andere Typen von Malware anwendbar, der tabellengesteuerte Prototyp ist jedoch nicht darauf ausgelegt. Im Falle einer Erweiterung müssten Input-Tabelle und darauf basierende Code-Elemente entsprechend angepasst werden, was mit einem erheblichen Aufwand verbunden ist.

**Aufgrund der durchgeführten Tests kann angenommen werden, dass die aufgestellte Hypothese valide ist.**

#### 6.2.4 Wie hoch ist die algorithmische Komplexität?

**Aufgrund zwei ineinander geschachtelter Schleifen ist die Ausgangshypothese, dass die Run Time Complexity im Bereich von  $O(n^2)$  liegt.**

Um die Laufzeit einzuschätzen gibt es zwei Ansätze. Zum einen kann die Laufzeit eines Algorithmus gemessen werden, diese Methode ist jedoch sehr ungenau, da sie sehr stark vom Rechner, der Rechnerlast und Konfiguration abhängt. Eine weitere Möglichkeit ist das Zählen der Elementaroperationen in Abhängigkeit der Größe der Eingabe. Elementaroperationen sind beispielsweise Vergleiche, Arrayzugriffe oder Zuweisungen.[16]

Um diese zu zählen wurde das Programm leicht modifiziert, um in einer Variable die durchgeführten Operationen festzuhalten. Weiterhin wurde die CSV-Datei eines Systems gewählt, das nicht infiziert ist, damit der Algorithmus nicht abbricht, sondern bis zum Ende durchläuft.

Der Test basiert auf **660 Einträgen in der Datenbank** und 9 verschiedenen CSV-Dateien, die in Tabelle 5 gezeigt werden. Eine Besonderheit bildet hier CSV 3, da diese den Export eines neu aufgesetzten Windows 11 Systems darstellt.

Test	Zeilen	Elementaroperationen
CSV 1	10.000	35.700.010
CSV 2	100.000	356.992.877
CSV 3	130.888	467.273.747
CSV 4	200.000	713.982.167
CSV 5	400.000	1.427.967.887
CSV 6	600.000	2.141.953.607
CSV 7	800.000	2.855.996.440
CSV 8	1.000.000	3.569.996.440
CSV 9	1.200.000	4.283.910.767

Tabelle 5: Laufzeitanalyse

Besser ersichtlich wird die Run Time Complexity durch Grafik 13. Diese zeigt ein

beinahe lineares Wachstum, was damit zusammen hängt, dass nur die Anzahl der Zeilen in der CSV-Datei zunimmt, die Einträge in der Datenbank, die die zu prüfenden Ransomware-Typen darstellen, verändern sich jedoch nicht.

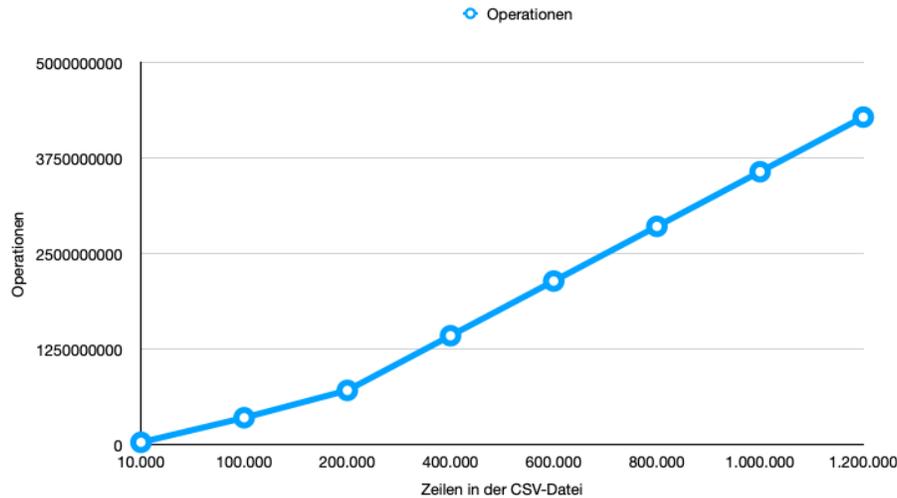


Abbildung 13: Laufzeitanalyse

Aufgrund der durchgeführten Tests kann angenommen werden, dass die aufgestellte Hypothese valide ist.

## 6.3 System Limitations

### 6.3.1 Direkte Abhängigkeit der Storage Complexity vom Umfang und Struktur der eingegebenen Daten

Die Storage Complexity ist abhängig von der Größe und dem Aufbau der zu prüfenden CSV-Datei, i.e. von der Anzahl der Spalten und Zeilen.

### 6.3.2 Grenzen der Detektion

Um eine Ransomware zu erkennen, muss sie in der Datenbank, die aktuell 660 Einträge enthält, vorhanden sein. Ist die Ransomware dort nicht vorhanden oder ist in der Datenbank keine Dateiendung für die Ransomware hinterlegt, kann das System sie nicht erkennen. Weiterhin kann das Ergebnis nicht eindeutig sein, wenn eine Ransomware die gleiche Dateiendung hat und keine Lösegeldforderung. Zu Unstimmigkeiten kann es außerdem kommen bei gleicher Dateiendung und einer Lösegeldforderung mit identischen Namen. Ist das Ergebnis nicht eindeutig, so werden alle möglichen Arten von Ransomware aufgelistet, um dem Anwender eine Auswahl an möglichen Typen zu geben.

**Fall 1:** *Die Ransomware ist nicht in der Datenbank vorhanden*

Um dies zu testen wurde die WannaCry Ransomware aus der Datenbank gelöscht und die CSV-Datei aus Kapitel 6.2.1 zur Analyse herangezogen. Die CSV-Datei enthält den Export aller Dateien eines mit WannaCry infizierten Systems. Nach dem Durchlauf der Analyse gibt das System aus, dass keine Ransomware gefunden wurde.

**Fall 2:** *Die Dateiendung ist nicht eindeutig*

Dieser Fall wurde getestet, indem in den CSV-Export eines nicht infizierten Systems die Dateiendung *.encrypted* eingefügt wurde. Das Ergebnis der Analyse sieht man in Grafik 14.



Folgende Ransomware wurde gefunden: Jigsaw KeRanger PassLock

Abbildung 14: Ergebnis Analyse mit *.encrypted*

Gefunden wurden drei mögliche Typen von Ransomware, da für diese in der Datenbank keine Lösegeldforderung hinterlegt ist, jedoch die Dateiendung *.encrypted*. In diesem Fall kann also keine klare Aussage getroffen werden, der Nutzer erhält jedoch eine Spur.

## 7 Diskussion der Ergebnisse

In Kapitel 3 wurden verschiedene Leitlinien beschrieben, die als Orientierung bei der Entwicklung der Artefakte dienten. Die Umsetzung wird im folgenden diskutiert.

### 7.1 Leitlinie 1: Design als Artefakt

In Leitlinie 1 wurde unter anderem beschrieben, dass das Artefakt an eine bestimmte Organisation angepasst werden muss um angewendet zu werden. Gerade hier bestand die Herausforderung darin, das Artefakt nicht auf eine Organisation auszurichten sondern die Anwendung möglichst offen zu halten, sodass es für alle KMUs funktioniert. Daher ist das entwickelte Artefakt keine Software, die auf Windows oder Linux installiert wird. Es wurde in Form einer Website konzipiert, sodass es von allen Plattformen aus über einen Browser aufgerufen werden kann. Auch Autopsy ist sowohl auf Windows, als auch Linux und MacOS lauffähig, so kann ein möglichst breites Spektrum an Geräten abgedeckt werden. Das Problem, dass es im Rahmen dieser Arbeit zu lösen gab, war es, kleinen und mittelständischen Unternehmen zu ermöglichen, eine Analyse ohne einen ausgebildeten IT-Forensiker durchzuführen. Die entwickelte Software sollte daher möglichst kostengünstig sein, in diesem Fall ist die Verwendung des entwickelten Artefakts kostenlos. Auch die Verwendung von Autopsy ist kostenlos, da es sich dabei um Open-Source Software handelt, genau wie die Verwendung von AOMEI Partition Assistant welche für die 1:1 bitgenaue Kopie benötigt wird, auch wenn diese für kommerzielle Verwendung lizenziert werden muss. Das entwickelte Artefakt beschreibt die Anwendung von Autopsy, um die Analyse einer Ransomware durchzuführen. Die Erkennung des Ransomware-Typs konnte durch das Tool automatisiert werden. Es greift auf eine Datenbank zurück in der verschiedene Ransomware-Typen mit deren Dateieindungen gespeichert sind und gleicht diese mit einem Export aller Dateien auf der zu analysierenden Festplatte ab. Ein geeigneter Export kann dabei von Autopsy erstellt werden, dieser muss in das Analyse-Tool eingefügt werden. Die Analyse einer Ransomware wird für den Nutzer somit so einfach wie möglich gestaltet, der Anwender sollte aber über gute IT-Kenntnisse verfügen, um die Analyse durchzuführen. Der Analyseprozess ist zwar gut beschrieben aber trotzdem noch nicht trivial.

### 7.2 Leitlinie 2: Problemrelevanz

Bei der Entwicklung der Software war das Hauptziel, eine Lösung für das Problem zu schaffen, dass sich viele kleine Unternehmen keinen teuren gut ausgebildeten IT-Forensiker leisten können. [30] Daraus folgen bei der Entwicklung des Tools sowohl Zielkriterien als auch Beschränkungen. Die Zielkriterien sind aufgeführt in Tabelle 6 und die Beschränkungen finden sich in Tabelle 7. Alle gegebenen Ziele konnten erreicht werden, genau wie alle Beschränkungen berücksichtigt wurden. Das Tool unterstützt bei der Post Mortem Analyse und kann mit Hilfe der bereitgestellten Informationen einem Menschen mit guten

IT-Kenntnissen ermöglichen, eine Analyse mit Autopsy selbst durchzuführen. Die Analyse läuft dabei nicht voll automatisch ab. Da jede Ransomware sich anders verhält und es immer mehr neue Ransomware-Typen gibt, ist eine volle Automatisierung nicht möglich. Die Erkennung, um welche Ransomware es sich handelt, konnte automatisiert werden, jedoch muss diese in der Datenbank eingetragen sein. Weiterhin ist die umgesetzte Software eine Website, wodurch sie auf vielen Betriebssystemen läuft, es wird lediglich ein Browser benötigt um diese aufzurufen.

Die Analyse ist eine Post Mortem Analyse	✓
Sie muss für Menschen mit guten IT-Kenntnissen durchführbar sein	✓
Die Analyse-Software soll auf möglichst vielen Betriebssystemen laufen	✓
Es darf keine gebührenpflichtige Software verwendet werden	✓
Die Datenbank der Ransomware-Typen soll erweiterbar sein um neue Ransomware zu erkennen	✓

Tabelle 6: Zielkriterien

Die Analyse ist keine Live-Response Analyse. Diese bietet zwar den Vorteil, dass sie auch flüchtige Daten aus dem RAM des Computers bei der Analyse berücksichtigt oder Informationen darüber geben kann, welche Programme aktuell laufen, ist jedoch für einen Laien schwer durchzuführen. Hier kommt der große Vorteil der Post Mortem Analyse, in der das System heruntergefahren wird und die Analyse ohne Zeitdruck durchgeführt werden kann. Eine Analyse eines Systems durchzuführen ist auch mit Hilfe des Tools ein komplexes Vorhaben, weshalb es einen Menschen mit guten IT-Kenntnissen braucht. Weiterhin war vorgegeben, die Analyse nur mit kostenlosen Tools durchzuführen, um den Unternehmen keine Kosten entstehen zu lassen. Verwendet wurde daher die Software *Autopsy - Digital Forensics* zur Analyse des Systems und *AOMEI Partition Assistant* zum erstellen der 1:1 bitgenauen Kopie der Festplatte. Die Erkennung von Cyber-Angriffen, die keine Ransomware ist, wurde außerdem ausgeschlossen. Das Tool dient nur zur Analyse dieser Angriffsart.

Es kann nicht an einem laufenden System gearbeitet werden	✓
Die Analyse ist für Menschen mit schlechten IT-Kenntnissen nur sehr schwer durchführbar	✓
Die Analyse muss mit kostenloser Software durchführbar sein	✓
Die Software kann nicht bei der Analyse anderer Cyberangriffe eingesetzt werden und beschränkt sich somit ausschließlich auf Ransomware	✓

Tabelle 7: Beschränkungen

### **7.3 Leitlinie 3: Bewertung des Designs**

In Leitlinie 3 werden einige Kriterien definiert, die zur Evaluierung des Prototypen beitragen. Diese werden benötigt um den Nutzen und die Qualität des Artefakts zu zeigen. Das Unternehmensumfeld oder das geplante Einsatzgebiet definieren dabei die Anforderungen an das fertige Produkt. Sind alle Anforderungen des Problems erfüllt, unter Berücksichtigung der gegebenen Zielkriterien und Einschränkungen, so kann das Design als vollständig und effektiv bezeichnet werden. Getestet wurden die Kriterien in Form von Usability Tests und funktionalen hypothesenbasierten Tests, wie in Kapitel 6 zu sehen.

### **7.4 Leitlinie 4: Beiträge zur Forschung**

Leitlinie 4 stellt die Frage nach den neuen interessanten Fakten, die das Projekt zur Forschung beiträgt. Es wurde weder eine neue Methode zur Bewertung des Designs entwickelt, noch eine neue Grundlage, wie ein Modell. Es wurde ein neues Artefakt designt, welches ein bisher ungelöstes Problem behebt. Es gibt zwar Websites, die die Verwendung von Autopsy beschreiben, jedoch konnte während der Recherche keine gefunden werden, die eine so detaillierte Hilfestellung bei der Analyse von Ransomware liefert. Ebenso gab es kein Tool, das eine automatische Erkennung des Ransomware-Typs ermöglicht. Die interessante Neuheit dieses Projekts ist somit das entwickelte Artefakt.

### **7.5 Leitlinie 5: Strenge der Forschung**

Das Artefakt hat bei der Entwicklung einen iterativen Prozess durchlaufen, durch den es sich immer weiter den Bewertungskriterien angepasst hat, mehr dazu findet sich in Kapitel 6. So konnte durch White Box-Tests mit ausgewählten Probanden ein Artefakt entwickelt werden, was die geforderten Kriterien möglichst gut erfüllt. Dabei wurde zunächst eine Wissensbasis über schon vorhandene Technologien und Methoden geschaffen, welche dann möglichst effektiv zur Umsetzung des Projekts eingesetzt worden sind. Angefangen bei der Konstruktion des Prozesses mit BPMN bis hin zur Umsetzung und Implementierung mit NextJS, Prisma und MongoDB.

### **7.6 Leitlinie 6: Design als Suchprozess**

Nach Leitlinie 6 ist ein optimales Ergebnis für ein Problem nicht zu erreichen, jedoch kann sich in einem iterativen Prozess einer optimalen Lösung angenähert werden. Dieser Prozess nennt sich der Generate und Test Zyklus, der in Grafik 3 zu sehen ist. Dabei wurden Mittel, Ziele und Einschränkungen ermittelt und ein erster Prototyp entwickelt. Zu Beginn basierte die automatische Analyse zum Beispiel nur auf den Dateieindungen der verschlüsselten Dateien. Diese ähneln sich jedoch sehr oft oder sind identisch und können das Ergebnis verfälschen. Um diesem Problem entgegen zu wirken, wurde in einem weiteren Zyklus die Funktionalität erweitert, indem auch die Lösegeldforderung der Ransomware mit in

die Analyse einbezogen wurde. Ähnlich hat sich die Nutzerfreundlichkeit durch mehrere Zyklen verbessert. Durch die Beobachtung, wie verschiedene Nutzer den Prototypen verwenden, konnte festgestellt werden, dass die Menüstruktur angepasst werden muss um den Nutzer besser durch die Analyse zu führen. Durch mehrere Iterationen, in denen immer wieder Anpassungen vorgenommen worden sind, kann nun ein Prototyp bereitgestellt werden, der für das gegebene Problem mit den Einschränkungen und Zielkriterien aus Kapitel 3.3 gut funktioniert.

## **7.7 Leitlinie 7: Kommunikation der Forschung**

Leitlinie 7 beschreibt die Kommunikation der Ergebnisse nach außen. Die erarbeitete Lösung für das gegebene Problem muss präsentiert werden. Dabei ist zu beachten, dass es sowohl für eine technologieorientierte, als auch für eine managementorientierte Zielgruppe präsentiert werden muss. Da das Management entscheidet, ob das entwickelte Artefakt in deren Unternehmen zum Einsatz kommt und Forscher die geschaffene Wissensbasis nutzen können, um darauf basierend weitere Artefakte zu entwickeln. Die Problemstellung und der aktuelle Stand in der Wissenschaft dieser Masterarbeit sind bereits im Rahmen des Masterseminars präsentiert worden. Die Funktionalität des Prototypen wurde lokal getestet. Die Ergebnisse der Masterarbeit können im nächsten Schritt auf einer Konferenz präsentiert werden.

## 8 Conclusio

Wie aus der im Rahmen der Masterarbeit durchgeführten Literaturanalyse hervorgeht, ist Ransomware eine der gefährlichsten Bedrohungen. Im Zuge der zunehmenden Angriffe auf Supplychains kann erwartet werden, dass vor allem KMUs betroffen sein werden. Ein Tool zur besseren Beurteilung einer Angriffssituation ist aus mehreren Gründen sehr vorteilhaft:

- Es zeigt, dass durch eine detaillierte Beschreibung ein Mensch mit guten IT-Kenntnissen, ohne fundierte IT-Forenische Ausbildung, dazu in der Lage ist, selbstständig eine Analyse durchzuführen, ein teurer IT-Forensiker ist somit nicht zwingend notwendig.
- Wird der Beschreibung im Tool gefolgt, ist eine Sicherung der Beweise ohne IT-Forensiker möglich. Bei Bedarf kann zu einem späteren Zeitpunkt ein IT-Forensiker hinzugezogen werden, um die Beweise für die Verwendung vor Gericht aufzubereiten.
- Den Unternehmen wird eine Möglichkeit gegeben, den Ransomware-Typ zu identifizieren und gegebenenfalls die Daten mit einem Decryptor selbstständig wiederherzustellen.
- Bei der selbstständigen Analyse können Schwachstellen im System gefunden werden, die im Anschluss behoben werden können, um weiteren Angriffen vorzubeugen.
- Es zeigt, dass eine Analyse auch mit kostenloser Open-Source-Software möglich ist und keine teure kommerzielle Software gekauft werden muss.
- Es gibt Menschen die Möglichkeit, etwas über Digitale Forensik und den Analyseprozess bei einem Angriff zu erlernen.

Das in der Masterarbeit entwickelte Vorgehensmodell und Implementierung eines Prototyps haben bestätigt, dass es möglich ist, auch für KMUs ein Lagebild zur Verfügung zu stellen, das mit vertretbarem Aufwand beherrschbar ist. Diese Masterarbeit kann somit als Ideengeber für die Abwehr von Ransomware-Angriffen auf Supplychains und KMUs gesehen werden.

## Literatur

- [1] Anzeigte fälle von cybercrime (gesamt) in Österreich von 2004 bis 2020. <https://de.statista.com/statistik/daten/studie/294141/umfrage/cybercrime-in-oesterreich/#professional>. Accessed: 23.03.2022.
- [2] It forensik - wenn der angriff passiert ist - integration ins cyber krisenmanagement. <https://www.de.information-security-informationssicherheit.com/home/cyber-defence/it-forensik>. Accessed: 29.03.2022.
- [3] National software reference library (nsrl). <https://www.nist.gov/itl/ssd/software-quality-group/national-software-reference-library-nsrl>. Accessed: 06.05.2022.
- [4] Nextjs api-routes. <https://nextjs.org/docs/api-routes/introduction>. Accessed: 19.10.2022.
- [5] Next.js docs. <https://nextjs.org/docs/basic-features/pages>. Accessed: 19.10.2022.
- [6] Nextjs fast refresh. <https://nextjs.org/docs/basic-features/fast-refresh>. Accessed: 19.10.2022.
- [7] Nextjs image optimization. <https://nextjs.org/docs/basic-features/image-optimization>. Accessed: 19.10.2022.
- [8] Open source digital forensics. <https://www.sleuthkit.org/>. Accessed: 29.04.2022.
- [9] Was ist mongodb? <https://www.mongodb.com/de-de/what-is-mongodb>. Accessed: 22.10.2022.
- [10] Was ist php? <https://www.php.net/manual/de/intro-what-is.php>. Accessed: 06.12.2022.
- [11] What is digital forensics and why is it important? <https://www.provendatarecovery.com/blog/what-is-digital-forensics/>. Accessed: 23.03.2022.
- [12] What is digital forensics and why is it important? <https://www.provendatarecovery.com/blog/preserve-ransomware-evidence/>. Accessed: 29.03.2022.
- [13] What is prisma? <https://www.prisma.io/docs/concepts/overview/what-is-prisma>. Accessed: 19.10.2022.
- [14] Why prisma? <https://www.prisma.io/docs/concepts/overview/why-prisma>. Accessed: 19.10.2022.

- [15] BAIER, P. D. H. Einführung in die digitale forensik. Vorlesungsskript, Universität der Bundeswehr München, 2021. [https://www.itsec.techfak.fau.de/files/2019/07/Einfuehrung\\_Digitale\\_Forensik\\_Leseprobe.pdf](https://www.itsec.techfak.fau.de/files/2019/07/Einfuehrung_Digitale_Forensik_Leseprobe.pdf).
- [16] BÖHM, P. D. C. Komplexität von algorithmen. Vorlesungsskript, Technische Universität München, 2007. <https://www.dbs.ifi.lmu.de/Lehre/NFInfoSW/WS0708/Skript/Folien09.pdf>.
- [17] CHINOSI, M., AND TROMBETTA, A. Bpmn: An introduction to the standard. *Computer Standards & Interfaces* 34, 1 (2012), 124–134.
- [18] COSIC, J., AND BACA, M. Do we have full control over integrity in digital evidence life cycle? In *Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces* (2010), IEEE, pp. 429–434.
- [19] COSIC, J., AND COSIC, Z. Chain of custody and life cycle of digital evidence. *Computer technology and application* 3, 2 (2012).
- [20] DAVIES, S. R., MACFARLANE, R., AND BUCHANAN, W. J. Evaluation of live forensic techniques in ransomware attack mitigation. *Forensic Science International: Digital Investigation* 33 (2020), 300979.
- [21] DOLLE, W. Computer-forensik in der praxis. *Datenschutz und Datensicherheit-DuD* 33, 3 (2009), 183–188.
- [22] FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK, B. Leitfaden „it-forensik“. *Datenschutz und Datensicherheit-DuD* (2011), 22–23.
- [23] GREENFIELD, R. S., ET AL. *Cyber forensics: a field manual for collecting, examining, and preserving evidence of computer crimes*. Auerbach Publications, 2002.
- [24] HEVNER, A. R., MARCH, S. T., PARK, J., AND RAM, S. Design science in information systems research. *MIS quarterly* (2004), 75–105.
- [25] HOQUE, N., BHATTACHARYYA, D. K., AND KALITA, J. K. Botnet in ddos attacks: Trends and challenges. *IEEE Communications Surveys Tutorials* 17, 4 (2015), 2242–2270.
- [26] KNOLL, M. Rezension „mongodb“. *HMD Praxis der Wirtschaftsinformatik* 53, 4 (2016), 555–556.
- [27] KÖHN, M., OLIVIER, M. S., AND ELOFF, J. H. Framework for a digital forensic investigation. In *ISSA* (2006), Citeseer, pp. 1–7.
- [28] LEVINSON, E. The mime multipart/related content-type. Tech. rep., 1998.

- [29] LI, M., HUANG, W., WANG, Y., FAN, W., AND LI, J. The study of apt attack stage model. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)* (2016), IEEE, pp. 1–5.
- [30] MEIER, S. *Digitale forensik in unternehmen*. PhD thesis, 2017.
- [31] MOHURLE, S., AND PATIL, M. A brief study of wannacry threat: Ransomware attack 2017. *International Journal of Advanced Research in Computer Science* 8, 5 (2017), 1938–1940.
- [32] POLLITT, M., AND WHITLEDGE, A. Exploring big haystacks. In *IFIP International Conference on Digital Forensics* (2006), Springer, pp. 67–76.
- [33] SAGLIETTI, F., OSTER, N., AND PINTE, F. White and grey-box verification and validation approaches for safety-and security-critical software systems. *Information security technical report* 13, 1 (2008), 10–16.
- [34] ZMARANDA, D., POP-FELE, L.-L., GYORÖDI, C., GYORÖDI, R., AND PECHERLE, G. Performance comparison of crud methods using net object relational mappers: A case study. *International Journal of Advanced Computer Science and Applications* 11, 1 (2020).

## A Anhang

### A.1 User Information

Ein Benutzerhandbuch ist nicht notwendig, da die entwickelte Software selbst ein Handbuch für Autopsy ist. Aufrufen lässt sich die Software über den QR-Code in Grafik 15 oder den Link <https://ransomware-analyzer.vercel.app>.

### A.2 Installation Guide

In diesem Kapitel wird erklärt wie sich das entwickelte Artefakt lokal ausführen lässt. Es ist jedoch auch über Vercel unter folgendem Link <https://ransomware-analyzer.vercel.app> oder durch den QR-Code in Grafik 15 aufrufbar.



Abbildung 15: QR-Code Vercel

Zunächst muss das Repository von github heruntergeladen werden. Es findet sich unter folgendem link:

<https://github.com/christjunior96/Ransomware-Analyzer>  
Auch der QR-Code in Grafik 16 führt zu dem Repository.



Abbildung 16: QR-Code GitHub

Ist das Repository geladen und entpackt, muss Node.js installiert werden. Ein Link zum Download für das jeweilige Betriebssystem findet sich hier:

<https://nodejs.org/en/download/>  
oder durch den QR-Code in Grafik 17.



Abbildung 17: QR-Code Node.js

Nach der Installation muss im Terminal in den entpackten Projektordner navigiert werden. Anschließend kann der Next.js "development Server" durch den folgenden Befehl ausgeführt werden.

```
1 npm run dev
```

Im Normalfall kann man auf das Programm oder die Website durch *http://localhost:3000* zugreifen.

### **Abstract**

The ability of many companies to work today is based on their IT systems. These can be shut down by a ransomware attack, leaving the company unable to work, and important data can be lost. For personnel and financial reasons, a cyberforensic investigation by a trained expert is often not feasible for small and medium-sized enterprises. In this work, the aim was to find out whether it is possible to secure and analyse a system affected by ransomware without the need for appropriate training or expensive analysis software. It is possible to carry out the analysis with the help of a guide developed as part of this work and the tool Autopsy - Digital Forensics, provided that good IT skills are available. The investigator is guided from the backup of the data to the evaluation of the traces and works out an answer to the question of which ransomware it was, when the attack took place and how the attacker got into the system. This information can be helpful in restoring the data and closing the security gap.

### **Zusammenfassung**

Die Arbeitsfähigkeit vieler Unternehmen basiert heutzutage auf deren IT-Systemen. Diese können durch einen Ransomware-Angriff stillgelegt werden, womit die Unternehmen arbeitsunfähig sind, weiterhin können wichtige Daten verloren gehen. Aus personellen und finanziellen Gründen ist eine Cyberforensische-Untersuchung von einem ausgebildeten Experten für kleine und mittlere Unternehmen oft nicht umsetzbar. In dieser Arbeit galt es herauszufinden, ob es möglich ist, ein von Ransomware betroffenes System zu sichern und zu analysieren, ohne eine entsprechende Ausbildung oder teure Analysesoftware. Es ist möglich, mit Hilfe eines im Rahmen dieser Arbeit entwickelten Leitfadens und dem Tool Autopsy - Digital Forensics die Analyse durchzuführen, vorausgesetzt es sind gute IT-Kenntnisse vorhanden. Der Untersuchende wird dabei von der Sicherung der Daten bis zur Auswertung der Spuren geleitet. Er erarbeitet sich eine Antwort auf die Frage, welche Ransomware es war, wann der Angriff stattgefunden hat und wie der Angreifer in das System gekommen ist. Diese Informationen können bei der Wiederherstellung der Daten und dem Schließen der Sicherheitslücke hilfreich sein.