HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

BACHELORARBEIT

Frau
Cécile Hofmann

Prozessautomatisierung im Rahmen eines Incident Response unter der Nutzung des Tools KAPE

Hamburg, 2022

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Prozessautomatisierung im Rahmen eines Incident Response unter der Nutzung des Tools KAPE

Autorin:
Frau Cécile Hofmann

Studiengang:
Allgemeine und digitale Forensik

Seminargruppe:
FO19w4-B

Erstprüfer:
Prof. Ronny Bodach

Zweitprüfer:
M. Eng. Svenja Mischur

Hamburg, September 2022

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELOR THESIS

Process automation in context of an incident response using the tool KAPE

Author:
Frau Cécile Hofmann

Study Programme:
General and Digital Forensic Science

Seminar Group:
FO19w4-B

First Referee:
Prof. Ronny Bodach

Second Referee:
M. Eng. Svenja Mischur

Hamburg, September 2022

Bibliografische Beschreibung:

Hofmann, Cécile:
Prozessautomatisierung im Rahmen eines Incident Response unter der Nutzung des Tools KAPE. 2022. 93 Seiten, 35 Abbildungen. Hochschule Mittweida, University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften

Bachelorarbeit, 2022

Referat

Diese Bachelorarbeit befasst sich mit der Prozessautomatisierung während einer Vorfallsreaktion (Incident Response) in der digitalen Forensik. Die Idee dafür kam während der Tätigkeit bei der intersoft consulting services AG auf.

Das Ziel der vorliegenden Arbeit ist es, zu beantworten ob die Automatisierung eines Incident Response-Prozess mit dem Tool KAPE schneller und effektiver gestaltet werden kann, ohne die forensischen Prinzipien außer Acht zu lassen. Dafür wurden eigene Konfigurationsdateien erstellt, welche auf die interne Arbeitsumgebung angepasst sind und anschließend geprüft, ob das Tool die Anforderungen hinsichtlich der forensischen Prinzipien erfüllt. Weiterhin wurde die Verwendung des Tools hinsichtlich seiner Geschwindigkeit mit dem bisherigen Vorgehen verglichen.

Die Untersuchung zeigte, dass das individualisierte Tool mit den eigens erstellten Konfigurationen eine enorme Zeitreduktion gegenüber dem bisherigen Vorgehen erreichen konnte und dies auch unter der Einhaltung der forensischen Prinzipien möglich ist.

Zusammenfassend lässt sich sagen, dass das Tool KAPE für die Prozessautomatisierung eines Incident Response eine merkliche Rolle spielen kann, insbesondere wenn es auf die interne Unternehmensumgebung angepasst ist und einer ständigen Weiterentwicklung folgt.

Inhalt

Abbildungsverzeichnis

1 Einleitung

Durch das Wachstum der Technologie sieht die Welt einen erheblichen Anstieg im Missbrauch ebendieser. [1]. Laut dem „National Institute of Standards and Technologies“ (NIST) sind stattfindende Cybersecurity-Attacken häufiger und auch schädlicher geworden, dabei können aber weitaus nicht alle Vorfälle verhindert werden [2]. Einige Unternehmen versuchen, das Risiko einfach zu ignorieren oder rechnen einen möglichen Sicherheitsvorfall sogar fest im Jahresbudget mit ein. Dementsprechend benötigt das Feld der Digitalen Forensik und Incident Response (Vorfallsreaktion) Prozesse und Methoden, um Vorfälle möglichst effizient und schnell zu bearbeiten [3]. Doch dies alleine reicht nicht, denn es ist auch wichtig, den Angreifern immer einen Schritt voraus zu sein. Diese haben im Laufe der Zeit einige Methoden entwickelt, um sich der Entdeckung zu entziehen. So kommt es oft zu einer Art zeitlichem Wettrennen zwischen den Gegnern und den Teams, welche einen Angriff verteidigen und abwehren möchten: Wird ein Exploit zuerst ausgenutzt oder kann vorher ein entsprechendes, entgegenwirkendes Update eingespielt werden [4]? Genau wie die Methoden und Werkzeuge der Angreifer sich verändert haben, muss sich deshalb auch der Bereich des Incident Response weiterentwickeln [3].

Schneier [5] bezeichnet IT-Sicherheit in seinem Artikel „The Future of Incident Response“ als Kombination von Schutz, Detektion und Reaktion. 1990 war demnach die Ära des Schutzes. In den Jahren um 2000 hat man realisiert, dass Detektion auch formalisiert werden muss und das heutige Jahrzehnt sei das der Reaktion. Er erläutert folgende Gründe, warum immer mehr Organisationen den Bereich des Incident Response als nützlich sehen: Es gibt einen Kontrollverlust über die weiten der Computerumgebung, die Attacken werden immer anspruchsvoller und Unternehmen investieren nicht genug in Schutz und Detektion. Umso wichtiger ist es deshalb, dass Forensik- und Incident Response-Teams entsprechend schnell, effizient und zuverlässig auf einen Vorfall reagieren können, um Schäden einzugrenzen und schlimmeres zu verhindern. [5]

1.1 Motivation

Durch die praktische Arbeit bei der intersoft consulting services AG und der Arbeit mit verschiedenen Tools und Artefakten (Beweismittelquellen) kam nach und nach die Frage auf, wie man den momentanen Prozess der Vorfallsreaktion (Incident Response) effektiver gestalten kann. Verantwortliche Incident Response Teams müssen Untersuchungen heutzutage schneller als je zuvor durchführen sowie eine weitaus größere Anzahl an Systemen untersuchen. Dies verlangt nach einer zunehmenden Skalierbarkeit und Automatisierung der Prozesse und Methoden [3]. Angesichts der Arbeitsbelastung eines durchschnittlichen Sicherheitsanalysten ist die Automatisierung ein

wichtiger Bestandteil der Erkennung von Vorfällen [6]. Laut einer Umfrage des SANS-Institut haben mehr als ein Viertel (26%) der befragten Fachkräfte im Bereich des Incident Response das Gefühl, die Fähigkeiten ihrer Organisationen sind ineffektiv. So bezeichnen sie sowohl die fehlende Zeit für die Überprüfung der Prozeduren und das mangelnde Training dieser (62%), als auch ein fehlendes Budget für Tools und Technologien (60%) als hauptsächliche Barriere zu einem effektiven Incident Response [6].

Dementsprechend soll das Hauptaugenmerk dieser Arbeit auf der Reduzierung des Aufwands bei einer IT-forensischen Analyse während eines Incident Response liegen. Der Fokus liegt dabei darin, einen zeitlichen Vorteil zu gewinnen, ohne die forensischen Prinzipien außer Acht zu lassen, denn der Faktor der Reaktionszeit spielt bei der Reaktion auf einen Sicherheitsvorfall eine große Rolle [7]. In der vorangegangenen Praxisarbeit wurde ein dafür nutzbares Tool Namens KAPE (Kroll Artifact Parser and Extractor) evaluiert und es wurde geprüft ob ein Einsatz in der internen Arbeitsumgebung zielführend erscheint. Dabei ist der Autor zum Entschluss gekommen, dass das Tool einen wesentlichen Beitrag in der Weiterentwicklung des Prozesses bei einem Incident Response leisten kann. Somit hat sich die Leitfrage herauskristallisiert, ob die Automatisierung eines Incident Response-Prozess mit dem Tool KAPE schneller und effektiver gestaltet werden kann, ohne die forensischen Prinzipien außer acht zu lassen. Die Reaktionszeit bei einem Sicherheitsvorfall in den nächsten zwei Jahren zu verbessern, ist auch das Ziel von 42% der Teilnehmenden der SANS-Umfrage [6].

Eine zeitliche Reduzierung, vor allem bei der Datenverarbeitung, sorgt nicht nur für eine kürzere Gesamtdauer der forensischen Analyse und eine schnellere Reaktionszeit bei Vorfällen, sondern entsprechend auch für tiefere Kosten für den Auftraggebenden. Es bewirkt auch, dass mehr Personal in der eigentlichen Datenanalyse eingesetzt werden kann und somit auch die internen Kosten für personelle Ressourcen tiefer gehalten werden. Aus wirtschaftlicher Sicht bedeutet dies auch, den Durchsatz erhöhen zu können. Mehr Vorfälle können in derselben Zeit abgearbeitet werden.

1.2 Zielstellung und Abgrenzung

Das Ziel der folgenden Arbeit ist es, eine Basis für die Unternehmensumgebung zu bieten, um schneller und effektiver zu arbeiten. Der momentane Prozess während eines Incident Response soll unter der Nutzung des Tools KAPE angepasst, verbessert und automatisiert werden. Schneier sagt aus, dass der Prozess des Incident Response aber nicht vollständig automatisiert werden kann, da es für eine erfolgreiche Vorfallsreaktion Personen benötige, die das Vermögen zum Denken haben [5]. Dies ist ein wichtiger Indikator zur Abgrenzung, denn die Analyse der Artefakte selbst soll nicht automatisiert werden, sondern nur deren Erhebung und Verarbeitung. Dabei liegt der Fokus hauptsächlich auf dem Prozessschritt der Datengewinnung und Verarbeitung. Genau in diesem Bereich arbeitet auch KAPE und kann so seinen Teil zur Automatisierung beitragen. Natürlich wird es immer Aufgaben geben, für welche es keine praktikablen oder

kostengünstigen Alternativen zum Menschen gibt. In diesen Fällen sollen Entwickler ihre Systeme aber so entwickeln, dass sie Menschen unterstützen können [5]. Genau dies ist der Anspruch der Arbeit, denn die angestrebte Prozessautomatisierung mit Hilfe des Tools KAPE soll die Arbeit des Incident Response-Teams unterstützen und nicht deren Arbeit ablösen.

Bei der Umsetzung wird sich ausschließlich auf die IT-forensische Analyse von Windows-Betriebssystemen ab Windows 7 und deren entsprechenden Artefakte fokussiert. Artefakte bezüglich Memory-Forensik werden nicht miteinbezogen, da diese eine hohe Komplexität aufweisen und dies den Rahmen der Arbeit überschreitet. Schlussendlich soll es möglich sein, im Falle eines Vorfalls eine Konfiguration im Tool KAPE zu wählen und entsprechend automatisiert, alle relevanten Artefakte für einen bestimmten Vorfalls-Typ in menschenlesbarer Form zu erhalten. Diese können dann analysiert werden bevor ein vollständiges Datenträgerabbild erstellt werden muss. Die Reaktionszeit kann so verkürzt, und auf erste Ergebnisse schneller reagiert werden.

1.3 Aufbau der Arbeit

In der folgenden Bachelorarbeit sollen nun, nach der vorgenommenen Klärung der Zielstellung und einer Abgrenzung, unter Kapitel 2 zunächst die Grundlagen geklärt werden. Diese umfassen Prozesse, Tools und relevante Artefakte, sowie das bisherige Vorgehen, welches weiterentwickelt und automatisiert werden soll. Im Kapitel 3 soll die praktische Umsetzung durch das Tool KAPE erläutert werden. Dazu werden entsprechend drei beispielhafte Vorfalls-Szenarien und deren entsprechende Artefakte gewählt. Anschließend findet eine Evaluierung durch ein bereitgestelltes Test-Image in zwei Schritten statt und es besteht die Möglichkeit einer Optimierung der bisher erreichten Ergebnisse durch erneute Verbesserungen und Veränderung des Tools. Die notwendigen Schritte werden zusätzlich insofern verallgemeinert, dass ein allgemeiner Nutzen aus dieser Arbeit gezogen werden kann. Nach der Evaluierung sollen abschließend sowohl die Ergebnisse unter zeitlichen sowie forensischen Aspekten als auch gewonnene Kenntnisse hervorgehoben und die Leitfrage beantwortet werden. Durch einen Ausblick und eine abschließende Zusammenfassung wird aufgezeigt, was erreicht wurde, welches Potenzial in der Methodik durch die stattgefundene Prozessautomatisierung ausgeschöpft werden konnte und an welchen Punkten die Methodik noch verbessert oder weiterentwickelt werden kann.

2 Grundlagen

In diesem Kapitel werden die Grundlagen für die folgenden Kapitel dargelegt. Zu Beginn wird eine Einführung in die digitale Forensik und die forensischen Prinzipien gegeben, um dann weiter auf den Bereich des Incident Response und dessen Prozess zu führen. Dabei wird besonderen Wert darauf gelegt, wo sich die angestrebte Automatisierung durch das Tool KAPE einordnen lässt. Anschließend wird der aktuelle Arbeitsprozess erläutert, um im Nachhinein ein Fazit darüber ziehen zu können, wie sich dieser unter der Nutzung vom verwendeten Tool verändert hat. Zusätzlich sollen auch wichtige verwendete Artefakte und Tools erläutert werden, damit eine Basis für das methodische Vorgehen gegeben ist.

2.1 Digitale Forensik

Der Begriff Forensik leitet sich aus dem lateinischen Wort „forum“ ab, welches für Marktplatz steht. Dies hat den historischen Hintergrund, dass früher Marktplätze oft die Orte waren, an denen Gerichtsbarkeit ausgeübt wurde [8]. Die Digitale Forensik oder auch IT-Forensik, ist einer der Teilbereiche der facettenreichen Forensik. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) definiert den Begriff folgendermaßen: „IT-Forensik ist die streng methodisch vorgenommene Datenanalyse auf Datenträgern und in Computernetzwerken zur Aufklärung von Vorfällen unter Einbeziehung der Möglichkeiten der strategischen Vorbereitung insbesondere aus der Sicht des Anlagebetreibers eines IT-Systems. Dies schließt auch Techniken der Vorfallsbearbeitung (Incident Response) mit ein.“ [7]

Das Ziel der digitalen Forensik ist im Allgemeinen die Aufklärung von Straftaten durch forensische Untersuchungen. Dabei sollen verschiedene Fragen beantwortet werden können, so unter anderem die Frage „Was ist geschehen?“, „Wo, wann und wie ist es passiert?“ und ferner auch „Wer hat es getan?“ sowie „Was kann gegen eine Wiederholung getan werden?“. Diese als „W-Fragen“ bezeichnete Anhaltspunkte geben einer forensischen Untersuchung eine Struktur vor, indem zu ihrer Beantwortung eine gewisse Richtung der Untersuchung eingeschlagen werden muss. Dabei ist es besonders wichtig, sich nicht auf eine Hypothese zu versteifen, sondern prinzipiell nach Spuren zu suchen, die eine These sowohl untermauern als auch widerlegen können. [7]

2.1.1 Forensische Untersuchung

Bei einer forensischen Untersuchung muss schnell erkannt werden, welche Geräte oder Daten relevant sind. Es ist essenziell, die richtigen Daten in einer riesigen Datenmenge zu finden, als relevant zu identifizieren und sie anschließend zu sichern. Dabei befinden sich

99% der Spuren meist nur in 1% der ganzen Datenmenge [9]. Deshalb beginnt eine forensische Untersuchung schon mit der gezielten Vorbereitung, bei welcher geeignete forensische Werkzeuge ausgesucht, geprüft und bereitgestellt werden. Auch als strategische Vorbereitung bezeichnet, können in diesem Bereich mehrere Tools vorbereitet werden, um dann damit entsprechende Daten über einen Vorfall zu gewinnen [7]. Genau an diesem Punkt knüpft auch diese Arbeit und die Automatisierung durch das Tool KAPE an.

Die allgemeine Vorgehensweise einer forensischen Untersuchung beginnt meist mit einem „Symptom“, sprich einem ungewöhnlichen Verhalten eines Systems. Je „lauter“ dabei ein Vorfall ist, desto leichter ist dieser zu erkennen [4]. Nach einer Überprüfung und Validierung, ob es sich tatsächlich um einen Vorfall handelt, erfolgt die Sammlung relevanter Daten von möglicherweise betroffenen Systemen. Diese werden dann extrahiert und können reduziert werden, sofern bestimmte Daten aus der weiteren Untersuchung ausgeschlossen werden können. Um dies zu verdeutlichen, werden in der praktischen Umsetzung daher entsprechend drei verschiedene Angriffsszenarien gewählt, um die Daten schon vor der eigentlichen Sammlung so zu reduzieren, dass nur noch für den Fall relevante Artefakte gesammelt werden. Darauf folgt schließlich die Datenanalyse, in welcher Ergebnisse in einem logischen Zusammenhang zusammengeführt werden. Bei den einzelnen Schritten nicht zu vergessen, ist die Dokumentation, welche die gesamte Vorgehensweise und deren Ergebnisse protokolliert. Schlussendlich werden diese in einem oder mehreren Berichten zusammengefasst. [7]

2.1.2 IT-Forensische Prinzipien und Methoden

Im Bereich der Forensik gibt es einige Prinzipien und Methoden, welche bei einer IT-forensischen Analyse stets zu beachten sind. Die Einhaltung derer ist unerlässlich, da ansonsten die Gerichtsverwertbarkeit einer vollständigen Analyse verloren gehen kann. Dazu gehört auch ein wissenschaftliches Vorgehen, welches für einen übersichtlichen und nachvollziehbaren Ablauf der Untersuchung sorgt. [7]

Zu den wichtigsten forensischen Prinzipien gehören die Akzeptanz, die Glaubwürdigkeit, die Wiederholbarkeit, die Integrität, die Dokumentation und das Vier-Augen-Prinzip. Unter der Akzeptanz versteht man die Anwendung von Methoden, die in der Fachwelt anerkannt und beschrieben sind. Dies bedeutet aber im Umkehrschluss nicht, dass neue Verfahren und Methoden nicht angewendet werden können. Diese sollen aber über einen entsprechenden Nachweis über deren Korrektheit verfügen. Dies geht mit dem Prinzip der Glaubwürdigkeit einher. Die Wiederholbarkeit beschreibt, dass die eingesetzten Mittel und Methoden bei der Anwendung Dritter auf dem gleichen Ausgangsmaterial, dieselben Ergebnisse liefern müssen. Der höchste Stellenwert ist bei einer IT-forensischen Untersuchung der Integrität der Daten zuzuordnen. Sichergestellte Beweismittel und Spuren dürfen auf keinen Fall durch die Untersuchung verändert werden. Dies muss jederzeit belegbar sein und ist nahe verbunden mit einer gut strukturierten

Dokumentation. Dafür gibt es die sogenannte „Chain of Custody“, eine Art Beweismittelkette, welche Informationen über den Verbleib digitaler Spuren liefert. [7]

Neben der Datenintegrität kommt auch der Datenschutz als wichtiges Thema hinzu. Um diesen zu gewährleisten, sollten personenbezogene Daten, die sich nicht auf den Untersuchungsgegenstand (Scope) beziehen, bei der IT-forensischen Untersuchung von Computern ausgeschlossen werden. Laut einem Paper von Frank et al. gestaltet sich dies aber schwierig, da es kein forensisch einwandfreies Modell gibt, um private Daten im Kontext einer digitalen Untersuchung zu schützen [10]. Dennoch ist es wichtig, sich jeweils über die lokal geltenden Gesetze zu informieren, da bei Fehlern ein Verwertungsverbot der Ergebnisse die Folge sein kann [3]. In Deutschland gibt es unter der Datenschutz Grundverordnung mehrere Gesetze in Bezug auf personenbezogene Daten.

In der IT-Forensik gibt es verschiedene Methoden eine Analyse und die damit verbundene Datensicherung durchzuführen. Die zwei Methoden der Analyse sind die Post-mortem-Analyse und die Live-Analyse. Bei der Post-mortem-Analyse, welche auch „Offline-Forensik“ genannt wird, wird ein Vorfall nachträglich durch eine Untersuchung eines Datenträgerabbilds aufgeklärt. So können mehrmals verschiedene Durchsuchungen durchgeführt werden, ohne Originaldaten zu verändern. Bei einer Live-Analyse („Online-Forensik“) beginnt die Untersuchung noch während der Laufzeit des Vorfalls. Dabei wird primär ein hoher Wert auf die Gewinnung von flüchtigen Daten gelegt. Beide Arten der Untersuchung bieten Vor- und Nachteile. Für welche Art sich entschieden wird, liegt zum einen an der Art des Vorfalls und zum anderen an den Gegebenheiten und Möglichkeiten des Zugriffs auf die Systeme. Nichtsdestotrotz muss eine beweissichere Anfertigung des Datenträgerabbilds stattfinden. Eine forensische Duplikation stellt sich als sehr zeit- und ressourcenintensiv heraus, dennoch sollte, wenn möglich, immer eine angefertigt werden. So erhält der Forensiker eine 1:1 Kopie, auch physikalische Sicherung genannt. Es werden alle Inhalte sektorenweise gelesen und geschrieben und somit sind möglicherweise auch gelöschte Dateien auffindbar. Dazu wird die Festplatte ausgebaut und an einen Schreibschutzadapter angeschlossen. Leider ist dies nicht immer möglich und es muss deshalb gegebenenfalls auf eine Live-Sicherung des Systems zurückgegriffen werden, diese wird auch logische Sicherung genannt. Dabei werden die Inhalte direkt aus dem laufenden Betrieb gesichert, gelöschte Dateien sind somit nicht sichtbar. [7,11]

2.2 Incident Response (IR)

Um den Begriff des “Incident Response” (deutsch: Vorfallsreaktion), zu definieren, gilt es zunächst zu klären, was ein Incident überhaupt ist. Unter einem Incident (Ereignis) versteht man eine beobachtbare Erscheinung in einem System oder Netzwerk. Handelt es sich um ein unerwünschtes Ereignis, ist dies oft mit negativen Konsequenzen verbunden [2]. Nahe verknüpft ist auch der Begriff des Computer-Incident, welcher ein Verstoß oder

eine unmittelbare Gefahr eines Verstoßes gegen die Computersicherheitsrichtlinien oder Standartsicherheitspraktiken darstellt [2]. Charakteristisch dafür ist oft eine Schädigungsabsicht, durchgeführt von einer Person unter Nutzung einer Computerressource [3]. Hierbei reicht der Anfangsverdacht [2]. Insgesamt ist es wichtig, dass der Begriff eines „Incident“ in einem Arbeitsumfeld klar definiert ist. Das sollte der erste Schritt sein, um auffällige Ereignisse optimal kategorisieren zu können, um dann zu entscheiden, ob diese der internen Definition eines „Incident“ entsprechen [2,3]. Häufig lassen sich Vorfälle wie Datendiebstahl, Erpressung, ein unautorisierter Zugang oder das Vorhandensein von Malware sowie auch der Besitz illegaler Materialien darunter einordnen.

Die IT-forensische Untersuchung lässt sich allgemein in die Vorfallsbearbeitung, respektive Krisenreaktion eingliedern und ist Teil des Notfallmanagements [7]. Unter dem Begriff „Incident Response“ versteht man den koordinierten und strukturierten Ansatz von der Erkennung eines Vorfalls bis zum Erreichen einer Lösung [3]. Das fängt mit der Entscheidung an, ob es sich bei einem Vorfall überhaupt um einen (vorher definierten) Incident handelt. Die schnelle Erkennung ist dabei essenziell für die spätere Eindämmung des Vorfalls. Je schneller die Angriffsvektoren identifiziert werden, desto höher stehen die Chancen, den Vorfall gut abzuwehren und Störungen und Schäden zu minimieren. Dazu gehört primär die Eingrenzung eines sogenannten „Scope“ (Untersuchungsumfang), um keine wertvolle Zeit zu verlieren und sich direkt auf die relevanten Informationen, Beweise und Artefakte zu fokussieren. Zu einer Vorfallsreaktion gehören auch die Wiederherstellung der normalen Funktion und die Bildung und Sensibilisierung der Mitarbeitenden inklusive einer allgemeinen Verbesserung der internen Sicherheitsstruktur. [2][3]

Zu den Hauptzielen eines Incident Response gehört das effektive Entfernen einer Bedrohung aus der Umgebung eines Unternehmens, die Minimierung des Schadens und des Datenverlustes und die schnellstmögliche Wiederherstellung des Normalbetriebs [3]. Diese Ziele können in die Schritte der Untersuchung und der Wiederherstellung eingeteilt werden, wobei sich diese Arbeit vorrangig auf den Teil der Untersuchung konzentriert. Ein weiterer Vorteil einer Analyse ist die Möglichkeit, gewonnene Informationen zu nutzen, um sich besser auf künftige Vorfälle vorzubereiten und einen stärkeren Schutz für Systeme und Daten bereit zu stellen [2]. Oftmals stehen sich nach einem Vorfall aber unterschiedliche Interessen bei der Reaktion auf einen Sicherheitsvorfall und der IT-Forensik gegenüber. Dennoch kann die Digitale Forensik als „Bindeglied zwischen der Reaktion auf einen Vorfall als Teil einer Incident-Response Strategie und der Strafverfolgung“ bezeichnet werden [7].

Erfolgreiche Fähigkeiten im Bereich der Vorfallsreaktion benötigen eine wesentliche Planung und Ressourcen, wozu auch gehört, klare Prozesse und effektive Methoden für die Sammlung die Analyse und die Berichterstattung zu entwickeln. Es stellt sich daher als sehr relevant heraus, Entscheidungen darüber zu treffen, welche Dienste das IR-Team anbieten soll, um dementsprechende formelle Kapazitäten zu schaffen [2].

Richtlinien und einen Plan basierend auf diesen Richtlinien sind nur ein kleiner Teil der Entwicklung eines Incident Response Prozesses und stellen den grundlegenden Fahrplan für die Implementierung der angebotenen Leistungen dar. Wichtig ist dabei vor allem, dass der jeweilige Plan zu der Mission, der Größe und der Struktur des Unternehmens passt. Weiter müssen Strategien und Verfahren festgelegt werden, wie Informationen gewonnen und ausgetauscht werden. Durch das Standardisieren dieser Verfahren sollen Fehler minimiert werden, welche in stressigen Situationen, z.B. während einer Vorfallsreaktion, durchaus passieren können [2]. Nicht zu vergessen ist aber besonders die Auswahl von geeignetem Personal mit entsprechenden Fähigkeiten. Meist sind sehr viele Personen an einem IR beteiligt, dazu gehört entsprechend nicht nur das Kern-Untersuchungs-Team sondern auch Teile des IT-Supports sowie Business Manager, Gesetzesvertretende und weitere personelle Ressourcen. [2][3]

2.2.1 Allgemeiner Prozess während eines IR

Ein Prozess soll alle notwendigen Tätigkeiten beinhalten, um die Ziele eines Incident Response zu erreichen. Dieser gesamte Prozess muss gut dokumentiert sein und vom Team verstanden werden. Wenn ein Unternehmen eine angemessene Incident Response-Kapazität hat, heißt das, dass sie Maßnahmen getroffen haben, um sicher zu gehen, dass sie auf jeden Schritt des Prozesses vorbereitet sind [12]. Im Buch Incident Response & Computer Forensics von J. T. Luttgens et. al [3], wird zwischen drei Haupttätigkeiten unterschieden: Die erste Reaktion, die Untersuchung bzw. Analyse, sowie die Wiederherstellung [3]. Ein Incident Response Prozess besteht also aus mehreren Phasen [2]. In der folgenden Prozessbeschreibung wird der gesamte Prozess dargelegt, der Fokus liegt allerdings auf der Vorbereitung und der eigentlichen Untersuchung.

Der allgemeine Prozess eines Incident Response wird nach G. Johansen [11] in sechs Schritte eingeteilt:

1. Vorbereitung
2. Erkennung
3. Analyse
4. Eindämmung
5. Beseitigung und Wiederherstellung
6. Tätigkeiten nach einem Vorfall

Wichtig ist dabei zu wissen, dass kein weiterer Schritt erfolgen kann, wenn keine Erkennung eines Vorfalls stattfindet [6]. Jeder Vorfall startet mit dem ersten Bemerken eines Ereignisses oder einer Serie von Ereignissen. Kam es zu einem Alarm, muss das Vorhandensein eines Vorfalls verifiziert und anschließend der Vorfall selbst analysiert werden, um das System dann Schritt für Schritt zum Normalbetrieb zurückzuführen. In der

folgenden Abbildung 1 ist verdeutlicht, dass der Prozess des Incident-Response als Kreislauf dargestellt wird. Die Gründe dafür werden folgend geklärt. [12]

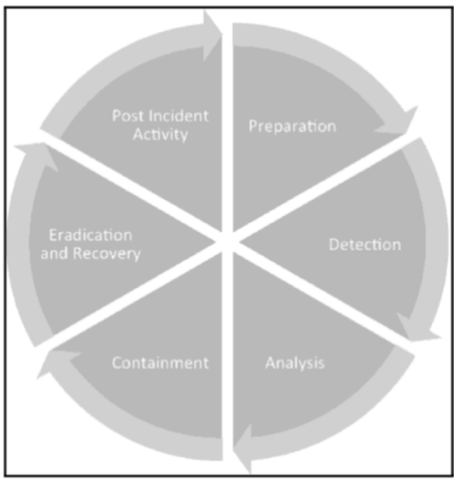


Abbildung 1 - Incident Response Prozess [12]

Vorbereitung

Ohne eine gute Vorbereitung besteht die Gefahr, dass eine Vorfallsrekation unorganisiert abläuft und der Vorfall gar verschlimmert wird [12]. Deshalb legen viele Methoden in diesem Bereich sehr viel Wert auf eine gute Vorbereitung [2,3,12]. Zu den kritischen Inhalten der Vorbereitung gehören die Erstellung eins Konstrukts, welches Prozesse, Prozeduren und Tools beinhaltet, inklusive einer entsprechenden Ausbildung und Training des Personals. Es sollten regelmäßige Übungen stattfinden, damit die Fachkräfte mit entsprechenden Prozessen vertraut sind [12].

Das Extrahieren notwendiger Daten eines Informationssystems stellt eine große Herausforderung dar, außer man bereitet sich entsprechend vor [3]. Dies gilt vor allem für die Infrastruktur, wozu auch entsprechende Tools und Ressourcen gehören. Auf Seiten der Hardware sind genügend Laptops und Ersatzrechner zwingend notwendig. Genauso wie leere Wechselmedien und Wechseldatenträger mit den gängigsten Programmen und Werkzeugen. Nicht zu vergessen ist das ganze Zubehör für die Beweiserhebung wie z.B. Beweiszettel, eine Digitalkamera, Safebags, um die Beweise sicher zu verstauen und Etiketten, um Beweismaterial eindeutig zu kennzeichnen. Softwaretechnisch ist es äußerst relevant, stets Zugang zu sauberen Betriebssystemen und Programmen für die Wiederherstellung zu haben. Vor jedem neuen Vorfall sollte der entsprechend genutzte Rechner neu aufgesetzt werden, damit Daten aus vergangenen Untersuchungen nicht in die Quere der neuen Untersuchung kommen. So spricht man in der Regel von einem „Jump-Kit“ [2], welches immer bereit für den Einsatz ist und mit allen genannten Hard- und Softwareressourcen ausgestattet ist. [2][11]

Da es in dieser Arbeit um die Prozessautomatisierung eines Incident Response mit Hilfe eines Tools geht, wird folgend entsprechend genauer darauf eingegangen, wie die Wahl und Entwicklung der Software sich in der Prozessphase der Vorbereitung widerspiegeln.

Zuerst muss immer evaluiert werden, ob die zu verwendenden Tools und Verfahren überhaupt geeignet sind [3]. Diese sollten auch immer eigenständig getestet werden, selbst wenn sie in anderen Umgebungen schon zum Einsatz kommen. Dies wurde in der vorangegangen Praxisarbeit für das Tool KAPE bereits erledigt. Das Tool wurde evaluiert und die Eignung unter Zuhilfenahme folgender Fragen aus dem Buch „Incident Response and Computer Forensics“ [3] noch einmal bestätigt:

- Ist das Tool generell akzeptiert in der forensischen Gemeinschaft?
- Deckt das Tool gängige Betriebssysteme in der eigenen Umgebung ab?
- Sammelt das Tool Daten, welche wichtig für die eigene Umgebung sind?
- Wie lange dauert eine Datenerhebung? (weniger als eine Stunde ist sehr gut)
- Ist die Sammlung der Daten selbst konfigurierbar?
- Ist die Ausgabe der Daten einfach zu überprüfen und zu verstehen?

Diese Fragen sind ein guter Richtwert bei der Auswahl und Evaluierung eines neuen Tools und können bei dem Tool KAPE allesamt mit ja beantwortet werden. Das Live-Response-Tool sollte Daten wie Systeminformationen, Informationen über Benutzerverhalten, eine Liste von Diensten und Programmen sowie geplante Aufgaben sammeln können [3].

„Wenn wir darüber nachdenken, welche Lösungen wir verwenden, geht es in der Regel um die Zeit“ – es wird generell das Tool verwendet, welches am schnellsten ist [3]. Dafür gibt es zwei ganz einfache Gründe: Einerseits findet man schneller die Bedrohung und kann Vorfälle schneller aufklären, andererseits können die Kosten gering gehalten werden, da die Bezahlung des Auftragnehmers meist per Stunde erfolgt.

Erkennung und Detektion

Die Erkennung potenzieller Vorfälle ist ein komplexes Bestreben [2]. Der schwierigste Teil dabei ist, die genaue Erkennung und Bewertung möglicher Vorfälle. Gegebenenfalls kann es sehr beschwerlich sein, diese überhaupt aufzudecken. Erschwerend kommt dazu, dass das Volumen potenzieller Anzeichen in der Regel sehr hoch ist [12]. Je nach Größe eines Unternehmens kann es bis zu 100 Millionen Events pro Tag geben [12], daher ist ein tiefes und spezialisiertes technisches Wissen sowie umfangreiche Erfahrungen in dem Bereich von großem Vorteil für eine effiziente Erkennung und Analyse [2]. Das Unternehmen kann entweder durch einen Sicherheitsbeauftragten der einen Alarm erhält oder auch aus externen Kreisen über einen möglichen Vorfall informiert werden [12]. Es ist sogar möglich, dass einem Mitarbeiter Einschränkungen in seiner Arbeitsweise (z.B. durch eine Verschlüsselung des Systems) auffallen und dahingehend auf einen Sicherheitsvorfall geschlossen werden kann.

Analyse

Wenn es sich tatsächlich um einen Vorfall handelt und dieser erkannt sowie verifiziert wurde, muss schnellstmöglich mit der ersten Reaktion und der Analyse begonnen

werden. Bei der ersten Reaktion geht es auch darum, genügend Informationen und Anhaltspunkte zu finden, damit das Team die Möglichkeit hat, eine angemessene Reaktion zu bestimmen [2,3]. Es ist wichtig, sich beginnend ein Gesamtbild über den Vorfall zu verschaffen, um dann die ersten Fakten und Informationen zu sammeln, welche das weitere Vorgehen bestimmen können. Dabei können auch Checklisten helfen, um keine wichtigen Punkte zu vergessen [3]. Unter anderem werden Personen befragt, welche den Vorfall gemeldet haben sowie IT-Fachkräfte, welche Einsicht in technische Aspekte haben. Aber auch Business-Personal wird befragt, um einen Kontext zum Vorfall zu bekommen [3]. Übliche Fragen können dabei sein:

- Was ist passiert?
- Was ist der Schaden?
- Welche Informationen wurden gestohlen?
- Welche Ressourcen sind beteiligt?
- Welche Meldepflichten gibt es?
- Läuft der Vorfall noch?
- Welche Schritte sollten getan werden, um den Vorfall zu beheben und das System für die Zukunft abzusichern?

Sind diese Fragen geklärt, beginnt die Analyse mit der Entscheidung welche Beweise auf welchen Systemen gesammelt werden sollen [12]. Je nach Fall und Methode kann die Sicherung von Daten wenige Stunden bis zu mehreren Tagen andauern und je nach Situation müssen entsprechend die Prioritäten angepasst werden [3,12]. Genau dort kann diese Bachelorarbeit eingeordnet werden, denn an diesem Punkt soll für die Prozessautomatisierung entschieden werden, welche Beweismittelquellen relevant sind und schnell erste Antworten liefern können. Es sollen keine Daten gesammelt werden, die nicht effektiv genutzt und untersucht werden, um Zeit einzusparen. Dabei ist nicht zu vergessen, dass schon allein Systemeinstellungen des jeweiligen Betriebssystems beeinflussen, welche Beweise überhaupt vorhanden sind und somit, welche Fragen überhaupt beantwortet werden können [3].

Eine Live-Datenerhebung, wie sie auch bei KAPE der Fall ist, wird unter anderem eingesetzt, damit mit der Beantwortung von Ermittlungsfragen begonnen werden kann, ohne ein ganzes Datenträgerabbild zu erstellen. Der alte Ansatz „image everything“ (alles kopieren), ist im Incident-Response Bereich manchmal zwar geeignet, aber nur selten effektiv [3]. Schnell erste Resultate bei einem Vorfall zu erlangen, kann auch dabei helfen, zu entscheiden, ob ein vollständiges Datenträgerabbild überhaupt notwendig ist. Aber auch bei einer Live-Sicherung ist es wichtig, Änderung am System während der Erhebung von Daten möglichst zu minimieren. Veränderungen sind zwar nicht vollständig zu vermeiden, aber das wichtigste Konzept ist es, diese zu minimieren und genau zu dokumentieren, was getan wurde. Trotzdem muss bedacht werden, dass eine Live-Analyse ein Risiko darstellen kann, weshalb gewissenhaft zwischen den Vor- und Nachteilen abgewogen werden muss. Zu den Vorteilen zählen dabei die Rettung volatiler Daten, schnellere Resultate und minimale Veränderungen des Systems. Die Nachteile

sind durch die Möglichkeit von Performance-Problemen bis hin zu einem abstürzenden System und dem damit einhergehenden Verlust von Daten geprägt. [3]

Bei dem darauffolgenden Prozessabschnitt der Untersuchung werden Fakten und Beweise entsprechend dem Untersuchungsumfang gesammelt und untersucht. In diesem Zusammenhang gelten viele traditionelle Ermittlungsgrundsätze weiterhin, denn sie sind generell effektiv bei der Aufklärung von Verbrechen [3]. Zu den obersten Zielen gehören dabei die Grundursache des Vorfalls zu ermitteln und die Aktionen des Angreifers von der Kompromittierung bis zur ersten Detektion zu rekonstruieren [12]. Hierzu können Leitfragen wie „Was ist passiert“ und „Wer ist dafür verantwortlich?“ behilflich sein. Diese erste Analyse soll genug Informationen bieten, um nachfolgende Aktivitäten zu bestimmen und zu priorisieren. Mithilfe von Tools kann dann genauer aufgedeckt werden, was passiert ist, welche Systeme betroffen sind und ob vertrauliche Daten abhandengekommen sind [12]. Das richtige Tool kann es einem Team erlauben, mehr mit weniger zu erreichen und in allen Etappen eines Incident Response Zeit zu sparen. Die Erhebung und Verarbeitung der Beweise soll genau in diesem Schritt so weit automatisiert werden, dass die verarbeitete Ausgabe durch KAPE direkt in einem menschenlesbaren und verständlichen Format zur Analyse für eine Fachkraft bereitstehen.

Die ganze Untersuchung stellt sich als einfacher dar, wenn ein Profil über Netzwerke und Systeme existiert, um den Normalbetrieb zu erkennen und entsprechende Veränderungen und Anomalien schneller zu identifizieren [2]. Dazu gehört auch ein Verständnis über das normale Verhalten von Programmen, Netzwerken und Systemen. Ein weiteres Thema ist die Unabhängigkeit von Beweismitteln. Schlussfolgerungen sind verlässlicher, wenn sie aus mehreren unabhängigen Beweismitteln kommen, weshalb mehrere verschiedene Beweismittelquellen gesammelt werden [3]. So können verschiedene Artefakte und Informationen gegengeprüft und verknüpft werden.

Eindämmung

Eines der wichtigsten Ziele bei der Vorfallsreaktion ist, den Verlust der Daten und die Auswirkungen auf die betroffene Organisation auf einem akzeptablen Level zu halten [6,12]. Deshalb sollten, sobald ein Verständnis über den Vorfall vorhanden ist und bekannt ist, welche Systeme involviert sind, Maßnahmen ergriffen werden, um die Angreifer in ihrem Handeln einzuschränken oder bestenfalls zu stoppen. Dazu können beispielsweise IP-Adressen blockiert oder Netzwerkverbindungen getrennt werden. Zwar benötigt jeder eine eigene Eindämmungsstrategie, dennoch ist es immer sinnvoll mehrere Optionen zu haben [12].

Beseitigung und Wiederherstellung

Im Prozessschritt der Beseitigung und Wiederherstellung geht es darum, das Problem und den Ursprung des Vorfalls zu beseitigen sowie zum Normalbetrieb zurückkehren.

Dazu gibt es einige Schritte, die zu beachten sind. Während dieser Phase wird unter anderem der Angreifende vom betroffenen Netzwerk entfernt oder bei Malwarebefall eine Anti-Malware Lösung eingesetzt. Es kann durchaus passieren, dass kompromittierte Benutzerkonten entfernt oder deren Anmeldeinformationen geändert werden müssen. Zum Teil müssen ganze Systeme neu aufgesetzt werden, frische Betriebssysteme und Programme installiert und lokale Daten aus Backups wiederhergestellt werden. Bei bekannten Schwachstellen sollte das System gepatcht und aktualisiert werden. [12]

Tätigkeiten nach dem Vorfall

Am Ende eines Vorfalls ist eine komplette Durchsicht des Vorfalls und seiner Behandlung mit allen Beteiligten vorgesehen. Dabei werden alle getroffenen Maßnahmen geprüft und es wird evaluiert, was gut funktioniert hat und was verbesserungswürdig ist. Diese Begutachtung ist dahingehend sehr wichtig, dass sie spezifische Aufgaben und Aktionen hervorhebt, welche entweder eine positive oder negative Auswirkung auf das Ergebnis und den Ausgang des Vorfalls haben. Dies ist auch die Phase, in der ein geschriebener Bericht vervollständigt wird. Die Dokumentation sollte dabei so detailliert sein, dass ein roter Faden erkennbar ist und die Grundursache des Vorfalls klar daraus hervorgeht. Außerdem muss ein Bericht auch entsprechend formuliert sein, dass er für Personen ohne IT-technischen Hintergrund leicht verständlich ist. [12]

Schlussendlich soll das Unternehmen den eigenen Prozess mit Hilfe der neu gewonnenen Informationen weiter entwickeln können. Diese sogenannten „Lessons learned“ (gelernte Lektionen) können mit in die nächsten Untersuchungen einfließen und helfen, den künftigen Prozess noch effektiver zu gestalten und besser auf neue Vorfälle vorbereitet zu sein. Infolgedessen ist der Prozess eines Incident Response als Kreislauf angedacht (Abbildung 1). [12]

2.3 Artefakte

Aufgrund der effektiven grafischen Benutzeroberfläche und der allgemeinen Benutzerfreundlichkeit ist Microsoft Windows eines der beliebtesten Betriebssysteme, aber auch das am häufigsten Angegriffene [1]. Deshalb gehört es auch zu den am meisten forensisch untersuchten Betriebssystemen [13]. Windows erstellt mehrere Artefakte als Resultat der Benutzeraktivitäten auf dem Computersystem. Unter Artefakten versteht man Objekte oder Bereiche innerhalb eines Computersystems, welche wichtige Informationen über die vom Benutzer ausgeführten Systeme enthalten [13]. Das NIST (National Institute of Standards and Technology) definiert den Begriff des Artefakts als „Ein Beweisstück, z.B. ein Text oder ein Verweis auf eine Quelle, das zur Unterstützung einer Antwort auf eine Frage vorgelegt wird“[14]. Diese Artefakte können IT-Forensikern helfen, Benutzeraktivitäten festzustellen, eine Timeline zu erstellen und eine Frequenz zu erkennen. So können Aktivitäten oder Vorfälle rekonstruiert werden [13]. Die Relevanz von Windows-Betriebssystemanalysen kann auch in der internen Arbeitsumgebung

beobachtet werden, weshalb sich diese Arbeit nur auf Windows-Artefakte und deren Auswertung fokussiert. Der Ort und die Art der Informationen unterscheiden sich nämlich je nach Betriebssystem. Es kommt dabei auch darauf an, wie ein System konfiguriert ist, je nachdem werden so einige Artefakte gar nicht erst aufgezeichnet oder sind nicht vollständig. Nur weil es aber zu einem Artefakt keine Informationen gibt, darf daraus nicht der trügerische Schluss gezogen werden, dass keine Aktivität auf dem Computersystem stattgefunden hat [13]. So können auch antiforensische Maßnahmen im Spiel sein, welche die zu untersuchenden Artefakte leeren oder gar ganz löschen [11]. Bei der Wahl der Artefakte ist es generell wichtig, zu bedenken, dass der Gegenspielende nicht darauf beschränkt ist, die den Verteidigenden bekannten Techniken und Taktiken anzuwenden, weshalb eine offene Denkweise zwingend notwendig ist [15].

Die Tabelle 1 beinhaltet eine Auswahl an Artefakten und deren Pfade sowie jeweils eine Kurzbeschreibung der einzelnen Artefakte. Sie wurde erstellt, um einen Überblick über die in der Arbeit verwendeten Artefakte zu behalten und dient als Anhaltspunkt und Referenz für das weitere Vorgehen. Im Nachhinein sollen einige dieser Artefakte und deren Speicherorte genauer erläutert werden und werden dazu in drei wesentliche Kategorien unterteilt. Das Dateisystem (Pfade beginnend mit `C:\*` zu erkennen), die Registry, welche sich in verschiedenen Hives befinden (Pfade mit Key-Bezeichnungen am Anfang, wie `HKCU`, `HKLM`) und die Eventlogs, welche sich unter `C:\Windows\System32\winevt\Logs\*`) befinden.

2.3.1 Dateisystem

Zu den Artefakten im Speicherort Dateisystem werden Artefakte gezählt, auf welche direkt im Dateisystem zugegriffen wird. Sie liegen im jeweiligen Laufwerkverzeichnis unter verschiedenen Unterverzeichnissen (siehe Tabelle 1) und müssen geparst oder mit bestimmten Programmen geöffnet werden, um sie lesen und analysieren zu können. Zu der Kategorie Systemanalyse gehört so beispielweise die $MFT, dies ist eine Datenbank im Dateisystem NTFS, welche Informationen über alle Dateien und Verzeichnisse verwaltet, welche im Volume vorhanden sind [16]. Sie liefert unter anderem Informationen über die Dateigröße, Pfade und kann auch gelöschte Dateien beinhalten, sofern diese noch nicht überschrieben sind [16]. Des Weiteren können aus dem Dateisystem Informationen über Dateiöffnungen und -erstellungen gewonnen werden. Als Beispiel kann das Artefakt „LNK-Files“ genannt werden, welches sowohl durch den Benutzer als auch durch das Betriebssystem für Dateien erstellt werden, die häufig verwendet werden [13,17]. Aufschluss über Programmausführungen können Prefetch-Dateien geben. Diese werden von Windows jedes Mal erstellt, wenn ein Programm aus einem spezifischen Quellpfad zum ersten Mal aufgerufen wird. Der Hintergrund dieses Artefakts ist es, denn Startup-Prozess eines Programms zu beschleunigen und die Performanz zu erhöhen, indem Code-Pages vorgeladen werden [13,17]. Es ist selbst möglich, Programme zu finden, welche bereits gelöscht oder deinstalliert wurden. Um Dateien und Programme zu identifizieren, welche von den Benutzern erstellt, aufgerufen oder favorisiert wurden,

können auch die „Jumplists“ hilfreich sein. Sie sind eines der Taskbar-Features ab Windows 7, welche es den Benutzern ermöglichen, alle zuletzt aufgerufenen Dateien basierend ihrer Dateikategorie anzuzeigen [13,17]. Auch Browserartefakte sind dem Dateisystem zuzuordnen, denn sie werden an verschiedenen Speicherorten im Dateisystem gefunden. Diese resultieren aus der Benutzeraktivität auf dem System durch die Verwendung verschiedener Browser. Dabei sind wichtige Informationen über die Internetaktivitäten der Benutzer hinterlegt, zu denen die aufgerufenen Websites, Emails und deren Inhalte, Chats und auch Uploads und Downloads gehören[17]. E-Mail-Artefakte werden gesammelt um Kommunikation nachvollzuziehen, Datentransfer zu ermitteln oder um bei einer Malware-Analyse danach zu suchen, ob es schadhafte Emails gibt, welche die Infektion ausgelöst haben.

2.3.2 Registry

Die Registry ist ein wichtiger Ort im Windows-System und wird als „Schatztruhe“ [13] und als „Herz und Seele des Windows Betriebssystems“ [1] bezeichnet. Sie enthält Spuren von Benutzeraktivitäten und anderen Konfigurationsdaten, die für forensische Ermittler bei der Sammlung potenzieller Beweise aus dem System sehr wertvoll sein können. Es werden Aktivitäten, die ein Benutzer auf den Systemen ausführt, bewacht und aufgezeichnet [13]. Jede Anwendung, welche auf dem Microsoft Betriebssystem läuft, tut absolut gar nichts, ohne vorher die Registry zu konsultieren. Allgemein ist die Registry eine hierarchische Datenbank, welche Daten enthält, die sowohl für den Betrieb von Windows selbst, als auch dessen Anwendungen und Dienste entscheidend sind. Die Daten sind dabei in Hives, in einer Baumstruktur angeordnet und jeder Knoten im Baum wird als Key bezeichnet. Jeder dieser Keys kann beliebig viele Subkeys oder auch Dateneinträge (Values) enthalten [19]. In einem Key kann eine beliebige Anzahl an Values in beliebiger Form vorliegen. Für die weitere Verarbeitung der Artefakte ist es wichtig, zu wissen, dass bei den Key-Namen nicht zwischen Groß- und Kleinschreibung unterschieden wird und dass jeder Subkey-Namen, in Bezug auf den Key der in der Hierarchie über ihm steht, einzigartig ist. Auch zu beachten ist, dass zwar die Key-Namen nicht in andere Sprachen übersetzt werden, deren Values hingegen schon. In folgender Abbildung 2 kann man den Registry-Hive SAM mitsamt seiner Baumstruktur und zugehörigen Keys und Subkeys erkennen. [1][19]

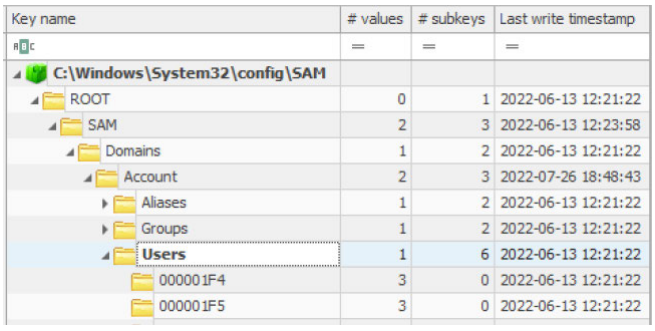


Abbildung 2 - Baumstruktur der Registry (Registry Explorer)

Die Artefakte sind in den Keys oder Subkeys und deren Values zu finden. Die eigentlichen Hives `SYSTEM`, `SOFTWARE`, `SAM` und `SECURITY` sind im Dateisystem unter dem Pfad `C:\Windows\System32` zu finden [13,20]. Der Hive `NTUSER.dat` ist unter `C:\Users\<user>\NTUSER.dat` zu finden und `USRCLASS.dat` in `C:\Users\<user>\AppData\Local\Microsoft\Windows\USRCLASS.dat`. In diesen Hives können verschiedene forensisch relevante Informationen vorhanden sein, die folgend genauer beschrieben werden [13].

Es gibt einige Registry-Keys, welche Informationen über das System, wie das installierte Betriebssystem, die Systemzeitzone, den Computernamen und sogar über die Benutzerkonten und deren Logins geben können [13]. Die Registry macht den größten Teil der forensischen Untersuchung der Dateiöffnungen und -erstellungen aus. In dem Artefakt „Shellbags“ werden nämlich Attribute der Verzeichnisse erfasst, welche mit dem Windows-Explorer durchsucht werden [13,17]. Fortführend können zuletzt verwendete Suchbegriffe (WordWheelQuery) und zuletzt geöffnete Dateien und Verzeichnisse (Shellbags, RecentDocs) nachgewiesen werden. Weitere Artefakte befinden sich in der Tabelle 1 unter Dateiöffnung und -erstellung. Malwarebezogene Spuren können in den Autostart-Keys ausgelesen werden. Dort befinden sich Informationen über geplante Aufgaben und benutzer- oder systemspezifische Autostarts, welche Programme anzeigen, die bei einem Start automatisch ausgeführt werden [20]. Auch hinsichtlich Datendiebstahl bietet die Registry einige wichtige Artefakte, vor allem zur USB-Nutzung [1,17,20]. Es können angeschlossene USB-Geräte, deren zugeordnete Laufwerkbuchstaben sowie die zu der Zeit jeweils angemeldeten Benutzer festgestellt werden [1,20]. Auch Netzwerkinformationen können aufschlussreich sein. In den Network Interfaces und der Network History können Informationen darüber gewonnen werden, welche IP-Adressen und Gateway vergeben wurden und auch welche Netzwerke mit dem Computer verbunden waren [17,20]. Bei einer zeitlichen Einordnung während einer forensischen Untersuchung kann vor allem das Artefakt „AppcompatCache“ hilfreich sein. Es lässt die Möglichkeit offen zu sehen, wann und von welchem Pfad Programme ausgeführt wurden [17,21].

2.3.3 Eventlogs

Eventlogs sind dafür zuständig, Computerbenachrichtigungen und -warnungen aufzuzeichnen, die als Ergebnis von Benutzer- oder Systemaktivitäten erzeugt werden [13,22]. Viele Anwendung zeichnen Fehler und Ereignisse in proprietären Fehlerprotokollen auf, die jeweils ein eigenes Format haben. Um eine optimale Fehlerbehandlung zu erreichen, müssen oft viele Quellen geprüft werden, denn Daten aus verschiedenen Anwendungen lassen sich nicht ohne weiteres zu einem vollständigen, sinnvollen Bericht zusammenführen. Die Eventlogs bieten aber eine standardisierte und zentrale Möglichkeit für Anwendungen und auch für das Betriebssystem, um wichtige Informationen und Hardwareereignisse aufzuzeichnen. Diese Ereignisse können durchaus eine forensische Relevanz haben und werden durch verschiedene Programme

in eine menschenlesbare Form gebracht, um die einzelnen Ereignisse (Events) dann zu analysieren. [22]

Zu den Eventlogs zählen einige Standardlogs und benutzerdefinierte Logs, welche ebenfalls unter Tabelle 1 abgebildet sind und für forensische Untersuchungen relevant sein können. Diese befinden sich unter dem Pfad `C:\Windows\System32\wievt\Logs` [13,23]. Im `Security.evtx` sind Events wie gültige und ungültige Anmeldeversuche inklusive Zeitstempel und weiterer Informationen vorhanden und im `System.evtx` befinden sich Events, welche von Systemkomponenten protokolliert werden [17,24]. Unter Custom-Logs versteht man Eventlogs, welche von Anwendungen protokolliert werden, die ein benutzerdefiniertes Eventlog erstellen [23]. Jedes Event besitzt dabei auch eine spezifische Event-ID, welche auf ein bestimmtes Ereignis hinweist [25]. In folgender Abbildung 3 ist die Struktur eines Eventlogs bei der Ansicht mit dem Event Log Explorer von FSPro Labs [26] dargestellt:

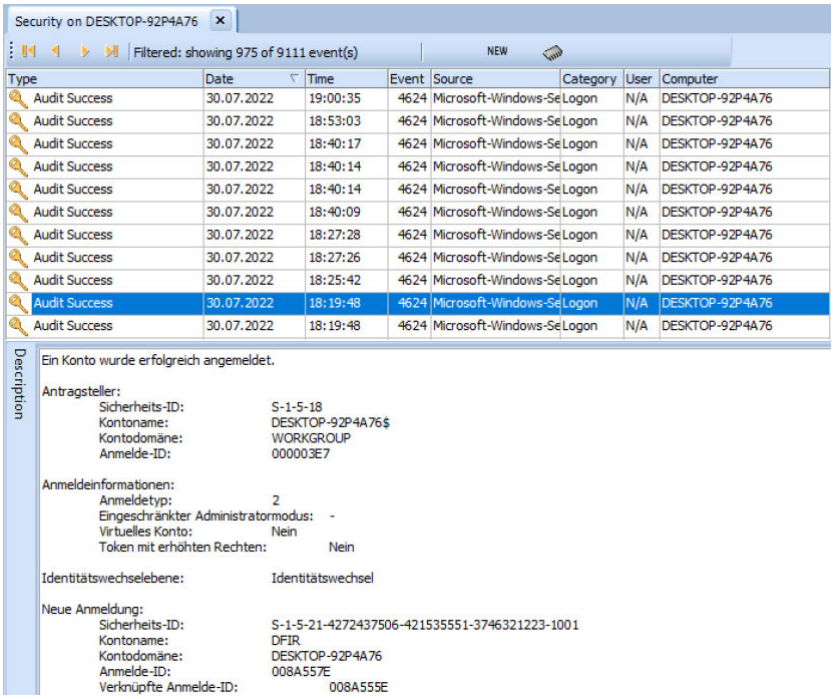


Abbildung 3 - Eventlog Beispiel

Eventlogs enthüllen dabei unter verschiedenen Event-IDs wichtige forensische Informationen über An- und Abmeldungen (Abbildung 3) sowie Programmaktivitäten und Veränderungen in den Einstellungen [13]. In Referenz auf die Tabelle 1 können so unter anderem Informationen im `Security.evtx` über Konten, deren Anmeldeversuche (Event-IDs 4624, 4625) sowie die Art der Anmeldung (Event-ID 4672) und auch neue angelegte Benutzer (Event-ID 4720) gewonnen werden [17,24]. Ein weiteres relevantes Artefakt, welches Hinweise auf Dateiöffnungen und -erstellung geben kann ist das Eventlog `OAlerts.evtx` unter der Event-ID 300 [27]. Dort können Hinweise zu verschiedenen Windows-Dialogen über Löschungsanfragen von Dateien gefunden werden. Sogar für die USB-Nutzung können im `System.evtx` Protokoll unter der Event

ID 20001 Informationen über installierte Plug&Play Treiber geben, welche auf USB-Geräte hinweisen [17]. Vor allem aber ist es möglich malwarebezogene Spuren oder Spuren eines Angriffs von außen in den Eventlogs zu finden [13]. Dazu gehören der ein- und ausgehende Remote Access und PowerShell-Remoting sowie ausgehende Map-Network-Shares und Auskünfte über den Server-Message-Block [17,28]. Weitere Informationen darüber befinden sich in der Tabelle 1.

2.4 KAPE

KAPE ist die Abkürzung für „Kroll Artifact Parser and Extractor“ und steht für ein Open-Source Tool, entwickelt von Eric Zimmerman. Das Tool dient dazu, Windows-Artefakte zu sammeln und diese anschließend so zu verarbeiten, dass sie menschenlesbar sind („Parsen“) [29]. Das Triage-Programm visiert einen Speicherort an, und findet die relevantesten Artefakte innerhalb Minuten. Dies funktioniert sowohl auf Live-Systemen als auch auf eingebundenen Datenträgerabbilder [29]. Zudem bietet KAPE viel Spielraum für eigene Funktionen oder Erweiterungen. Zwar gibt es vorinstallierte Targets und Module (unter anderem die EZ-Tools von Eric Zimmerman) für die gängigsten Vorgehensweisen. Es ist aber auch möglich, eigene auf die interne Arbeitsumgebung angepasste Konfigurationen zu erstellen. KAPE erlaubt es, sehr granular damit umzugehen, welche Informationen gesammelt werden und wie diese verarbeitet werden, wodurch das Tool sehr individuell abgestimmt werden kann. [18]. Daher wird KAPE als sehr nützlich für die Live-Forensik und die forensische Triage bezeichnet [18]. Es bietet sogar die Möglichkeit, mittels Remote Zugriff auf notwendige Zieldateien zuzugreifen, diese zu sichern und sie dann zu bearbeiten, statt ein ganzes E01-Datenträgerabbild zu verlangen. Die gesammelten Artefakte können dann auch in eine Cloud oder andere Webservices hochgeladen werden [18]. Zudem kann KAPE auch als Stickware ausgeführt werden, was es als Tool sehr praktikabel macht. Denn es bedarf keiner Installation, kann überall hin mitgenommen und jederzeit ohne großen Aufwand angewendet werden. Allerdings werden zur Verwendung von KAPE Administratorrechte benötigt. Eric Zimmerman, der Autor des Tools KAPE sagt darüber selbst, er habe erkannt, wie DFIR-Fachpersonal von einem Programm, welches forensisch wertvolle Daten schnell sammelt und verarbeitet, profitieren könne [30]. Weiter bezeichnet er KAPE als effizientes und hochgradig konfigurierbares Triage-Programm, welches im Wesentlichen auf jedes Gerät oder jeden Speicherort abzielt um forensisch nützliche Artefakte zu finden und diese innerhalb Minuten zu Parsen [30].

Funktionsweise

Das Tool kann in zwei wesentliche Hauptfunktionen unterteilt werden. Zum einen das Sammeln von Daten mittels sogenannten Targets und zum anderen das Verarbeiten dieser auf Grundlage von Modulen. Auch eine Kombination beider Verfahren ist möglich [29]. Dies ist auch in folgender Abbildung 4 sichtbar. Beim Sammeln wird jede Datei von ihrem Quellverzeichnis in ein Zielverzeichnis kopiert und die originalen Zeitstempel aller

Verzeichnisse und Dateien werden auf die Zieldateien angewendet. Des Weiteren werden auch Metadaten in Protokolldateien gesammelt [30,31]. Bei der zweiten Funktion werden eines oder mehrere Programme auf gesammelte Daten angewendet. Die Ergebnisse werden in kategorisierten Verzeichnissen gespeichert [30,32]. So muss eine Fachkraft nicht mehr zwingend wissen, wie die einzelnen Artefakte aussehen oder wo sie sich befinden, sondern kann direkt mit der menschenlesbaren Ausgabe arbeiten [30]. In folgender Abbildung 4 ist der Arbeitsablauf von KAPE dargestellt.

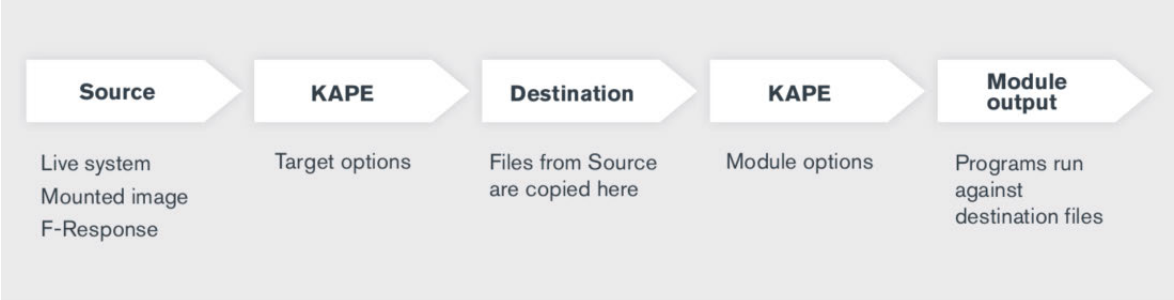


Abbildung 4 - KAPE Arbeitsablauf [30]

Als kommandozeilenbasiertes Tool erstellt man mit KAPE verschiedene Befehle zu entsprechenden Problemstellungen. In der grafischen Benutzeroberfläche besteht die Schnittstelle aus den Abschnitten Target und Module, welche sich durch Anwählen von „Use Targets/Module Options“ einschalten lassen. Beim Halten der Maus über der jeweiligen Konfigurationsmöglichkeit, wird der zugehörige Kommandozeilenbefehl angezeigt. Wählt man dann eine Konfiguration an, formt sich der jeweilige Befehl unter „Current command line“. Wurde ein Befehl erfolgreich zusammengestellt, kann man ihn entweder unter „Copy“ kopieren und später verwenden oder ihn direkt unter „Execute“ ausführen. Dann öffnet sich eine Kommandozeile, auf welcher die durchgeführten Aktivitäten mit einer Fortschrittsanzeige zusammen ausgegeben werden. [33] Damit ein Befehl funktioniert, gibt es einige erforderliche Felder, diese sind auf der grafischen Oberfläche (GUI) rot eingerahmt und sind in Abbildung 5 ersichtlich. Dazu gehören die „Target Source“, die „Target Destination“ sowie die „Module Destination“. Sofern der Reiter Targets angewählt wurde, kann die „Module Source“ weggelassen werden, da KAPE dann die Dateien automatisch aus der „Target Destination“ zieht. [18,34]

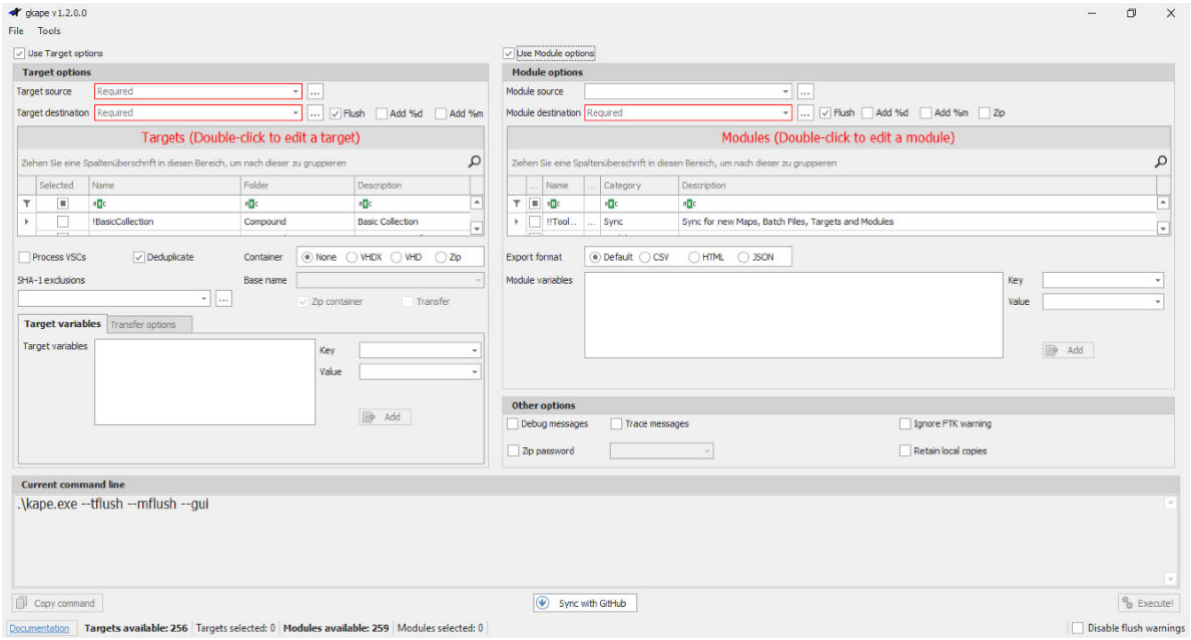


Abbildung 5 - KAPE GUI gkape.exe

Die grafische Oberfläche (GUI) kann auch als Editor für die verschiedenen Target- und Modulkonfigurationen gesehen werden, denn es kann mit einer vorhandenen Vorlage begonnen und Anpassungen vorgenommen werden, bis die neue Konfiguration den persönlichen Wünschen entspricht [35]. Allerdings funktioniert eine Speicherung oder Aktualisierung der neuen Targets oder Module nur, wenn das Skript fehlerfrei ist. Die Targets und Module sind in „YAML“ verfasst, dies ist eine vereinfachte Sprache zur Datenserialisierung, welche einfach zu lesen und zu schreiben ist [36]. Die Verarbeitung der Targets und Module erfolgt im Folgenden durch den von Windows 10 mitgelieferten Editor.

2.4.1 Targets

Targets sind eine Art Konfigurationsdateien, welche festlegen, was KAPE sammeln kann und wo sich dies befindet. Die jeweilige Datei im .tkape-Format beinhaltet Informationen darüber, dass von bestimmten Programmen oder Diensten gespeicherte Dateien an bestimmten Orten vorzufinden sind. Diese Dateien werden dann aus ihren Ordnern in einen Zielordner kopiert, allerdings werden die Zeitstempel und weitere Metadaten der Originaldatei übernommen. Zusammengefasst kann man also sagen, dass Targets verwendet werden, um Parameter zu definieren, die auf bestimmten Dateipfaden im Volume basieren. Eine beispielhafte Vorlage einer Target-Konfigurationsdatei ist unter Abbildung 6 dargestellt. Dabei ist es wichtig, die Targets sehr spezifisch auf einen bestimmten Dateityp auszurichten. Sollen mehr als eine Art von Dateikategorie auf einmal gesammelt werden, kann ein neues Target erstellt werden, welches auf andere, bereits bestehende Targets verweist. Dies nennt man dann „Compound“. Der Unterschied zu

einer gewöhnlichen Target-Datei besteht dabei darin, dass die Variable „Path“ auf den Namen einer der Zieldateien (weitere .tkape-Datei) verweist. Sobald KAPE dies erkennt, erweitert es die referenzierte Zieldatei und zieht alle zugehörigen Dateispezifikationen heran. So können beliebig viele Ebenen in den Targets erstellt werden. [31]

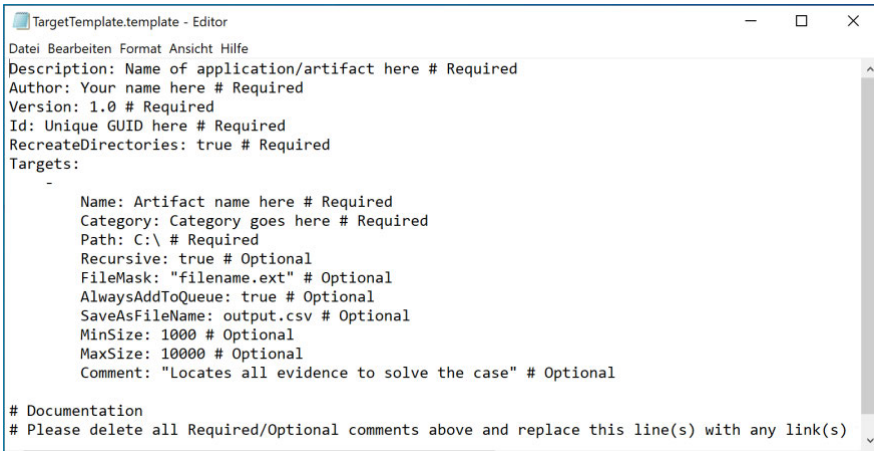


Abbildung 6 - Target Template

Da im folgenden Kapitel mit den Target-Konfigurationsdateien gearbeitet werden soll, werden die einzelnen beschreibenden Felder sowie Dateispezifikationen und Konfigurationsmöglichkeiten näher erläutert. Alle Targets haben denselben Aufbau:

Zu der allgemeinen Beschreibung gehören eine „Description“, in welcher der Zweck des Targets detailliert genug beschrieben wird, dass man weiß, welche Arten von Daten gesammelt werden. Ergänzt wird dies durch den Autor, die Versionsnummer und eine eindeutige ID, die es KAPE ermöglicht, zu erkennen, ob diese Targets schon einmal ausgeführt wurden. Dies vermeidet Dopplungen. Die Option „RecreateDirectories“ sorgt, wenn auf „True“ gesetzt, dafür, dass die Dateistruktur vom Quellgerät im Zielordner nachgeahmt wird. [31]

Diese Felder sind alle erforderlich und darauf folgt dann die eigentliche Liste der Target-Objekte, welche sich jeweils wie folgt zusammensetzt:

Mit dem „Name“ wird ein Name vergleichbar mit einem Titel vergeben, dieser wird nicht zum Auffinden der Datei verwendet. Die „Category“ beschreibt die Kategorie des Targets und unter „Path“ wird der Pfad der zu sammelnden Datei angegeben. Dabei sollte jedes Target so behandelt werden, als befände es sich auf dem Laufwerk `C:\`. Der tatsächliche Laufwerkbuchstabe wird während der Verwendung von KAPE ersetzt. Zudem kann der Pfad auch Platzhalter in Form des *-Symbols enthalten, was besonders hilfreich ist, wenn für einen bestimmten Pfad mehr als ein Verzeichnis existiert. Wichtig ist auch, dass der Pfad aus Konsistenzgründen mit einem „\“ enden sollte. [31]

Zu diesen erforderlichen Optionen kommen noch einige hilfreiche optionale Optionen. „Recursive“ sorgt, wenn auf „TRUE“ gesetzt, dafür, dass alles aus dem Pfad rekursiv

kopiert wird und mit „FileMask“ wird im Pfad nach Dateien gesucht, welche der angegeben Dateierweiterung entsprechen. Durch „SaveAsFileName“ wird der Name der Datei, welche im Zielordner gespeichert werden soll, überschrieben. Eine sehr missliche Option ist „AlwaysAddToQueue“, denn wenn diese auf „True“ gesetzt ist, werden Dateien unabhängig von der Auskunft des Betriebssystems in die Warteschlange gestellt. Mit „Comment“ ist die Möglichkeit gegeben, in einem freien Feld notwendige Informationen zu übermitteln. Sucht man nur nach Daten in einer entsprechenden Größe, kann man mit „MinSize“ und „MaxSize“ beschränken, welche Dateigrößen miteinbezogen werden sollen. [31]

2.4.2 Module

Module sind ähnlich zu Targets, denn diese Konfigurationsdateien legen fest, welche Programme KAPE ausführen kann. Sie arbeiten auch ähnlich wie Targets, indem sie nach bestimmten Dateitypen suchen. Nur kopieren sie diese nicht an einen Ort, sondern lassen Programme dagegen laufen und parsen die Daten entsprechend. Die Ausgabe-Dateien sind dann in einer menschenlesbaren Form aufbereitet und können analysiert werden. Es ist wichtig zu wissen, dass hinter jedem Modul auch eine ausführbare Datei steht, welche die gewünschte Aufgabe tatsächlich erfüllen kann. Für die Verwendung eigener Programme in den Modulen benötigt KAPE dementsprechend die passende .exe-Datei unter `.\KAPE\Modules\bin`. Das Tool benachrichtigt den Nutzenden allerdings über externe Abhängigkeiten und benötigte Binaries für Module. Diese können unter folgendem Kommando eingesehen werden: `--mlist . -mdetail`. Auch Module sind alle gleich aufgebaut (Abbildung 7) und besitzen die Erweiterung .mkape. [32]

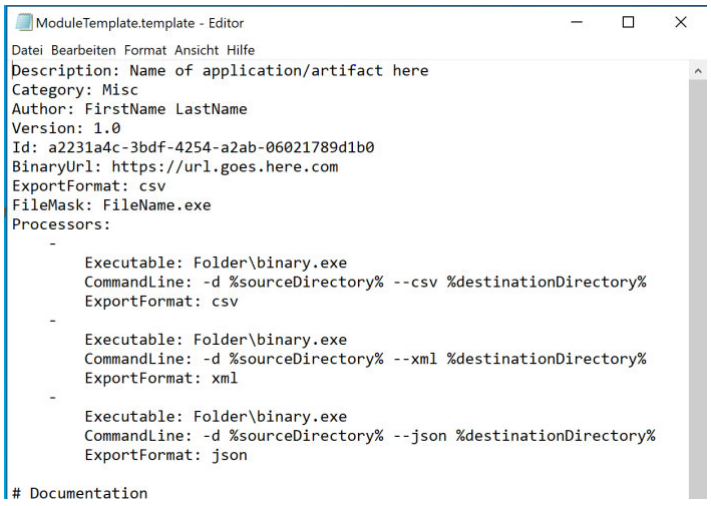


Abbildung 7 - Modul Template

Auch bei den Modulen fängt die Konfigurationsdatei mit einer „Description“ an, welche detailliert genug sein sollte, um zu verstehen, welche Art von Dateien verarbeitet werden. Die „Category“ gibt an, zu welcher Kategorie das Modul gehört und welche Art von Artefakt verarbeitet wird. Der Sinn dahinter ist es, Module zu gruppieren, welche auf

dasselbe Ziel gerichtet sind. Dazu kommen die bereits unter 2.4.2 beschriebenen Optionen wie der Autor, die Versionsnummer und eine eindeutige ID. Die „BinaryUrl“ liefert den Link zu den jeweils erforderlichen Binaries der Module. Durch „ExportFormat“ kann das Ausgabeformat definiert werden, dies wird bei dieser Arbeit CSV sein. Mit „WaitTimeout“ ist es möglich, die Zeit, die KAPE für ein Modul aufwendet, zu beschränken. Ist diese Zeit (angegeben in Minuten) abgelaufen, wartet KAPE nicht länger auf die Beendigung des Moduls und fährt mit seiner Arbeit fort. Weiterhin gibt es die optionale Option „FileMask“, mit welcher nach spezifischen Dateien per Name gesucht werden kann. Nach diesen Feldern erfolgt die Liste der eigentlichen Prozessoren, von welchen allerdings jeweils nur eines ausgewählt und verwendet wird. [32]

Diese setzen sich aus folgenden Optionen zusammen:

Zuerst wird mit „Executable“ der Name des auszuführenden Programms genannt, dabei ist nicht der gesamte Pfad zur ausführbaren .exe-Datei erforderlich. Mit „CommandLine“ werden die Parameter, die an ausführbare Dateien übergeben werden sollen, definiert. Dazu können mehrere Platzhalter-Variablen verwendet werden, welche zur Laufzeit durch KAPE ersetzt werden. So zum Beispiel %sourceDriveLetter% (für den Quell-Laufwerkbuchstaben) oder %sourceDirectory% (für das Quellverzeichnis). Die beiden Optionen „ExportFile“ und „Append“ stehen in enger Beziehung zueinander. Während erstere einen optionalen Wert darstellt, der Standard-out und Standard-Fehler an angegeben Dateinamen umleitet, entscheidet Append darüber, ob eine neue Ausgabedatei erzeugt wird oder alle Ausgaben für das Modul aneinandergehängt werden. Es ist anzumerken, dass auch Module zu einem Compound zusammengeschlossen werden können. Dies geschieht unter denselben Regeln und Strukturen wie bei den Targets. [32]

2.4.3 Logfiles

Während seiner Anwendung stellt KAPE mehrere Logfiles an unterschiedlichen Orten bereit [30,37]. Wenn das Tool startet, wird so jeweils der Zeitstempel in die Logfile-Namen mit aufgenommen, um eine bessere Zuordnung zu ermöglichen. Das ConsoleLog.txt, protokolliert alles was im Befehlsfenster zum Zeitpunkt der Ausführung von KAPE erscheint. So werden alle Aktionen die KAPE ausgeführt hat inklusive Zeitstempelgespeichert. Diese primäre Logdatei kann durch die Optionen `--debug` und `--trace` sogar noch detailliert gestaltet werden. Dies ist IT-forensisch sehr wertvoll, vor allem für die Nachvollziehbarkeit und Wiederholbarkeit. Das ConsoleLog.txt befindet sich meistens in der Target-Destination oder/und in der Modul-Destination. Nur wenn ein Fehler auftritt, findet sich die Logdatei im Verzeichnis, unter dem auch `kape.exe` gespeichert ist vor. [37]

Im Copylog und Skiplog werden detaillierte Protokolle zu kopierten und übersprungenen Dateien im CSV-Format erstellt. Die CSV-Dateien beinhalten in ihrem Namen sowohl den

Zeitstempel als auch den Wert der in der Befehlszeile unter `--target` angegeben wurde. Das _Copylog.csv enthält Details von verschiedenen Dateien die von `--tsource` kopiert wurden. Zu den Details gehören neben Datum und Uhrzeit der Protokollierung des Dateikopiervorgangs auch der vollständige Pfad zur Quell- und Zieldatei. Zusätzlich ist die Größe der Datei in Bytes vermerkt und auch der SHA-1-Wert der Quelldatei ist angegeben. Es stehen auch Informationen über verschiedene Zeitstempel wie die Zeit der Dateierstellung, Dateiänderung oder die Zeit des letzten Zugriffs zur Verfügung. Auch vermerkt wird, wie lange der Kopiervorgang gedauert hat. Das _Skiplog.csv enthält dementsprechend Einzelheiten zu den verschiedenen Dateien, die nicht von `--tsource` kopiert wurden. Dazu gehört der vollständige Pfad zur Quelldatei so wie der SHA-1-Wert dieser. Zusätzlich wird noch der Grund aufgeführt, warum die Datei übersprungen wurde, sie kann entweder ausgeschlossen (durch die `--hex` Option) worden sein oder der fehlgeschlagene Vorgang lag an einer Deduplizierung. [37]

2.5 Eric Zimmerman Tools

Unter den Eric Zimmerman Tools versteht man eine Open-Source-Toolsammlung für die digitale Forensik, welche im Laufe der Jahre durch E. Zimmerman entwickelt und kontinuierlich verbessert wurde. Diese Tools können für verschiedene Ermittlungen eingesetzt werden, unter anderem kann damit eine gegenseitige Validierung von Ergebnissen anderer Tools stattfinden. Die Sammlung kann zum Teil Einblicke in technische Details geben, die von anderen Tools nicht aufgedeckt werden. Die verschiedenen Parser decken die gängigsten Windows-Artefakte ab, welche im vorherigen Kapitel unter 2.4 näher betrachtet wurden. Bei KAPE, welches auch von E. Zimmerman entwickelt wurde, ist diese Toolsammlung schon vorinstalliert und auch in einigen Targets und Modulen schon integriert. Dadurch sollen sie auch im Prozess der Automatisierung als Parser verwendet werden und werden deshalb im Anhang unter Tabelle 2 aufgeführt und folgend erläutert [38,39]:

Die Arbeit mit den Zimmerman-Tools wird intern als „händische Analyse“ bezeichnet. Darunter wird die Analyse der einzelnen Artefakte mit den verschiedenen Parsern (Tabelle 2) verstanden [39]. Dies ist sehr aufwändig und nimmt viel Zeit in Anspruch, weil jedes einzelne Artefakt in der Verzeichnisstruktur gesucht und mit einem spezifisch darauf ausgerichteten Parser geparst wird. Für jedes Artefakt wird also einzeln ein Befehl ausgeführt. Dieser enthält die Art des Parsers, den Pfad, der zum Artefakt führt, sowie das gewünschte Ausgabeformat inklusive Zielordner. Beispielhaft ist folgend ein Kommando angegeben:

```
AmCacheParser.exe -f C:\Windows\appcompat\AmCache.hve --csv
„C:\Users\User1\Desktop\Beweise“.
```

Für diese Tools gibt es verschiedene Befehlskonfigurationen, von welchen die gängigsten folgend kurz zusammengefasst werden:

- -d um ein zu verarbeitendes Verzeichnis (Directory) anzugeben
- -f um eine zu verarbeitende Datei (File) anzugeben
- --csv um das Ausgabeformat als CSV-Datei zu deklarieren
- --csvf um die Ausgabedatei im CSV-Format zu benennen
- --mp um eine bessere Auflösung der Zeitstempel zu ermöglichen
- --debug um debug-Informationen anzuzeigen
- --trace um eine genauere Dokumentation des Vorgehens zu erhalten

[40,41,42]

Dabei sind nicht alle Optionen für jedes Tool verfügbar. Informationen zu jedem einzelnen Tool befinden sich auf der GitHub-Webseite von Eric Zimmerman [39].

2.6 Bisheriges Vorgehen / State of the Art

Das aktuelle Vorgehen bei einer IT-forensischen Untersuchung während eines Incident Response umfasst, wenn möglich, eine vollständige Sicherung des zu untersuchenden Datenträgers. Dies geschieht, wann immer möglich, durch das Ausbauen der Festplatten des betroffenen Systems und das Anschließen deren an einen Write-Blocker. Von dort aus wird eine 1:1 Kopie angefertigt. Wenn dies nicht möglich ist, wird eine Live-Sicherung des Systems vorgenommen, dabei ist es wichtig das Vier-Augen-Prinzip zu wahren und auf eine saubere, lückenlose Dokumentation zu achten. Dazu dienen die Beweismittelzettel, auf welchen genau dokumentiert wird, wo sich das Beweismittel zu welcher Zeit befand. Die Analyse findet dementsprechend auf einer Arbeitskopie statt und niemals auf einem Originaldatenträger oder der Masterkopie. Die zwei Letzteren werden in einer zutritt-gesicherten Asservatenkammer in sogenannten „Safebags“ gelagert, um sicherzustellen, dass keine Veränderungen an den Asservaten vorgenommen werden können.

Während der eigentlichen Analyse finden im Wesentlichen zwei Methoden Anwendung. Es wird mit dem vollautomatisierten Programm „Magnet Axiom“ ein automatischer Suchlauf nach verdächtigen Dateien, URLs, Ereignissen oder Benutzern gestartet. Dies dauert meist mehrere Stunden und dient bei der Untersuchung später als grober Anhaltspunkt dafür, wo man noch mehr Beweise oder wichtige Hinweise finden kann. Der eigentliche Fokus liegt auf der manuellen (händischen) Auswertung der einzelnen, für den Fall relevanten Artefakte. Zurzeit wird dafür nur die Eric Zimmerman Toolsammlung in Begleitung anderer weiterer Tools angewandt und KAPE kommt nicht zum Einsatz. Die Dauer dieser händischen Auswertung (ohne Analyse) beträgt in der Regel zwei bis drei Minuten pro Artefakt. Die so geparsten CSV-Dateien werden meist in einem spezifischen „Beweise“-Ordner für den Fall gespeichert. Von dort aus werden diese dann analysiert und auf Auffälligkeiten untersucht. Entsprechend der gefundenen Ergebnisse werden die CSV-Dateien in einen vorgefertigten Bericht unter Erklärung der vorgefundenen Ergebnisse eingefügt.

Von diesem internen „State of the Art“ aus soll KAPE eine wesentliche Verbesserung in der Geschwindigkeit der Datenverarbeitung bringen, da es die bis dato händisch getätigte Arbeit automatisiert. Wie genau KAPE dies erreicht, ist im nächsten Kapitel beschrieben.

3 Methodisches Vorgehen

In diesem Teil der Arbeit wird das praktische Vorgehen und die Umsetzung der Prozessautomatisierung mittels KAPE beschrieben. Dafür werden zunächst drei Szenarien bezüglich eines „Incident“ vorgestellt, zu welchen dann unter entsprechender Erörterung relevante Artefakte gewählt werden. Diese werden dann mittels zu erstellender Targets und Module abgebildet, um im Endeffekt drei zusammengefasste Compound-Targets und -Module entsprechend den Vorfallszenarien zu entwickeln. Diese Target- und Modul-Compounds enthalten Gruppen von anderen Targets und Modulen, was die Optionen minimiert, welche angewählt werden müssen, um die gewünschte Konfiguration zu erreichen [18]. Dabei wird zur Orientierung auf vordefinierte Targets und Module zurückgegriffen, um dann Individualisierte zu erstellen.

3.1 Wahl der Szenarien und entsprechender Artefakte

Die drei am häufigsten vorkommenden Vorfälle betreffen laut einer SANS-Umfrage aus dem Jahr 2014 Schadsoftware (82%), unautorisierte Zugriffe (70%) oder neben falschen Alarmen auf dem dritten Platz die Datenschutzverletzung [6]. Bis in die heutige Zeit kann durch das interne Arbeitsumfeld die Tendenz der Häufigkeit dieser drei Vorfälle bestätigt werden.

Deshalb werden folgend diese drei Szenarien genauer erläutert und entsprechende Artefakte, die zu der Aufklärung der jeweiligen Vorfälle beitragen, gewählt und evaluiert. Die Auswahl der Artefakte ist in der Tabelle 3 im Anlagenteil übersichtlich dargestellt. Dazu wird auch jeweils das Tool genannt, mit welchem das Artefakt in den Modulen geparst werden soll. Die dazugehörigen Pfade lassen sich für das jeweilige Artefakt aus der Tabelle 1 entnehmen. Es ist dabei ersichtlich, dass die Registry-Hives für jedes Szenario genutzt werden. Dies resultiert darin, dass sich weitere unterschiedliche Artefakte in diesen Hives befinden, welche für die jeweiligen Untersuchungspunkte relevant sein können. Zudem können diese, durch die Target Konfiguration kopierten Hives, auch im Rohformat in den Registry-Explorer von Eric Zimmerman [39] geladen werden und von dort aus weiter analysiert werden. Auch die zur Systemanalyse gehörenden Artefakte, wie die $MFT, die installierte Betriebssystem-Version, der Computername und Zeitzoneninformationen werden bei jedem der drei Szenarien gesammelt. Sie geben grundlegende Informationen über das zu untersuchende System und können für die weitere Analyse maßgebend sein. Zusätzlich wird auch in allen drei Fällen das `Security.evtx` Eventlog analysiert, da es Auskunft über An- und Abmeldungen verschiedener Benutzer sowie weitere Details über Anmeldungen bietet [17,24]. Diese Artefakte werden nachfolgen deshalb als allgemeine Artefakte bezeichnet.

Das Ziel bei der weiteren Artefakte-Wahl ist es, eine möglichst schnelle erste Analyse durchzuführen. So sollen nicht einfach alle bekannten und möglicherweise nützlichen Artefakte aus Tabelle 1 gesammelt und verarbeitet werden, sondern nur die tatsächlich für den ersten Überblick relevante Artefakte. Diese unterscheiden sich in den drei Szenarien erheblich. Wird im Nachhinein bekannt, dass weitere Artefakte für die Aufklärung des Vorfalls notwendig sind, können diese ohne Probleme aus der nachträglich erstellten 1:1 Kopie oder mittels Live-Response zusätzlich gesammelt werden. So ist die Auswahl der Artefakte auf erste mögliche Anhaltspunkte des jeweiligen Vorfalls begrenzt, um eine sehr schnelle erste Reaktion zu ermöglichen.

Werden die bereits genannten allgemeinen Artefakte, welche für jedes Szenario gesichert und analysiert werden, weggelassen, ergibt sich folgende Statistik (Abbildung 8) der spezifischen Artefakte für die jeweiligen Szenarien:

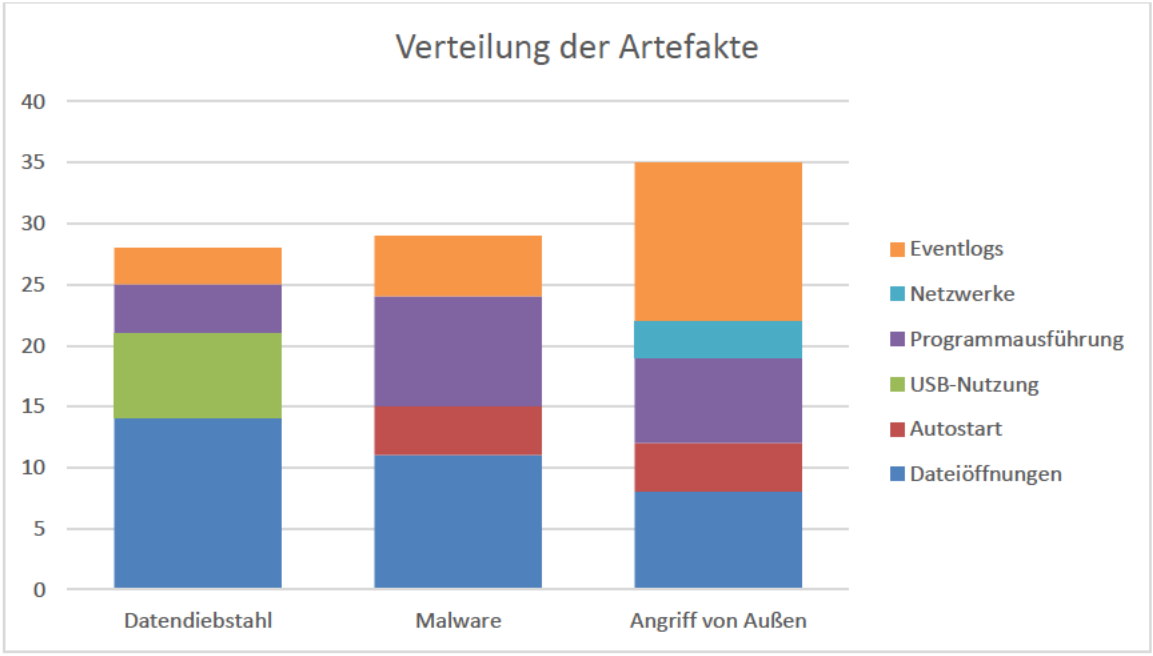


Abbildung 8 - Verteilung der Artefakte

Dabei ist ersichtlich, dass von 50 möglichen spezifischen Artefakten (ohne die allgemeinen Artefakte) in den drei Szenarien jeweils 28-35 Artefakte genutzt wurden. Dies entspricht etwa nur 56%-70% im Vergleich zu der Nutzung aller möglicher Artefakte aus der Vorauswahl. Auffällig ist auch, dass sich die drei Szenarien in der Artefakte-Wahl merklich unterscheiden. Während beim Datendiebstahl die USB-Nutzung untersucht wird, nicht aber die Autostarts und die Netzwerke, wird hingegen bei der Malwareanalyse keine USB-Nutzung untersucht jedoch die Autostarts. Beide vorher genannten Szenarien nutzen aber im Gegensatz zum Szenario Angriff von außen nicht die Netzwerkartefakte. Auch auffällig ist die Verteilung der Eventlogs. Während beim Szenario Datendiebstahl lediglich fünf Eventlogs untersucht werden, sind es bei einem Angriff von außen ganze 13. Wie diese Verteilungen zustande kommen, wird folgend genauer geklärt.

3.1.1 Szenario 1 – Datendiebstahl

In der heutigen Zeit sind geistiges Eigentum und persönliche Daten sehr kostbar und wertvoll. Mehrere Gesetze der DSGVO schützen diese Bereiche und entsprechend sensible Daten [43]. Eine Datenschutzverletzung wird als Diebstahl von sensiblen Daten definiert, wie z.B. geistigem Eigentum oder Datensätze, welche die Namen von Mitarbeitenden oder Kunden sowie weitere Informationen über diese Personen enthalten. Dabei werden am häufigsten personenbezogene Daten gestohlen [6]. Das Abhandenkommen dieser, bringt ein Unternehmen in eine sehr schwerwiegende und teure Situation. Der Ruf des Unternehmens kann sogar nachhaltig geschädigt werden und dadurch kann ein großer wirtschaftlicher Schaden entstehen [6]. Datendiebstahl von Firmeneigentum war auch laut FBI und CSI wiederholt die finanziell schädlichste Kategorie in der Kategorie Computerkriminalität [44].

Um einen Datendiebstahl nachzuweisen, werden unter anderem Artefakte der Kategorie Dateiöffnung genutzt, um zu sehen, ob relevante Dateien und Verzeichnisse überhaupt geöffnet wurden. Diese können dann mit weiteren Artefakten wie der USB-Nutzung oder den E-Mail-Artefakten verknüpft werden, um einen Tathergang möglichst gut zu rekonstruieren. Dabei ist es wichtig, zu wissen, dass keine offensichtlichen Artefakte in Bezug auf die Kopier-Operation von Dateien oder Verzeichnissen existieren. Trotzdem können verschiedene, durch kopieren entstehende Muster erkannt werden. So kann der Unterschied zwischen einem Routinezugriff und einem Kopiervorgang laut J. Grier anhand verschiedener Eigenschaften ausgemacht werden [44]. Es wird zwischen selektivem und nicht selektivem Zugriff und temporär unregelmäßigem und kontinuierlichem Zugriff unterschieden. Dies resultiert daraus, dass bei einem Routinezugriff meist nur einige Dateien geöffnet werden und bei einem Kopiervorgang hingegen jede Datei und jedes Unterverzeichnis angesprochen wird, was dann in den jeweiligen Zeitstempeln ersichtlich ist. [44]

Hinsichtlich Dateiöffnungen können in dem Artefakt „Shellbags“ zuletzt aufgerufene Dateien und Verzeichnisse sowohl lokal als auch in Netzwerken oder über externe Datenträger inklusive Zeitstempel vorgefunden werden, wobei auch Strukturen von bereits gelöschten Verzeichnissen und Dateien vorhanden sein können [17]. Weitere Hinweise auf Dateiöffnungen befinden sich unter „RecentDocs“, „LNK-Files“ und in der Browser-History, welche auch lokale Dateizugriffe beinhaltet [17]. Spezifisch für Microsoft Office Dateien gibt das Artefakt „OfficeUserMRU“ sowie das Microsoft Office Log Auskunft über Öffnungen und Veränderungen. Auch die beiden Artefakte unter „ComDlg32“ mit den Namen „LastVisitedMRU“ und „OpenSaveMRU“ geben Details über geöffnete Dateien preis, indem sie Dateien erfassen, welche innerhalb des Windows-Shell-Dialogfelds geöffnet oder gespeichert werden. Zudem werden .exe-Dateien erfasst, die von Anwendungen verwendet werden, um Dateien zu öffnen [45]. Auch das Artefakt „WordWheelQuery“ kann ein Hinweis auf einen Datendiebstahl geben, da es in der Taskbar gesuchte Begriffe beinhaltet [17]. In den „TypedPaths“ können zuletzt eingegebene Pfade für den File-Explorer festgestellt werden [46]. Wurden dort

verdächtige Begriffe gesucht und sind diese in weiteren Artefakten der Dateiöffnungen wiederzufinden, bedeutet dies, dass diese Dateien mit hoher Wahrscheinlichkeit zumindest geöffnet wurden. In seltenen Fällen werden auch in den „Thumbcaches“ Hinweise auf einen möglichen Datendiebstahl gefunden. Die Thumbcache Datenbanken speichern verkleinerte Versionen von Bildern, welche im System als Vorschaubilder genutzt werden [47]. Nicht zu vergessen ist auch der Papierkorb („Recycle Bin“), dort können sich gelöschte relevante Dateien oder Spuren von Programmen finden, welche für einen Datendiebstahl sprechen [17].

Insgesamt geben die Dateiöffnungen allein aber keine ausreichende Beweiskraft ab, um einen Datendiebstahl nachzuweisen. Dafür bedarf es an weiteren Artefakten, welche mit den bereits gewonnenen Ergebnissen korrelieren. Dabei gibt es verschiedene Möglichkeiten des Datendiebstahls, denn dieser kann entweder über die Kopie auf einen Wechseldatenträger geschehen, über einen Upload auf verschiedene Cloud-Dienste oder Dateien können per Chat oder E-Mail versendet werden. Jeder Benutzer mit einem Wechseldatenträger hat dabei die Möglichkeit, diesen zu nutzen, um Daten von einem System zu kopieren. Deshalb spielen vor allem die Artefakte der USB-Nutzung eine merkliche Rolle. Zu den grundlegenden Artefakten gehören dabei „USBSTOR“, „USB“ und „EMDMgmt“, welche eine Liste der angeschlossenen Geräte, inklusive Details bietet [17,20]. Letztere gibt dabei die Volume Serial Number an, welche nicht zu verwechseln ist mit der Seriennummer. Genauere Informationen wie der zugeordnete Laufwerkbuchstabe und das jeweils letzte Gerät, welches einem Laufwerkbuchstaben zugeordnet werden kann, kann aus den Artefakten „Mounted Devices“ und „Devices“ gewonnen werden [17]. Der Laufwerkbuchstabe kann so, je nach Möglichkeit und gegebenen Artefakten, zeitlich mit den Dateiöffnungen unter diesem Laufwerkpfad verglichen werden. Um den Benutzer zu bestimmten, der angemeldet war, als ein bestimmtes USB-Gerät angeschlossen wurde, ist das Artefakt „MountPoints2“ hilfreich [17,20]. Hierbei muss aber darauf geachtet werden, dass ein Benutzerkonto nicht eindeutig einer Person zugeordnet werden kann. [1]

Die Validierung der Ergebnisse kann aber auch durch weitere Artefakte wie Programmausführungen, Webzugriffe oder E-Mail-Artefakte geschehen. Webzugriffe können in diesem Szenario Aufschluss über potenziell genutzte Cloud-Dienste oder Webmail-Dienste für den Datentransfer geben. Auch in E-Mail-Artefakten kann nach relevanten Dateien gesucht werden, denn auch dies ist ein Weg, Daten zu transferieren. Dabei kommt es häufig vor, dass Daten von einer geschäftlichen Adresse auf eine private E-Mail-Adresse weiteregeleitet werden. Ferner werden auch einige Eventlogs betrachtet (Tabelle 3), welche Hinweise auf ausgehende Network-Shares liefern können, die auch für einen Datendiebstahl genutzt werden können. Aber nicht nur Web- oder E-Mail-Dienste können nützlich sein, es gibt auch weitere Programme die einen Anfangsverdacht bestätigen können. Deshalb werden bei einem Datendiebstahl auch einige Artefakte aus der Kategorie Programmausführung analysiert. So zum Beispiel die Artefakte „UserAssist“, „AppCompatCache“ und „AmCache“, welche aktuelle Programmausführungen sowie Informationen über die Installation und die Kompatibilität

von Programmen bereitstellt [17]. So können sowohl Programme für den Datentransfer als auch Antiforensik-Tools, die Spuren verwischen sollen, festgestellt werden.

3.1.2 Szenario 2 – Malware

Der Begriff „Malware“ ist Englisch und stammt von einer Verbindung der beiden Wörter „Malicious“ (deutsch: bösartig) und Software (deutsch: Programm) ab. So kann Malware als schadhaftes Programm übersetzt werden [48]. Dabei kann zwischen verschiedenen Malware-Arten unterschieden werden. So gibt es z.B. Viren und Würmer, welche die Eigenschaft haben, sich selbst replizieren zu können. Als Trojaner werden Schadprogramme bezeichnet, welche sich als nützliche Anwendungen tarnen und mit Spyware sind Programme gemeint, welche persönliche (sensible) Daten sammeln und das Benutzerverhalten protokollieren. Weiterhin gibt es sogenannte Scareware, welche für gewöhnlich ein Sicherheitsproblem vortäuscht, um anschließend kostenpflichtige Sicherheitssoftware zu verkaufen. Bei der Adware wird dem Benutzer durch schadhafte oder modifizierte Software besonders viel Werbung gezeigt. Die wohl bekannteste Art von Schadsoftware ist aber die Ransomware. Diese verschlüsselt Dateien und verhindert den Zugang zum gesamten Rechner oder führt sogar zu einer Sperrung anderer in einem Netzwerk erreichbarer Geräte wie z.B. ein internes Firmennetzwerk. Durch sogenannte „Double Exortion“ können bei unterschiedlichen Arten von Ransomware nicht nur Systeme verschlüsselt werden, sondern es ist auch möglich, dass sensible Daten ausgeleitet werden und damit gedroht wird, diese zu veröffentlichen. So leiden Betroffene unter noch größerem Druck, da die Vertraulichkeit von Daten gefährdet ist. Zuzüglich hat sich auch das Modell „Ransomware as a Service“ etabliert, bei welchem eine Gruppe von Kriminellen eine Ransomware programmiert und Operatoren anheuert, welche unter Beteiligung am Gewinn, die Software auf das Zielsystem laden. [49,50]

Laut dem „SANS 2022 Ransomware Defense Report“ [4] waren die Jahre 2020 und 2021 unzweifelhaft die Jahre der Ransomware. Dabei hat auch die Agentur der europäischen Union für Cybersicherheit (ENISA) festgestellt, dass Ransomware-Angriffe zwischen April 2020 und Juli 2021 um 150% zugenommen haben [4]. Zusätzlich wird auch im „Cybercrime Bundeslagebild 2019 über eine steigende Tendenz von Ransomware-Attacken berichtet [50]. Umso wichtiger ist es deshalb, eine Möglichkeit zu bieten, solche Vorfälle möglichst schnell zu untersuchen, denn solche Angriffe haben das Potenzial, existenzielle Bedrohungen auszulösen. Konkret handelt es sich strafrechtlich gesehen beim Einsatz von Ransomware um eine Kombination der Delikte Computersabotage gemäß §303b StGB und Erpressung gemäß §253 StGB [50].

Um eine Infektion mit Malware nachzuweisen, gibt es verschiedene wichtige Artefakte, welche analysiert werden können. Die erste Phase des Malwareangriffs startet dabei typischerweise mit dem Versand einer Phishing-Mail [50]. Diese kann entweder schadhafte Dokumente beinhalten, welche eine Schadsoftware nachladen oder der Benutzer kann auf eine gefälschte Website weitergeleitet werden und wird dort dazu

verleitet, seine Anmeldedaten einzugeben, welche dann kompromittiert werden. Malware kann aber auch durch eine ausgenutzte Sicherheitslücke in die verschiedenen genutzten Systeme gelangen. Deshalb spielen in diesem Szenario sowohl Artefakte zu Programmausführungen und Dateiöffnungen, als auch Artefakte über Webzugriffe und E-Mails eine merkliche Rolle. In der Tabelle 3 ist sichtbar, dass alle Artefakte zu Dateiöffnungen analysiert werden sollen, bis auf drei folgend erläuterte Artefakte. Gesuchte Begriffe in der Taskbar sind bei der Suche nach Malware in der ersten Triage wenig hilfreich, weshalb die „WordWheelQuery“ nicht analysiert wird. Auch die lokalen Dateizugriffe in der Browser-History werden in diesem Falle nicht mitgesichert, da es in den weiteren Artefakten in der Regel genügend Hinweise auf die Öffnung einer verdächtigen Datei gibt. Ebenfalls für den Anfang wenig aufschlussreich sind die „Thumbcaches“, da diese für die erste Triage keine hilfreichen Hinweise für das weitere Vorgehen beinhalten. Dennoch können auch diese Artefakte in der späteren Untersuchung relevant sein, denn in den „Thumbcaches“ können auch Icons von schadhaften Programmen gefunden werden. Da es sich bei der Untersuchung mittels KAPE jedoch um eine Triage handelt, wird sich auf die wichtigsten Artefakte, welche das weitere Vorgehen und die weitere Untersuchung einleiten, beschränkt.

Selbstverständlich wird in den Artefakten der Programmausführung nach malwarebezogenen Spuren gesucht. Dabei können auch bereits gelöschte Programme festgestellt werden. Im Artefakt „AppCompatCache“ kann jede ausführbare Datei, die zu einem Zeitpunkt auf dem System gestartet wurde, nachgewiesen werden [17]. Durch vorhandene Zeitstempel ist es sogar möglich, die letzte Aktivität verdächtiger Programme auszumachen und so eine erste zeitliche Eingrenzung vorzunehmen. Der „AmCache“ speichert zu jeder ausführbaren Datei auch ein SHA-1 Wert, welcher überprüft werden kann, sowie der Pfad, indem sich die Datei befindet [17]. Dies kann Aufschluss darüber geben, wo sich im Dateisystem ein schadhaftes Programm befindet und dementsprechend kann in diesen Verzeichnissen nach weiteren Hinweisen gesucht werden. In den „Jumplists“ kann jeweils der erste und letzte Zeitpunkt der Ausführung eines Programms festgestellt werden [17]. Zusätzlich dazu gibt das Artefakt „Prefetch“ Auskunft darüber, wie oft ein Programm ausgeführt wurde und unter welchen Geräte- und Dateihandhabungen dies stattfand. Ausführbare Dateien mit einer grafischen Benutzeroberfläche, welche vom Desktop aus gestartet werden, sind in den „UserAssist“ für jeden Benutzer einzeln zu finden [17]. Unter „RunMRU“ und „Powershell Console History“ sind die zuletzt ausgeführten Befehle sichtbar [20,28]. Auch die Systemperformanz und Hintergrundaktivitäten können wichtige Hinweise auf einen möglichen Befall mit Schadsoftware geben. Ist eine außergewöhnlich hohe Auslastung im „SRUM“ (System Ressource Usage Monitor) oder im „BAM“ (Background Activity Monitor) auszumachen, kann dies auf das Laufen einer Schadsoftware zurückzuführen sein. Deshalb werden auch diese Artefakte in die erste Triage miteinbezogen.

Um gegebenenfalls Quellcode und ausgeführte Befehle der Malware zu identifizieren, können verschiedene Eventlogs ausgelesen werden. So können unter dem Eventlog

`Windows Powershell.evtx` und `Microsoft-Windows-Powershell%4Operational.evtx` je nach Konfiguration auffällige Skripte gespeichert werden. Durch das Eventlog `Microsoft-Windows-WinRM%4Operational.evtx` können unter verschiedenen Event-IDs (siehe Tabelle 1) auch weitere Informationen über das PowerShell-Remoting gefunden werden. Außerdem ist das `Microsoft-Windows-WMI-Activity%4Operational.evtx` nützlich, um temporäre und permanente Events zu erkennen. [28]

Es gibt durchaus Artefakte, welche die Malware selbst direkt auf dem System erstellt. Dies können bereits erwähnte Programme oder Dateien sein, es können aber auch Registry-Einträge durch eine Malware entstehen [3]. In der Registry werden deshalb mit hoher Priorität die geplanten Aufgaben, Dienste und System- sowie Benutzerautostarts untersucht. Diese geben Informationen über Programme, die in regelmäßigen Abständen beim Starten des Computers oder bei der Anmeldung eines Benutzer gestartet werden sollen [20]. Schadsoftware verwendet die Funktion der Autostarts auch, um sich eine Persistenz zu verschaffen und sicher zu gehen, dass sie weit genug im System verankert ist, um einen Neustart zu überleben. Verdächtige Programme, welche automatisch Starten oder in den geplanten Aufgaben auftauchen, können mit anderen Artefakten im Kreuzvergleich Aufschluss über die Schadhaftigkeit geben. Die Korrelation verschiedener Artefakte kann dann bestenfalls nicht nur die Quelle der Attacke, sondern auch das Verhalten und Vorgehen der Malware offenbaren [51].

3.1.3 Szenario 3 – Angriff von außen

Ein Angriff von außen wird in der Forensik auch als „Intrusion“ bezeichnet. Eine Intrusion kann folgendermaßen definiert werden: „Ein Sicherheitsereignis oder eine Kombination mehrere Sicherheitsereignisse, das einen Sicherheitsvorfall darstellt, bei dem ein Eindringling Zugang zu einem System oder einer Systemressource erlangt oder zu erlangen versucht, ohne dazu berechtigt zu sein“ [52]. Dabei ist der Begriff immer relativ in Bezug auf die Sicherheitsrichtlinien definiert, denn diese entscheiden schlussendlich, was auf einem System verboten oder erlaubt ist [53]. Es gibt bei diesem Szenario einige Überschneidungen mit der Analyse von Malware, da Malware für gewöhnlich auch durch einen Angriff von außen initiiert wird. So kann, wenn Malware gefunden wurde, auch weiter nach dem ursprünglichen Angriff unter dem Szenario „Angriff von außen“ geforscht werden. Das Kapitel 3.1.2 kann also auch als Vertiefung eines Angriffs von außen gesehen werden, wobei ein Angriff nicht immer nur bedeuten muss, dass Schadsoftware abgelegt wurde. Er kann auch stattfinden, um Systeme zu verändern, Daten zu entwenden, auszuspionieren oder die Erreichbarkeit zu gefährden (Denial of Service Attacken). Im Buch „Incident Response & Computer Forensics“ von J. T. Luttgens [3] wird das Konzept des Lebenszyklus eines Angriffs folgendermaßen aufbereitet: Nach dem ersten Angriff schaffen sich die Angreifer ein festes Standbein, um dann ihre Rechte auszuweiten. So finden interne Erkundungen und Seitwärtsbewegungen (Lateral

Movement) statt. Abschließend gilt es, die Präsenz aufrecht zu erhalten und die ursprünglichen Ziele des Angriffs zu erreichen. [3]

Um Angriffe von außen zu erkennen, kommen vor allem Eventlogs zum Einsatz. In der Tabelle 3 ist ersichtlich, dass alle Eventlogs der vorgetroffenen Auswahl genutzt werden. Ob diese Protokolle allerdings Einträge besitzen oder gar vorhanden sind, hängt von der Systemkonfiguration und dem Einsatz von Antiforensik ab. Sofern vorhanden, können sie auf unübliche Logeinträge untersucht werden. Das können Verbindungen von ungewöhnlichen Orten oder verdächtige Zeitstempel mitten in der Nacht sein. Um Anomalien zu erkennen, muss aber bekannt sein, welche Verbindungen normal sind. Dazu bedarf es einer guten Kommunikation mit der jeweiligen internen IT-Abteilung, um diese Basislinien zu erkennen und abweichendes Verhalten möglichst schnell zu identifizieren. Ein erster wichtiger Anhaltspunkt stellt dabei das `Security.evtx` dar. Dort können sowohl verschiedene Anmeldetypen inklusive der Quellnetzwerkadresse als auch fehlgeschlagene Anmeldungen und neu angelegte Benutzer identifiziert werden [24]. Neu angelegte Benutzer, welche der internen IT-Abteilung nicht bekannt sind, können auf einen Persistenz Mechanismus hinweisen und sofern sie nicht entfernt werden, dem Angreifenden wieder eine Möglichkeit bieten, einen Eintrittspunkt in das System zu bekommen. Wenn es sich um einen Fernzugriff handelt, findet dieser entweder über Remote-Desktop-Verbindungen oder über sogenannte Map-Network-Shares statt. Die in der Tabelle 3 unter „RDP“ zusammengefassten Eventlogs sind verantwortlich dafür, ein- und ausgehende Remote-Verbindungen zu protokollieren. Ziel-Hostnamen, Ziel- und Quell-IP-Adressen sowie Anmeldenamen des Benutzers können so festgestellt werden [28]. Weiterhin sind auch Verbindungsversuche und erfolgreiche Verbindungen protokolliert. Durch die unter „SMB“ zusammengefassten Eventlogs (Tabelle 3), können ausgehende Map Network Shares festgestellt werden. Darunter sind fehlgeschlagene Anmeldungen sowie Informationen zu Netzwerkverbindungen und unautorisierte Zugriffsversuche eingetragen [28,54]. Bei den Eventlogs zur Powershell („PS“ unter Tabelle 3) handelt es sich um eine Fernausführung, dort ist vor allem das Log Windows `PowerShell.evtx` hervorzuheben. Unter verschiedenen Event-IDs können diese Informationen über den Start und das Ende einer Remote Session sowie auffälligen Skriptcode enthalten [28]. Hat der Angreifende einer dieser Wege genutzt und keine Antiforensik zur Löschung oder Überschreibung der Logfiles betrieben, hat er dort wichtige Spuren, gegebenenfalls auch zu Seitwärtsbewegungen (Lateral Movement), hinterlassen. [55,56]

Wie bereits beschrieben, ist ein wichtiger Teil des Angriffs, die Verschaffung von Persistenz. Diese wird oft angestrebt, um trotz Neustart oder Änderung von Zugangsdaten einen Zugriff zu behalten. Deshalb werden die Artefakte aus der Registry wie Dienste, geplante Aufgaben, sowie System- und Benutzerautostarts analysiert [57]. Dort wird nach unüblichen Prozessen und geplanten Aufgaben Ausschau gehalten. Dabei ist vor allem auf Programme zu achten, welche unbekannt sind und sich nicht in Systemverzeichnissen befinden [28,53].

Auch die Programmausführungen sollten bei einem Angriff von außen untersucht werden. Für dieses Szenario werden deshalb alle Artefakte, bis auf „SRUM“ und „BAM“, welche Informationen über Systemressourcen und Hintergrundaktivitäten enthalten, verwendet [17]. Vom Angreifenden genutzte Programme können Hinweise auf sein weiteres Vorgehen sowie auf den Eintrittspunkt geben. Auch hier bedarf es der Detektion einer Grundlinie, um zu wissen, welche Dateien verdächtig sind oder von der Norm abweichen. Ein übliches Programm, welches nicht selbst installiert wurde, ist auch verdächtig [53] . Es wird aber auch nach Überwachungsprogrammen gesucht. Auch normale Programme wie z.B. Virenscanner oder Systemdateien können sich durch kleine Unterschiede wie z.B. eine fehlende Signatur oder einen merkwürdigen Speicherort als Schadsoftware identifizieren lassen.

Können in den vorherigen Artefakten verdächtige Dateien, Programme oder Skripte ausgemacht werden, ist es sinnvoll, sich auch die Dateiöffnungen näher anzuschauen. Durch diese Artefakte kann man bei gegebenen Anhaltspunkten Rückschlüsse auf das Einfallstor ziehen und als nächsten Schritt auch die Webzugriffe und E-Mail-Artefakte untersuchen. Dort kann dann gegebenenfalls der Ursprung einer schadhaften Datei oder einer Phishing-Attacke ausgemacht werden. Hinsichtlich der Dateiöffnungen werden die Artefakte „OfficeUserMRU“, „Shortcuts“, „RecycleBin“ sowie Dateizugriffe in der Browser-History und im Microsoft Office Log aus der verfügbaren Sammlung in Tabelle 3 gewählt [17]. Diese geben Informationen zu zuletzt bzw. häufig geöffneten Dateien sowie Remote-Dateizugriffen (Dateizugriffe in der Browser-History) und gelöschten Dateien. Außerdem sind auch die „Typed Paths“ relevant, da diese zuletzt eingegebene Pfade im Datei Explorer speichern. Zusätzlich werden auch Angaben zu Netzwerken in Form verschiedener Artefakte gesichert. So können Netzwerke identifiziert werden, welche mit dem Computer verbunden waren und auch zu welcher Zeit dies geschah [20].

3.2 Vorbereitung und Anpassung von KAPE

In diesem Kapitel geht es darum, das Tool KAPE soweit vorzubereiten und anzupassen, dass es für den Einsatz geeignet ist. Dabei wird ein großer Wert darauf gelegt, die forensischen Prinzipien einzuhalten und eine entsprechende Dokumentation durch KAPE selbst zu erreichen. Die gesamte Arbeit mit KAPE geschieht auf einem Windows-System, da KAPE nur unter Windows verfügbar ist. Dabei handelt es sich um Windows 10 Pro in der Version 21H2.

3.2.1 Installation und Update

Nach dem Download wurde KAPE mit dem Programm 7ZIP [58] entpackt und in einen entsprechenden gleichnamigen Ordner verschoben. Es muss nichts weiter installiert werden. Abbildung 9 zeigt dabei den Inhalt des entpackten Ordners inklusive des im Folgenden selbst erstellten Verzeichnisses `\output`.

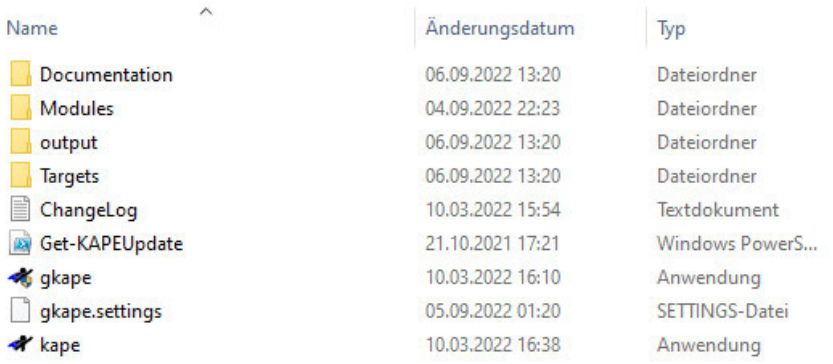


Abbildung 9 - KAPE Ordnerstruktur

Unter `\Documentation` befindet sich dabei eine .txt-Datei mit Verlinkungen zur offiziellen Dokumentation. Im Ordner `\Module` befinden sich die verschiedenen vorgefertigten Module und auch Modul-Vorlagen inklusive einer Anleitung dazu sowie ein Verzeichnis `\bin`, welches die ausführbaren Dateien der entsprechenden Module enthalten. Unter `\Targets` befinden sich ebenfalls verschiedene vorgefertigte Target-Dateien sowie Target-Vorlagen und auch hier ist eine Anleitung dazu vorhanden. Das `ChangeLog.txt` gibt die aktuelle Version inklusive aller Änderungen aus. Schließlich befinden sich im Verzeichnis noch die `kape.exe` Datei und die grafische Oberfläche namens `gkape.exe`.

Um sicherzugehen, dass sich KAPE sowie die Targets und Module auf dem neusten Stand befinden, wird das mitgelieferte Skript `Get-KAPEUpdate.ps1` ausgeführt. KAPE meldet Änderungen und Aktualisierungen an bestehenden Konfigurationen, indem ein Vergleich Mittels SHA-1 stattfindet [33]. Die lokale Version und die aktuelle Version stimmen überein, sodass kein Aktualisierung vorgenommen werden muss. Wie in der Abbildung 10 zu erkennen ist, handelt es sich dabei um die Version 1.2.0.0.

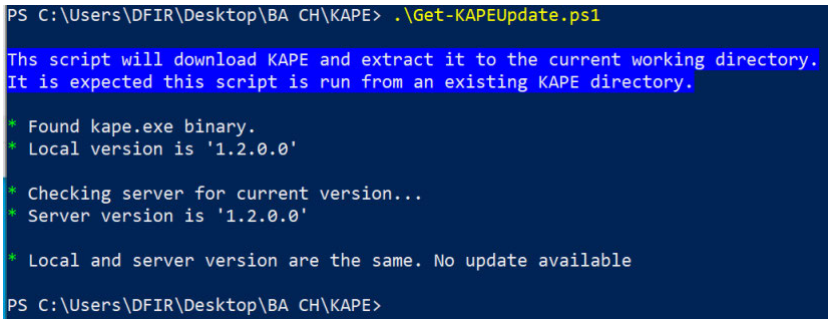


Abbildung 10 - Ausführung Get-KAPEUpdate.ps1

Zusätzlich wird mit dem Befehl `kape.exe --sync` nach Aktualisierungen für die Targets und Module gesucht. Dabei konnten mehrere neue und aktualisierbare Targets und Module festgestellt werden, welche entsprechend heruntergeladen wurden. Bei einer zweiten Überprüfung des Befehls konnten keine neuen Targets oder Module mehr festgestellt werden. Somit ist KAPE auf dem neusten Stand und es kann mit der Arbeit begonnen werden.

3.2.2 Grundlegende Konfigurationen

Um die Target- und Modul-Ausgabe einfach und übersichtlich zu halten, werden zu Beginn zwei neue Verzeichnisse im ebenfalls neu angelegten Verzeichnis `KAPE\output` erstellt. Diese werden `\tout` für die Target Ausgabe und `\mout` für die Modul Ausgabe genannt. Zusätzlich wird in den `\Targets`- und `\Modules`-Verzeichnissen jeweils ein Ordner mit dem Namen `ICS` und ein Ordner mit dem Namen `ICS_Compound` angelegt, um die selbst erstellten Targets und Module sowie die Compound-Konfigurationen direkt darin abzulegen. Dies hat zur Folge, dass die Dateien in der grafischen Nutzeroberfläche von KAPE eine eigene Kategorie erhalten, nach welcher gefiltert werden kann. Zusätzlich werden alle vordefinierten Targets und Module in das jeweilige `\!Disabled`-Verzeichnis geschoben, damit diese nicht mehr in `gkape.exe` sichtbar sind. Dies dient der Übersichtlichkeit.

Die jeweiligen Versionen von KAPE werden nach dem Erstellen der Targets und der Module in Versionsschritten auf einem externen Datenträger abgespeichert, um bei einem Verlust von Daten oder einer falschen Konfiguration zum letzten Sicherungspunkt zurückkehren zu können.

Während eines Incident Response soll möglichst auf die Verwendung der grafischen Nutzeroberfläche verzichtet werden. So ist das Ziel, KAPE auf dem Analysesystem so vorzubereiten, dass auf dem zu analysierenden System nur noch eine Kommandozeile aufgerufen und ein Befehl ausgeführt werden muss [59]. Die Dateien sollen dann lediglich auf einen externen Datenträger kopiert werden (mittels Targets) und noch nicht geparst. Das Parsen (durch die Module) findet dann unter der Nutzung von KAPE und den bereits gespeicherten Target-Ausgabedateien auf dem Analyserechner statt.

3.3 Erstellung der Targets

In diesem und den folgenden Abschnitten geht es um den Kern der Arbeit. Die in 3.1 gewählten Artefakte werden im Tool KAPE umgesetzt. Dabei ist der Ansatz, zu den drei Szenarien jeweils alle notwendigen Targets zu erstellen, um diese dann den drei Target-Compounds zuzuordnen, welche alle gewünschten Artefakte für ein jeweiliges Szenario kopieren. Dies geschieht unter Zuhilfenahme der Tabelle 1, um die jeweiligen Pfade der Artefakte zu erhalten. Außerdem wird die mitgelieferte Vorlage TargetTemplate.template genutzt, welche bereits unter 2.4.1 in Abbildung 6 ersichtlich ist.

Bei der Erstellung der Targets sollen immer nur Artefakte der gleichen Kategorie zusammengefasst werden [31]. Diese erstellten .tkape-Dateien können dann in weiteren sogenannten Compound-Target-Dateien zusammengefasst werden. Die Benennung der Targets erfolgt nach dem Schema „ICS_%Artefaktname%.tkape“ und die Beschreibung ihres Inhalts wird, wie in Abbildung 11 ersichtlich, nach einem festgelegten Schema

aufgebaut. Die Speicherung der jeweiligen Targets erfolgt unter dem Verzeichnis `KAPE\Targets\ICS\`.

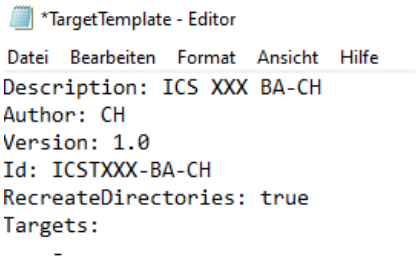


Abbildung 11 - Individualisiertes Target-Template

In der Beschreibung (Description) wird der Platzhalter XXX durch das jeweilige Artefakt ersetzt, welches kopiert werden soll. Sofern es ein selbst verfasstes Target ist, wird der Autor mit dem Kürzel „CH“ angegeben. Da es sich hierbei um die erste Bearbeitung handelt, wird die Version auf 1.0 gesetzt. Um die einzigartige „Id“ einheitlich zu halten, wird sie mit ICSTXXX-BA-CH angegeben, wobei XXX jeweils um eine Stelle inkrementiert wird. „RecreateDirectories“ wird dabei auf dem booleschen Wert „True“ belassen, damit die ursprüngliche Verzeichnisstruktur im Zielverzeichnis beibehalten wird und die Dateien durch die üblichen Modulpfade geparst werden können. [31]

Das allgemeine Vorgehen bei der Erstellung eines Targets ist, nach der Vergabe der akkuraten Beschreibung, die eigentliche Erstellung der Targets. Dabei sind sowohl der Name, als auch der Pfad worunter sich das Artefakt befindet (ersichtlich in Tabelle 1), erforderlich. Zusätzlich muss das Artefakte in eine Kategorie eingeordnet werden. Dabei werden die in Tabelle 1 und 3 verwendeten Kategorien verwendet. Weitere optionale Konfigurationen werden, wenn notwendig, für das jeweilige Artefakte folgend genauer ausgeführt. In Abbildung 12 ist beispielhaft die erstellte Konfigurationsdatei des Artefaktes $MFT zu sehen.

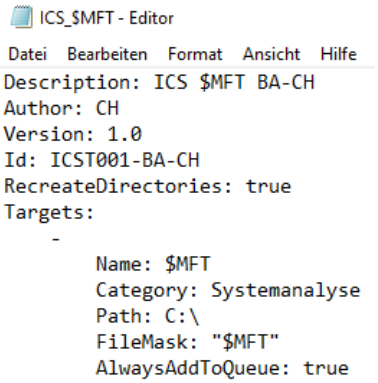


Abbildung 12 - ICS_$MFT.tkape

Hierbei ist in der Beschreibung zu erkennen, dass der $MFT die „Id“ ICST-001-BA-CH zugeordnet wurde. Des Weiteren hat sie den Namen $MFT und die Kategorie (ersichtlich unter Tabelle 1) Systemanalyse zugeordnet bekommen. Der Pfad `C:` wurde ebenfalls aus

der Tabelle 1 entnommen. In dieser Konfiguration ist zu sehen, dass auch eine „FileMask“ vergeben wurde. So sucht das Target dann spezifisch nach diesem Namen. Zusätzlich wurde die Funktion „AlwaysAddToQueue“ gewählt. Da sich die $MFT möglicherweise bei einer Live-Sicherung noch in Benutzung befindet, wird so sichergestellt, dass sie trotzdem kopiert wird. Folgend wird die Erstellung der weiteren einzelnen Targets, nach Speicherort abgegrenzt, beschrieben. Die weiteren Target-Konfigurationsdateien sind von Abbildung 13 bis 17 abgebildet.

3.3.1 Dateisystem

Für den Speicherort Dateisystem sollen die Artefakte mit den jeweiligen Pfaden aus Tabelle 1 in die Targets eingespeist werden: Für einige Artefakte existieren schon vorgefertigte Targets, welche aber bezüglich Kategorien und weiteren Einzelheiten erneut angepasst werden sollen. Die Konfiguration der $MFT wurde bereits unter Abbildung 12 beschrieben. Das grundlegende Vorgehen der Erstellung ist bei jedem der Targets gleich, weshalb folgend nur noch auf Besonderheiten der Erstellung der einzelnen .tkape-Dateien eingegangen wird. Folgende Targets wurden erstellt:

- ICS_$MFT.tkape
- ICS_Amcache.tkape
- ICS_Jumplist.tkape
- ICS_LNK.tkape
- ICS_OutlookPSTOST.tkape
- ICS_Prefetch.tkape
- ICS_PS_ConsholeHistory.tkape
- ICS_RecycleBin.tkape
- ICS_SRUM.tkape
- ICS_Thumbcache.tkape

Dabei wurden jeweils verschiedene Konfigurationsmöglichkeiten genutzt. Zum einen wurde die Option „Recurisve“ genutzt, um darüber zu entscheiden, ob alle weiteren Unterverzeichnisse auch mitkopiert werden sollen und zum anderen wurde die Option „FileMask“ genutzt, um nach einer Datei mit einem definierten Namen zu suchen. Weiterhin wurde sowohl in der „FileMask“ als auch in den anzugebenden Pfaden mit Platzhaltern gearbeitet. Das Symbol „*“ steht dabei für eine beliebige Anzahl Zeichen und kann so mehrere Dateien oder Verzeichnisse miteinbeziehen. Die Dateien unter dem Artefakt „Prefetch“ haben beispielsweise beliebige Namen, enden aber alle mit „.pf“. Unter der „FileMask“ wird so der Wert „*.pf“ angegeben.

Ein anderes Beispiel stellt das Artefakt „RecycleBin“ (Abbildung 13) dar. Dabei werden zwei verschiedene Dateien jeweils mit einem Platzhalter gesichert. Es handelt sich dabei um die „$I*“, welche die jeweiligen Metadaten wie z.B. den Pfad der Datei, Zeitstempel und die Dateigröße enthält. Weiterhin werden die Dateien unter der „FileMask“ „$R*“

gesichert, welche die eigentliche gelöschte Datei beinhalten [60]. Der Platzhalter unter den Pfaden (`C:\$Recycle.Bin\*\`) steht dabei für die jeweilige Benutzer SID (Secure Identifier), denn jeder Benutzer besitzt einen eigenen Papierkorb.

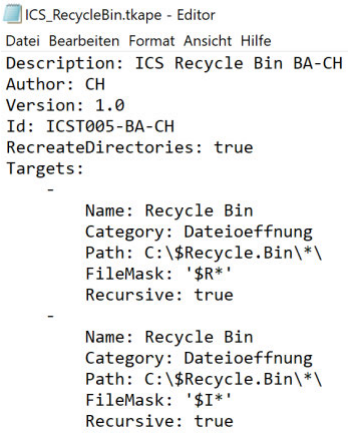


Abbildung 13 - ICS_RecycleBin.tkape

Die Webzugriffe für das Artefakt „Browser-History“ sind in vier .tkape-Dateien für den jeweiligen Browser unterteilt:

- ICS_Chrome_Browserhistory.tkape
- ICS_Edge_Browserhistory.tkape
- ICS_Firefox_Browserhistory.tkape
- ICS_IE_Browserhistory.tkape

Bei der folgenden Abbildung 14 handelt es sich um die .tkape-Datei für den Edge-Webbrowser.

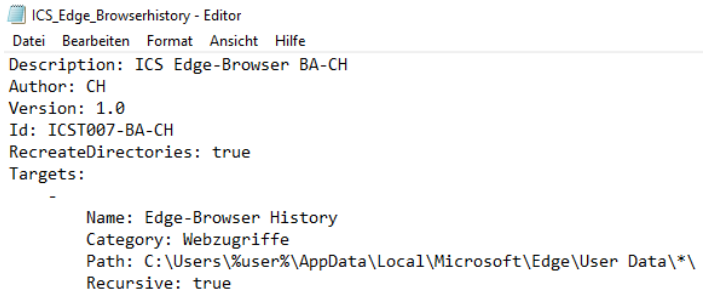


Abbildung 14 - ICS_Edge_Browserhistory.tkape

Die weiteren drei Targets für die Browser Internet-Explorer, Firefox und Google Chrome sind identisch aufgebaut und unterscheiden sich nur in den Pfaden, welche für die jeweiligen Browser der Tabelle 1 entnommen werden können. Auch hier wird mit mehreren Platzhaltern gearbeitet. Zum einen gibt es den Platzhalter %user%, welcher dafür sorgt, dass die jeweiligen Webverläufe aller Benutzer gesichert werden, zum anderen gibt es einen Platzhalter „*“ am Ende des Pfades, um sicher zu gehen, dass alle Ordner unter User Data mitgesichert werden.

Um die Artefakte unter der Kategorie Webzugriffe übersichtlicher zu gestalten, wurden diese wie folgt (Abbildung 15) zu einem Compound zusammengefasst. Dies vereinfacht die Auswahl, da so nur noch diese Target-Datei anstelle der vier einzelnen Dateien angewählt werden muss.

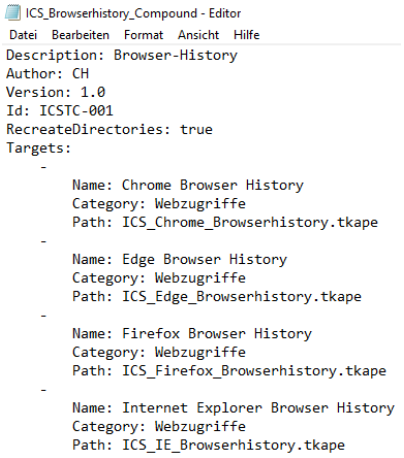


Abbildung 15 - ICS_Browserhistory_Compound.tkape

3.3.2 Registry

Die Registry-Hives beinhalten mehrere Artefakte und werden in den Modulen für die einzelnen Artefakte spezifisch weiterverarbeitet und geparst, weshalb sie für alle Szenarien mit gesichert werden. Darin befinden sich alle Artefakte, welche in Tabelle 3 mit dem Tool RECmd weiterverarbeitet werden, zusätzlich der „Shellbags“ und dem „AppCompatCache“, welche zwar einen eigenen Parser haben, ihre Daten aber auch aus der Registry beziehen. Die Registry-Hives werden im Rohformat kopiert und können so auch mit dem RegistryExplorer der Eric Zimmerman Tools untersucht werden.

Dabei handelt es sich um folgende Dateien:

- SYSTEM
- SOFTWARE
- SAM
- NTUSER.dat
- UsrClass.dat

Für diese Dateien existieren bereits mehrere Compound-Targets, welche von Eric Zimmerman erstellt wurden. In der „RegistryHivesSystem.tkape“ befinden sich unter anderem die SAM-, SOFTWARE-, und SYSTEM-Hives, welche verarbeitet werden sollen. In der „RegistryHivesUser.tkape“ finden sich unter anderem Konfigurationen für die NTUSER.dat und die UsrClass.dat. Des Weiteren beinhalten diese Dateien aber auch Targets für mehrere verschiedene Betriebssystemversionen sowie weitere Hives welche in dieser Arbeit nicht genutzt werden. Da sich diese Arbeit nur auf Windows 7 und höher beschränkt und der zeitliche Aspekt eine merkliche Rolle spielt, werden diese

vorgefertigten Targets nicht in dieser Form verwendet, sondern umgeschrieben. Nur die relevanten Targets werden in ein eigenes Target ICS_Hives.tkape kopiert und angepasst. Zusätzlich werden für die Hives jeweils die Transaktionsprotokolle kopiert. Diese sind an der Endung .LOG* zu erkennen und darin werden alle Transaktionen und Datenbankänderungen aufgezeichnet. Bei Systemfehlern werden Transaktionsprotokolle genutzt, um die Datenbank wieder in einen konsistenten Zustand versetzt [61]. Sie können auch dabei helfen, die Zuverlässigkeit der Registry zu maximieren. .LOG-Dateien werden auch verwendet, wenn aufgrund von Sperrungen oder Beschädigungen nicht direkt in die Registry-Hives geschrieben werden kann [62].

In der folgenden Abbildung 16 befindet sich die erstellte .tkape-Datei für die Registry-Hives. Diese umfasst nun alle Hives, in welchen später in den Modulen nach weiteren Artefakten gesucht werden soll.

ICS_Hives - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
Description: ICS Hives BA-CH
Author: CH
Version: 1.0
Id: ICST002-BA-CH
RecreateDirectories: true
Targets:
    -
        Name: NTUSER.DAT registry hive
        Category: Registry
        Path: C:\Users\%user%\
        FileMask: NTUSER.DAT
    -
        Name: NTUSER.DAT registry transaction files
        Category: Registry
        Path: C:\Users\%user%\
        FileMask: NTUSER.DAT.LOG*
    -
        Name: UsrClass.dat registry hive
        Category: Registry
        Path: C:\Users\%user%\AppData\Local\Microsoft\Windows\
        FileMask: UsrClass.dat
    -
        Name: UsrClass.dat registry transaction files
        Category: Registry
        Path: C:\Users\%user%\AppData\Local\Microsoft\Windows\
        FileMask: UsrClass.dat.LOG*
    -
        Name: SYSTEM registry hive
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: SYSTEM
    -
        Name: SYSTEM registry transaction files
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: SYSTEM.LOG*
    -
        Name: SOFTWARE registry hive
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: SOFTWARE
    -
        Name: SOFTWARE registry transaction files
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: SOFTWARE.LOG*
    -
        Name: SAM registry hive
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: SAM
    -
        Name: SAM registry transaction files
        Category: Registry
        Path: C:\Windows\System32\config\
        FileMask: SAM.LOG*
```

Abbildung 16 - ICS_Hives.tkape

3.3.3 Eventlogs

Da sich die Eventlogs alle unter dem Pfad `C:\Windows\System32\winevt\Logs` befinden, wird bei der Erstellung der Target-Dateien hauptsächlich mit der „FileMask“ gearbeitet. Dort wird spezifisch nach den Namen der jeweiligen Eventlogs gesucht. Da ein Target je Eventlog unwirtschaftlich wäre, wurden die Eventlog-Targets bereits nach Szenario zusammengefasst. Dafür wurde die Tabelle 3 genutzt, um die jeweiligen Eventlogs auszumachen und diese in einer .tkape-Datei zu vereinen. Beispielhaft ist die ICS_DT_Logfiles.tkape in Abbildung 17 zu sehen, welche die Eventlogs für die Analyse eines Datendiebstahls enthält.

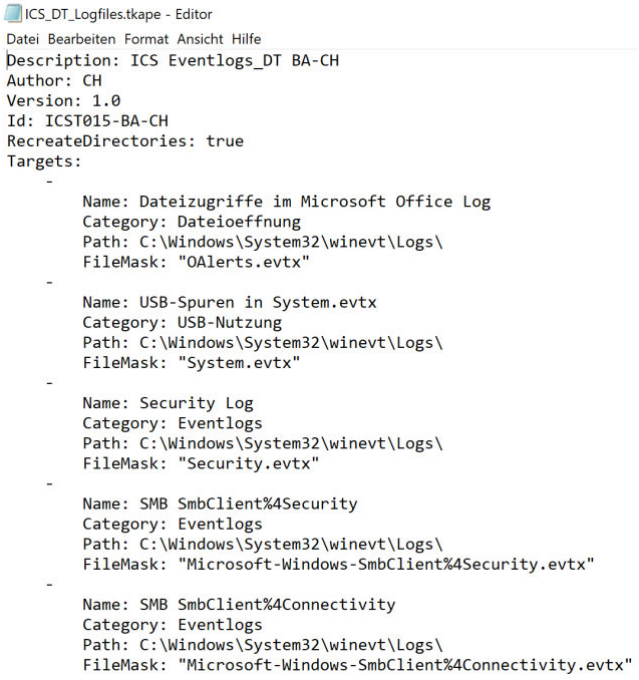


Abbildung 17 - ICS_DT_Logfiles.tkape

Die ICS_MW_Logfiles.tkape Datei enthält dabei die Eventlogs zum Szenario Malware und die ICS_IO_Logfiles.tkape enthält entsprechende Eventlogs zum Szenario Angriff von außen. Dabei steht IO für „Intrusion from Outside“, sprich Angriff von außen.

3.4 Target-Compound nach Szenario

Unter der Nutzung der Tabelle 3 werden nun die einzelnen erstellten Targets zu jeweils einem Compound für die jeweiligen Szenarien zusammengefasst. Dazu wird die Compound-Target-Vorlage verwendet. Die Compound-Targets werden nach dem Schema „ICS_%Szenario%.tkape“ benannt und jeweils unter dem Verzeichnis `KAPE\Targets\ICS_Compound\` gespeichert. Die Beschreibung unterscheidet sich nur im Hinblick der „Id“ von der eines Targets, denn statt ICSTXXX-BA-CH wird die „Id“ mit ICST**C**XXX-BA-CH dargestellt.

Weiterhin ist ein Name der Target-Datei sowie die zugehörige Kategorie erforderlich. Dabei werden für die Compound-Targets die Speicherorte aus der Tabelle 1 als Kategorie genutzt (z.B. Registry, Eventlogs). So ist ersichtlich, aus welchem der vorangehenden Kapiteln das Artefakt stammt. Als Pfad wird hier kein absoluter Pfad angegeben, sondern ein relativer Pfad, nämlich der zu einer anderen .tkape-Datei. Das kann sowohl eine weitere Compound-Datei sein als auch eine einfache Target-Datei mit absoluten Pfaden.

In der Abbildung 18 ist beispielhaft die Target-Compound-Datei für das Szenario Datendiebstahl dargestellt. Dafür werden die erstellten Targets zu einem Compound-Target je Szenario zusammengeführt. Die Entscheidung über die gewählten Artefakte je Szenario wurde im Kapitel 3.1 getroffen und ist zusätzlich unter Tabelle 3 einsehbar.

ICS_DATATHEFT.tkape - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
Description: ICS Datendiebstahl Compound BA-CH
Author: CH
Version: 1.0
Id: ICSTC002-BA-CH
RecreateDirectories: true
Targets:
    -
        Name: Registry-Hives
        Category: Registry
        Path: ICS_Hives.tkape
    -
        Name: Logfiles bzgl. Datendiebstahl
        Category: Eventlogs
        Path: ICS_DT_Logfiles.tkape
    -
        Name: $MFT
        Category: Dateisystem
        Path: ICS_$MFT.tkape
    -
        Name: LNK-Files
        Category: Dateisystem
        Path: ICS_LNK.tkape
    -
        Name: Thumbcache
        Category: Dateisystem
        Path: ICS_Thumbcache.tkape
    -
        Name: Recycle Bin
        Category: Dateisystem
        Path: ICS_RecycleBin.tkape
    -
        Name: Browserhistory Compound
        Category: Dateisystem
        Path: ICS_Browserhistory_Compound.tkape
    -
        Name: Jumplists
        Category: Dateisystem
        Path: ICS_Jumplist.tkape
    -
        Name: Amcache
        Category: Dateisystem
        Path: ICS_Amcache.tkape
```

Abbildung 18 - ICS_DATATHEFT.tkape

Mit identischem Aufbau werden so auch die zwei weiteren Szenarien erstellt. Dafür werden ebenfalls ausschließlich die selbst erstellten Targets unter 3.3 genutzt. Für die Szenarien Malware unter dem Target Compound „ICS_MALWARE.tkape“ und Angriff von außen unter dem Target Compound „ICS_INTRUSION_OUTSIDE.tkape“ sind dies folgende Targets:

- ICS_Hives.tkape
- ICS_MW_Logfiles.tkape oder ICS_IO_Logfiles.tkape
- ICS_$MFT
- ICS_LNK
- ICS_RecycleBin
- ICS_Browserhistory_Compound
- ICS_Outlook_PSTOST
- ICS_Prefetch
- ICS_Amcache

Bis auf die Eventlogs werden dabei identische Targets verwendet. Trotzdem werden sich die Artefakte im Endeffekt zwischen den zwei Szenarien noch merklich unterscheiden, da bei der Arbeit mit den Modulen im folgenden Kapitel aus den Registry-Hives unterschiedliche Artefakte geparst und analysiert werden sollen.

3.5 Erstellung der Module

Nachdem nun die Targets erstellt und nach den gewählten Szenarien in die jeweiligen Compound-Targets zusammengefasst wurden, werden nun die einzelnen Module erstellt. Diese verarbeiten mittels verschiedenen Binaries (ausführbare Dateien), die kopierten Dateien. In der Tabelle 3 werden die jeweiligen Programme zu den Artefakten gelistet, welche für das Parsen zuständig sind. Die mitgelieferte ModuleTemplate.template Datei dient dabei zur Orientierung. Auch diese wurde spezifisch angepasst, wie unter Abbildung 19 sichtbar ist.

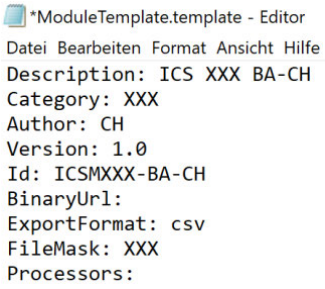


Abbildung 19 - Individualisiertes Module Template

Die Benennung der Module erfolgt dabei nach dem Schema „ICS_%Toolname%.mkape“ und deren Speicherung erfolgt unter dem Pfad `KAPE\Modules\ICS\`. Die Platzhalter XXX werden in der Beschreibung durch das jeweilige Tool ersetzt und in der „Id“ durch eine inkrementierende Nummer. Dabei wird die „Id“ äquivalent zu den Targets mit ICS**M**XXX-BA-CH angegeben. Auch die Kategorie und die „FileMask“ gehören bei den Modulen noch zu der Beschreibung. Für die „FileMask“ wird dabei, wenn notwendig, jeweils die spezifische, zu parsende Datei angegeben. Die Kategorien orientieren sich an den Artefakt-Kategorien der Tabelle 1 und sind verantwortlich für die später kategorisierte Ausgabe der geparsten Dateien in verschiedenen übersichtlichen Unterverzeichnissen.

Als Export-Format wird standardmäßig der .csv-Dateityp gewählt. Schließlich ist es auch möglich unter „BinaryURL“ den verwendeten Link zu einer ausführbaren Datei zu gewährleisten. So kann dieser nachgeladen werden, falls dieser nicht standardmäßig in KAPE vorhanden ist.

Das Vorgehen bei der Erstellung eines sogenannten „Processors“ in einem Modul ist immer gleich. Als "Executable“ muss eine ausführbare Datei angegeben werden. Diese wird aus der Tabelle 3 unter „Verwendetes Tool“ für das jeweilige Artefakt bezogen. Bis auf die zwei Tools „browsinghistoryview.exe“, „thumbcache_viewer_cmd.exe“ sowie das Skript „Move-KAPEConsoleHost_History.ps1“ werden die Eric Zimmerman Tools (Tabelle 2) genutzt. Diese befinden sich bereits vorgeladen im Verzeichnis `KAPE \Modules\bin.`

Weiterhin wird unter „CommandoLine“ der jeweilige Befehl für das verwendete Tool angegeben. Dieser kann aus bereits bestehenden Modulen entnommen werden, ist aber auch in den jeweiligen Dokumentationen der Tools vorhanden. Im Falle der Verwendung von Eric Zimmerman Tools kann unter Aufruf des jeweiligen Tools in der Kommandozeile eine Dokumentation eingesehen werden. Das Exportformat ist wie bereits erwähnt auch hier CSV, um schlussendlich eine Excel-Übersicht erstellen zu können. In der Abbildung 20 ist beispielsweise das erstellte Modul für das Artefakt „$MFT“ zu sehen.

ICS_MFTECmd.mkape - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
Description: MFTECmd
Category: Systemanalyse
Author: CH
Version: 1.0
Id: ICSM001-BA-CH
BinaryUrl: https://f001.backblazeb2.com/file/EricZimmermanTools/MFTECmd.zip
ExportFormat: csv
FileMask: $MFT
Processors:
    -
        Executable: MFTECmd.exe
        CommandLine: -f %sourceFile% --csv %destinationDirectory%
        ExportFormat: csv
```

Abbildung 20 - ICS_MFTECmd.mkape

Dabei ist ersichtlich, dass der Befehl `-f %sourceFile% --csv %destinationDirectory%` genutzt wird. Es wird also mit Platzhaltern gearbeitet, welche KAPE zu Laufzeiten mit den eigentlichen Variablen ersetzt [32]. Weiterführend ist unter „BinaryURL“ der Link zum entsprechenden Tool vermerkt. Mit der „FileMask“ wird explizit nach der Datei mit dem Namen §MFT gesucht und als Kategorie wird die Systemanalyse gewählt.

3.5.1 Dateisystem

Die zu parsenden Artefakte aus dem Speicherort „Dateisystem“ werden mit dem entsprechenden Tool aus der Tabelle 3 geparst. Dabei werden folgende Artefakte unter

dem gleichen Prinzip von verschiedenen Parsern aus der Eric Zimmerman Toolsammlung verarbeitet:

- $MFT (MFTEcmd)
- Shellbags (SBECmd)
- LNK-Files (LECmd)
- Recycle Bin (RBCmd)
- Prefetch (PECmd)
- Jumplists (JLECmd)
- AppCompatCache (AppCompatCacheParser)
- Amcache (AmCacheParser)
- SRUM (SRUMECmd)

Eine beispielhafte .mkape-Konfigurationsdatei ist in Abbildung 20 sichtbar. Alle Binaries für diese Module befinden sich bereits in KAPE unter dem Pfad `\KAPE\Modules\bin\EZTools\`. Der Aufbau dieser Konfigurationsdateien ist identisch und es ändert sich lediglich das „Executable“ und die „CommandLine“. Für die Erstellung der passenden Befehle werden die unter Kapitel 2.5 vorgestellten Befehlsoptionen verwendet. Zur Erstellung der Module werden unter anderem Vorlagen von Eric Zimmerman, Barrie Hill und Andrew Rathbun herangezogen. Dabei wird zum Teil lediglich die Benennung des Moduls sowie die „Id“ geändert und eine Anpassung der Kategorie vorgenommen.

Thumbcache

Das Artefakt „Thumbcache“ wird durch ein vorinstalliertes Modul von Dennis Reneau geparst. Dies befindet sich unter dem Verzeichnis `\KAPE\Modules\Apps\GitHub`. Auch hier wird, wie bereits erwähnt, die „Id“ und die Kategorie angepasst. Zusätzlich muss aber noch das passende Binary unter „BinaryURL“ auf GitHub heruntergeladen werden. Die erstellte mkape-Datei ist unter Abbildung 21 ersichtlich.

ICS_ThumbCacheViewer.mkape - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
Description: ThumbCacheViewer CMD
Category: Dateioeffnung
Author: CH (Dennis Reneau)
Version: 1.0
Id: ICSM004-BA-CH
BinaryUrl: https://github.com/thumbcacheviewer/thumbcacheviewer/releases/download/v1.0.1.7/thumbcache_viewer_cmd.zip
ExportFormat: csv
FileMask: "*.db"
Processors:
    -
        Executable: thumbcache_viewer_cmd.exe
        CommandLine: -o %destinationDirectory%\ThumbCache_Results -w -c -z %sourceFile%
        ExportFormat: html
        ExportFile: thumbcache_results.csv
```

Abbildung 21 - ICS_ThumbCacheViewer.mkape

Der in „CommandLine“ ausgeführte Befehl wird folgend genauer aufgelöst. Dabei steht der Parameter `-o` für das Ausgabe-Verzeichnis, die Parameter `-c` und `-w` für das Generieren eines CSV- sowie HTML-Reports und der Parameter `-z` dient dazu, Dateien

mit einer Größe von 0 Byte auszuschließen. Auch hier werden Platzhalter für den jeweiligen Ziel-Pfad (%destinationDirectory%) und die Quelldatei (%sourceFile%) verwendet [63].

Webzugriffe

Auch das Artefakt „Webzugriffe“ wird durch ein externes Tool geparst. Dies stammt von NirSoft und wird ebenfalls unter der vorhandenen URL in „BinaryURL“ nachgeladen. Für die Erstellung des Moduls wird ein vorgefertigtes Modul von Mike Cary abgeändert. Dies befindet sich unter `\KAPE\Modules\Apps\NirSoft`. In Abbildung 22 ist die neue Version zu sehen. Dabei ist zu beachten, dass die Kommando-Zeile nur für die Verbildlichung auf 2 Zeilen geteilt wurde und es durch diese Syntax zu einer Fehlermeldung kommen würde.

Abbildung 22 - ICS_NirSoft_BrowsingHistoryView.mkape

Für den Befehl unter „CommandLine“ werden mehrere Parameter verwendet. `/HistorySource 3` gibt dabei an, dass die Browser-Historie von einem spezifischen Ordner geladen werden soll. Dieser ist folgend unter `HistorySourceFolder` mit einem Platzhalter in Form von „%sourceDirectory%\Users“ angegeben. Der Parameter `/VisitTimeFilterType 1` legt fest, dass die gesamte Historie ohne zeitliche Einschränkung geladen werden soll. Der Befehl `\scomma` mit anschließendem Ziel Pfad sorgt für die Ausgabe in Form einer CSV-Datei. [64]

Powershell ConsoleHistory

Das Artefakt „Powershell ConsoleHistory“ stellt beim Parsen eine Ausnahme dar. Die durch das Target „ICS_PS_ConsoleHistory.tkape“ gesicherte Datei liegt bereits nach dem Kopiervorgang im Verzeichnis `KAPE\output\tout` als menschenlesbare .txt-Datei vor. Damit diese aber sauber in der Ausgabe der Module zu finden ist, und nicht in der Ausgabe der Targets untergeht, gibt es ein Powershell-Skript. Dies schiebt die .txt-Datei im Wesentlichen einfach nur weiter in die Modul-Ausgabe und wurde von Andrew Rathbun erstellt. Das dafür notwendige Skript wurde unter der verfügbaren „BinaryURL“ nachgeladen [65]. Zusätzlich ist zu beachten, dass sich das Binary „PowerShell.exe“ auf dem Computer befinden muss. Wie in Abbildung 23 zu sehen ist, wurde auch von dieser

.mkape-Konfigurationsdatei eine neue Version erstellt, um dem gewählten Schema zu entsprechen.

Abbildung 23 - ICS_PS_Move_ConsoleHost_history.mkape

E-Mail-Artefakte

Abschließend bleiben noch die E-Mail-Artefakte übrig. Diese werden nicht geparst, sondern nur durch die Targets kopiert und befinden sich im Verzeichnis `KAPE\output\tout`. Von dort aus können sie durch spezifische Programme eingesehen werden. Die Entscheidung, diese Artefakte nicht weiter zu parsen, resultiert daraus, dass durch geeignete Tools eine weitaus übersichtlichere Analyse erfolgen kann als mit einer CSV-Datei. Zu diesen Tools gehört der Kernel-PSTOSTViewer. Durch die bessere Übersicht ist es möglich, schneller nachzuvollziehen, in welchem Ordner sich die E-Mails befinden, welche Kommunikationen stattfanden und selbst Anhänge sind sauber dargestellt.

Somit wurden folgende .mkape-Dateien erstellt, welche im Nachhinein beliebig zu einem Modul-Compound zusammengeführt werden können:

- ICS_MFTECmd.mkape
- ICS_SBECmd.mkape
- ICS_RBCmd.mkape
- ICS_PECmd.mkape
- ICS_JLECmd.mkape
- ICS_PS_Move_ConsoleHost_history.mkape
- ICS_AppCompatCacheParser.mkape
- ICS_AmcacheParser.mkape
- ICS_LECmd.mkape
- ICS_SRUMECmd.mkape
- ICS_ThumbCacheViewer.mkape
- ICS_NirSoft_BrowsingHistoryView.mkape

Nach der Erstellung dieser Module wird zusätzlich noch überprüft, ob für einige der Module noch Binaries fehlen. Dies wird durch den Befehl `kape.exe --mlist . --mdetail` umgesetzt. Dafür wird zunächst eine Eingabeaufforderung (CMD) mit Administratorrechten gestartet, da diese für KAPE notwendig sind. Nach dem Navigieren zum entsprechenden Verzeichnis, in welchem KAPE vorhanden ist, wird

der Befehl ausgeführt. Dabei konnten keine fehlenden Binaries festgestellt werden, welche zusätzlich benötigt werden.

3.5.2 Registry

Alle aus der Registry stammenden Artefakte werden mit dem Tool RECmd geparst. Die jeweiligen Artefakte werden aus der Tabelle 3 entnommen. Für das Tool RECmd existiert ein Batch-Mode, welcher die Daten aus der Registry mit Hilfe von Plugins in eine standardisierte CSV-Ausgabe exportiert [66]. So soll jeweils eine Batch-Datei für die drei Szenarien erstellt werden, welche dann zur Erstellung der eigentlichen Module verwendet werden kann. Dort werden alle für den Fall relevanten Artefakte der Registry zusammengefasst und können jederzeit ergänzt oder verändert werden. Dies läuft nicht über KAPE, sondern über das Tool RECmd selbst. Die geschriebenen Batch-Dateien werden dann im Verzeichnis `KAPE\bin\RECmd\BatchExamples` abgelegt. Dort existiert auch eine RECmdBatch.template (Abbildung 24), welche als Vorlage genutzt wird. Die Benennung der Batchfiles erfolgt unter dem Muster „ICS_%Szenario%.reb“ und die jeweilige „Id“ ist ICSRXXX und wird bei jeder Datei um einen Zähler inkrementiert.

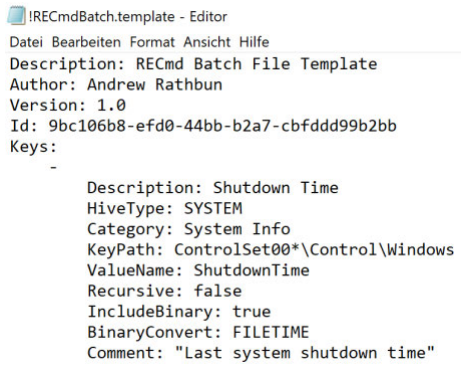


Abbildung 24 - RECmdBatch.template

Die Batch-Datei kann in zwei Teile unterteilt werden. Der Header beinhaltet die Beschreibung sowie den Autor, eine Version und die ID. Der Key-Teil beinhaltet eine Beschreibung, welche dann auch in der Ausgabe erscheint, sowie den Hive-Typ, den Pfad, und optional ein „ValueName“, welcher vergeben werden kann. Ebenfalls erforderlich ist die Option „Recurisve“, welche darüber entscheidet, ob der gewählte Pfad rekursiv durchsucht werden soll. [66]

Das Vorgehen bei der Erstellung einer .reb-Datei liegt sehr nahe an der bereits bekannten Vorgehensweise der Erstellung von Targets und Modulen. Unter Zuhilfenahme der Tabelle 3 werden die für die Szenarien notwendigen Artefakte, welche mit dem Tool RECmd geparst werden sollen, gewählt und mit den entsprechenden Optionen in die Datei geschrieben. Neben den drei Szenarien wird zusätzlich für die Systemanalyse eine Batch-Datei erstellt und anschließend in die Module der Szenarien integriert, da diese

Artefakte in allen Szenarien relevant sind. Diese .reb-Datei ist beispielhalft in Abbildung 25 abgebildet.

ICS_Systemanalyse.reb - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
Description: RECmd Systemanalyse
Author: CH
Version: 1.0
Id: ICSR001-BA-CH
Keys:
    -
        Description: CurrentVersion
        HiveType: SOFTWARE
        Category: Systemanalyse
        KeyPath: Microsoft\Windows NT\CurrentVersion\
        ValueName: CurrentBuild
        Recursive: true
        Comment: "BS Version"
    -
        Description: CurrentVersion
        HiveType: SOFTWARE
        Category: Systemanalyse
        KeyPath: Microsoft\Windows NT\CurrentVersion\
        ValueName: ProductName
        Recursive: false
        Comment: "BS Version"
    -
        Description: ComputerName
        HiveType: SYSTEM
        Category: Systemanalyse
        KeyPath: ControlSet*\Control\computername\ComputerName\
        ValueName: ComputerName
        Recursive: true
        Comment: "Computername"
    -
        Description: TimeZoneInformation
        HiveType: SYSTEM
        Category: Systemanalyse
        KeyPath: ControlSet*\Control\TimeZoneInformation\TimeZoneKeyName
        ValueName: TimeZoneKeyName
        Recursive: false
        Comment: "Systemzeit"
    -
        Description: SAM
        HiveType: SAM
        Category: Systemanalyse
        KeyPath: Domains\Account\Users\User Accounts\
        Recursive: true
        Comment: "Benutzerkonten"
    -
        Description: ProfileList
        HiveType: SOFTWARE
        Category: Systemanalyse
        KeyPath: Microsoft\Windows NT\CurrentVersion\ProfileList
        Recursive: false
        Comment: "Benutzerkonten"
```

Abbildung 25 - ICS_Systemanalyse.reb

Zu sehen sind die jeweiligen Artefakte „Current Version“, „ComputerName“, „TimeZoneInformation“ und „Users“. Der jeweilige verwendete Registry-Hive wird unter „HiveType“ angegeben. Unter „KeyPath“ wird der genaue Pfad innerhalb der Registry angegeben, welcher unter Tabelle 1 für jedes Artefakt einsehbar ist. Die Kategorie wird nach der Kategorisierung der beiden Tabelle vergeben und ist später in die CSV-Ausgabe integriert. Teilweise wird ein genauer „ValueName“ angegeben, um nur einen bestimmten Wert aus den Registry-Keys zu erhalten. Dies wird getan, um eine bessere Übersicht zu behalten. Außerdem reduziert die ausschließliche Speicherung bestimmter Werte die Dateigröße sowie die Zeit, die zum Parsen benötigt wird. Letztendlich wird noch für jeden Eintrag mittels „Recurisve“ entschieden, ob die jeweiligen Subkeys und dazugehörigen Values mitgeladen werden sollen oder nicht. Dieses Vorgehen wird für die drei Szenarien wiederholt, um folgend die Batch-Dateien ICS_IO.reb, ICS_Malware.reb und ICS_Datendiebstahl.reb im Verzeichnis `\BatchExamples` zu erstellen.

Für die jeweiligen Batch-Dateien wird anschließend jeweils ein Modul pro Szenario für die Verwendung in KAPE verfasst, welches den Befehl inklusive der jeweiligen Batch-Datei für die Ausführung des Moduls beinhaltet. Dabei wird in der „Command“-Option auf diese zugegriffen. Für die Systemanalyse ist die .mkape-Datei in Abbildung 26 zu sehen.

Abbildung 26 - ICS_RECmd_Systemanalyse.mkape

Unter „CommandLine“ ist dann die zuvor unter RECmd erstellte Datei ICS_Systemanalyse.reb ersichtlich, auf welche durch den Befehl `--bn` zugegriffen wird. Die anderen drei Module für die jeweiligen Szenarien sind identisch aufgebaut, beinhalten jeweils nur die entsprechende Batch-Datei für das jeweilige Szenario und erhalten folgende Namen:

- ICS_RECmd_DT.mkape (mit ICS_Datendiebstahl.reb)
- ICS_RECmd_MW.mkape (mit ICS_Malware.reb)
- ICS_RECmd_IO.mkape (mit ICS_IO.reb)

3.5.3 Eventlogs

Die Eventlog-Module werden zuerst einzeln verfasst, um diese dann zu einem Compound-Modul zusammen zu fassen, welches dem jeweiligen Szenario gilt. Die jeweiligen relevanten Artefakte werden der Tabelle 3 entnommen und die zugehörigen Eventlog-Namen sowie die relevanten Event-IDs befinden sich in der Tabelle 1. Die Eventlogs werden dafür mit dem Tool EvtxECmd von Eric Zimmerman geparst. Mit diesem Tool ist es auch möglich, nur relevante Event-IDs für die jeweilige Untersuchung zu inkludieren. Dies geschieht unter dem Kommando `--inc „EventID1,EventID2,EventID3“` [67]. In Abbildung 27 ist beispielhaft die .mkape-Datei für das Eventlog Security.evtx ersichtlich.

Abbildung 27 - ICS_EtvtxECmd_Security.mkape

Die Kategorie bleibt für jede .mkape-Datei identisch und entspricht dem Tag „Eventlogs“. Später werden alle CSV-Dateien in Bezug auf diese Kategorie unter einem gleichnamigen Verzeichnis auffindbar sein. Für die weiteren Module ändert sich somit lediglich die „FileMask“, welche dem Namen des jeweiligen Eventlogs entspricht und die jeweils relevanten Event-IDs, welche in die Untersuchung miteinbezogen werden sollen. Dies reduziert die Zeit für das Parsen und die Größe der Datei auf die relevantesten Einträge. Man kann daher von einer Art der Vorfilterung sprechen. Insgesamt wurden folgende Module erstellt:

- ICS_EvtxECmd_Security.mkape
- ICS_EvtxECmd_System.mkape
- ICS_EvtxECmd_RDPClient.mkape
- ICS_EvtxECmd_OAlerts.mkape
- ICS_EvtxECmd_LocalSessionManager.mkape
- ICS_EvtxECmd_RdpCoreTS.mkape
- ICS_EvtxECmd_RemoteConnectionManager.mkape
- ICS_EvtxECmd_SmbClient%4Security.mkape
- ICS_EvtxECmd_SmbClient%4Connectivity.mkape
- ICS_EvtxECmd_SMBServer%4Operational.mkape
- ICS_EvtxECmd_SMBServer%4Security.mkape
- ICS_EvtxECmd_WMI-Activity.mkape
- ICS_EvtxECmd_Powershell%4Operational.mkape
- ICS_EvtxECmd_Powershell.mkape
- ICS_EvtxECmd_WinRM.mkape

Diese erstellten .mkape-Dateien werden nun entsprechend der Szenarien noch zu Compounds zusammengefasst und unter `KAPE\Modules\ICS_Compound` abgelegt. Somit muss im gesamten Modul-Compound für das jeweilige Szenario nur noch das entsprechende EvtxECmd-Compound angewählt werden und nicht jedes Eventlog einzeln. In Abbildung 28 ist beispielsweise das Compound-Modul bezüglich des Szenario Datendiebstahls zu sehen. Das Namensschema entspricht dabei „ICS_EvtxECmd_%Szenario%.mkape“.

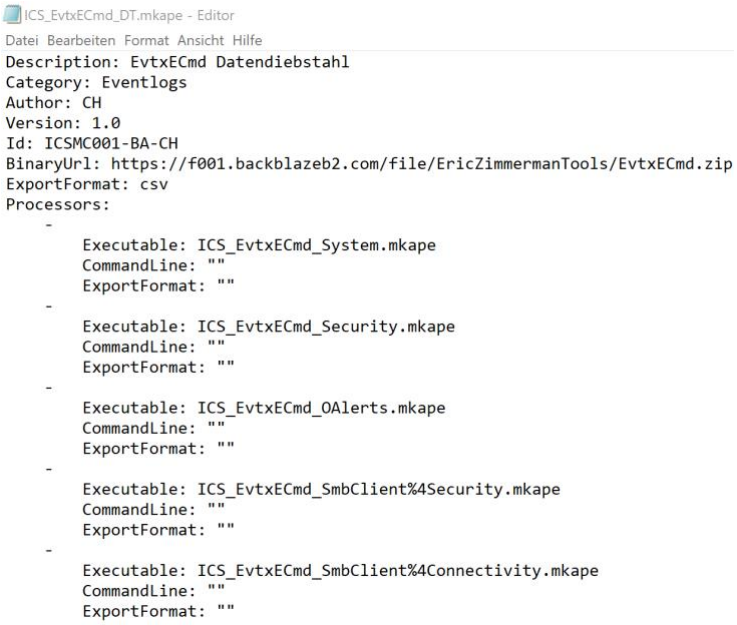


Abbildung 28 - ICS_EvtxECmd_DT.mkape

Für die weiteren zwei Szenarien ist die Zusammensetzung der Eventlogs der Tabelle 3 zu entnehmen. Die .mkape-Dateien sind entsprechend des Beispiels aufgebaut und tragen die Namen „EvtxECmd_MW.mkape für das Szenario Malware und entsprechend „EvtxECmd_IO.mkape“ für den Angriff von außen.

3.6 Modul-Compound nach Szenario

In diesem Abschnitt werden die vorher erstellten Module der jeweiligen Speicherorte zu einem Compound-Modul für das jeweiligen Szenario zusammengefasst. Dafür gibt die Tabelle 3 eine Orientierung. Unter Verwendung der Compound Modul-Vorlage werden die jeweiligen .mkape-Dateien nach dem Schema „ICS_%Szenario%.mkape“ benannt und im Verzeichnis `KAPE\Modules\ICS_Compound\` gespeichert. Die Beschreibung unterscheidet sich nur im Hinblick auf die „Id“ eines Moduls und wird mit ICSM**C**XXX-BA-CH dargestellt. Folgend ist in Abbildung 29 die Datei ICS_DATATHEFT.mkape dargestellt.

ICS_DATATHEFT.mkape - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
Description: ICS Datendiebstahl Compound
Category: Datendiebstahl (Multiple)
Author: CH
Version: 1.0
Id: ICSMC004-BA-CH
ExportFormat: csv
Processors:
    -
        Executable: ICS_EvtxECmd_DT.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_RECmd_Systemanalyse.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_RECmd_DT.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_MFTECmd.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_SBECmd.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_LECmd.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_ThumbCacheViewer.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_RBCmd.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_NirSoft_BrowsingHistoryView.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_JLECmd.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_AppCompatCacheParser.mkape
        CommandLine: ""
        ExportFormat: ""
    -
        Executable: ICS_AmcacheParser.mkape
        CommandLine: ""
        ExportFormat: ""
```

Abbildung 29 - ICS_DATATHEFT.mkape

Als Kategorie wird dabei jeweils der Term „Multiple“ angegeben, da Module mehrerer Kategorien als auch mehrerer Speicherorte zusammengefasst werden. Das „ExportFormat“ bleibt, wie bereits erwähnt, bei CSV. Für die „Processors“ werden jeweils unter „Executable“ die erforderlichen .mkape-Dateien eingetragen. Neben den spezifisch erstellten EvtxECmd- und RECmd-Modulen werden die erforderlichen Module des Speicherorts Dateisystem miteingefügt.

Die beiden anderen Modul-Compounds ICS_MALWARE.mkape und ICS_INTRUSION_OUTSIDE.mkape unterscheiden sich nur in wenigen Punkten. Für das Szenario Malware werden dabei folgende Module benutzt:

- ICS_EvtXECmd_MW
- ICS_RECmd_MW
- ICS_MFTEcmd
- ICS_SBECmd

- ICS_LECmd
- ICS_RBCmd
- ICS_NirSoft_BrowsingHistoryView
- ICS_PECmd
- ICS_JLECmd
- ICS_PS_Move_ConsoleHost_history
- ICS_AppCompatCacheParser
- ICS_AmcacheParser

Logischerweise werden für das Szenario Angriff von außen die jeweiligen EvtxECmd- und RECmd-Module benutzt. Der einzige weitere Unterschied der beiden Compound-Targets ist die Verwendung des SBECmd bei dem Malware-Compound, denn dieses wird beim Angriff von außen nicht benötigt. Die restlichen Module sind identisch.

4 Evaluierung und Ergebnisse

In diesem Kapitel wird ein beispielhafter Überblick über das Vorgehen der Evaluierung der erstellten Targets und Module gegeben. Darüber hinaus werden die entstandenen Ergebnisse präsentiert. So wird zum einen festgestellt, wie schnell das Tool ist, um einen Vergleich mit der „händischen Arbeit“ durch die Eric Zimmerman Tools zu ziehen. Zum anderen wird damit evaluiert, ob die forensischen Prinzipien hinreichend eingehalten werden und die Artefakte entsprechend dargestellt werden. Die Ergebnisse dessen werden dann im folgenden Kapitel 5 diskutiert und sollen die Leitfrage beantworten, ob unter der Nutzung von KAPE die Automatisierung eines Incident Response schneller und effektiver gestaltet werden kann, ohne die forensischen Prinzipien außer acht zu lassen.

4.1 Vorgehen

Um eine Aussagekräftige Evaluierung zu erreichen, werden in einer allgemeinen Evaluierung zunächst alle erstellten Targets und Module in einem Test-Compound mit dem Namen ICS_Test.tkape und ICS_Test.mkape zusammengeführt und mit einem Test-Image durch die Anwendung `gkape.exe` überprüft. Das zur Verfügung stehende Datenträgerabbild stammt dabei aus dem Repertoire der Plattform `cfred.nist.gov` [68]. Auf dieser Webseite stehen verschiedene Datenträgerabbilder zur Übung und zur Testung forensischer Tools zur Verfügung. Bei dem gewählten Image handelt es sich um das „Capture the Flag“-Event aus dem Jahre 2019 von Magnet Forensics.

Die entstehende Ausgabe unter `KAPE\output\tout` für die Targets wird auf seine Vollständigkeit geprüft. Dazu wird unter Zuhilfenahme der Tabelle 1 und 3 geprüft, ob alle Artefakte kopiert wurden. Zusätzlich werden die von KAPE erstellten CSV-Dateien Copylog und Skiplog geprüft, um festzustellen, welche Dateien tatsächlich kopiert und welche übersprungen wurden. Für die Modul-Ausgabe unter `KAPE\output\mout` wird geprüft, ob die jeweiligen Module die kopierten Daten parsen konnten und die CSV-Dateien vollständig und übersichtlich sind, sprich nur die gewünschten Inhalte in Form einer Triage enthalten. Zusätzlich wird das ConsoleLog.txt geprüft, um festzustellen, ob es Fehlermeldungen oder Optimierungsmöglichkeiten in einem der Module gibt.

Anschließend wird für jedes Szenario einzeln die Funktionsweise und Geschwindigkeit evaluiert. Dazu wird jedes Szenario mit entsprechenden Target- und Modul-Dateien durch das gewählte Datenträgerabbild überprüft. Schlussendlich wird neben der zeitlichen Dauer geprüft, ob die jeweiligen Szenarien und deren Ausgabe die notwendigen Artefakte aus Tabelle 3 enthalten und damit soll eine Einschätzung darüber möglich werden, ob die Ausgabe bei einem Incident Response optimal gestaltet ist.

4.2 Allgemeine Evaluierung

Zunächst wird das Datenträgerabbild unter folgendem Link heruntergeladen: https://cfreds.nist.gov/all/MagnetForensics/2019WindowsMagnetCTF (Zugriff 05.09.2022). Anschließend wird die heruntergeladene .E01-Datei mit dem Arsenal Image Mounter [69] unter der Option „WriteTemporary“ als Laufwerk D:\ eingebunden. Die Option „WriteTemporary“ wird gewählt, um temporär in einer .diff-Datei Veränderungen am Datenträgerabbild vorzunehmen, ohne Veränderungen an der Originaldatei zu erzeugen. Anschließend wird KAPE mit der grafischen Nutzeroberfläche gestartet (`gkape.exe`) und die entsprechenden Optionen werden angewählt. Als „TargetDestination“ wird dabei das Laufwerk D:\ ausgewählt. Für die erforderlichen Felder „TargetOutput“ und „ModuleOutput“ werden die vorbereiteten Verzeichnisse `\tout` und `\mout` gewählt. Des Weiteren wird jeweils die Option „flush“ angewählt, welche die gewählten Ausgabe-Verzeichnisse entleert und neu aufsetzt. Schließlich werden die erstellten Test-Dateien ICS_Test.tkape und ICS_Test.mkape aus dem jeweiligen Target- und Modul-Reiter angewählt und die Konfiguration `--debug` gesetzt. Dadurch entsteht folgender Befehl in der „CommandLine“ des Tools:

```
.\kape.exe --tsource D: --tdest
C:\Users\DFIR\Desktop\BACH\KAPE\output\tout\ --flush --target
ICS_Test_Compound --mdest
C:\Users\DFIR\Desktop\BACH\KAPE\output\mout\ --mflush
ICS_Test_Compound --debug --gui
```

4.3 Ergebnisse der Allgemeinen Evaluierung

Folgend werden die erreichten Ergebnisse und Feststellungen präsentiert. Während das Tool läuft, kann an der Fortschrittsanzeige festgestellt werden, wie viel Prozent bereits abgeschlossen sind und wie viele Dateien kopiert oder übersprungen wurden. Zusätzlich können die Schritte, die KAPE ausführt, live in der Befehlszeile verfolgt werden. Nachdem das Tool durchgelaufen ist, kann folgender Inhalt der jeweiligen Ordner `\tout` (Abbildung 30) und `\mout` (Abbildung 31) festgestellt werden:

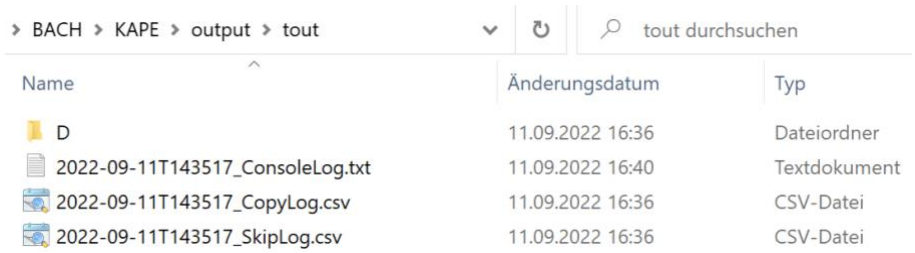


Abbildung 30 - Ausgabe Verzeichnis \tout

In diesem Ordner ist zu erkennen, dass sowohl ein ConsoleLog.txt, als auch jeweils ein _CopyLog.csv und ein _Skiplog.csv abgelegt wurden. Diese Dateien sind im Kontext der

forensischen Prinzipien sehr wertvoll. Wie in Kapitel 2.4.3 bereits beschrieben, sorgen diese Dateien für die Protokollierung der Geschehnisse während der Laufzeit von KAPE. Damit ist das forensische Prinzip der Dokumentation abgedeckt. Dazu gehören sowohl die Dokumentation aller ausgeführten Befehle inklusive eines Zeitstempels im ConsoleLog.txt, als auch eine Übersicht über die kopierten und übersprungenen Dateien, inklusive einer Begründung, warum diese Dateien übersprungen wurden. Für die forensischen Prinzipien der Glaubwürdigkeit und Wiederholbarkeit, welche ebenfalls im Kapitel 2.1.2 aufgegriffen wurden, sind diese Protokolldateien von hoher Bedeutung.

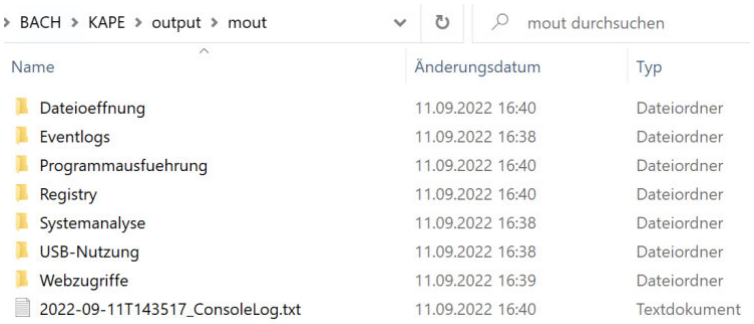


Abbildung 31 - Ausgabe Verzeichnis \mout

In Abbildung 31 ist zu erkennen, dass die in Kapitel 3.5 gewählten Kategorien der Module nun genutzt wurden, um die Ausgabe entsprechend zu gliedern. Diese ist nun der internen Berichtsvorlage angepasst und beinhaltet die entsprechend kategorisierten Artefakte so, wie sie auch im Bericht aufgeführt werden.

Der gesamte Vorgang für alle Targets und Module dauert dabei rund 249,5891 Sekunden, was gerundet etwa vier Minuten entspricht. Es ist dabei zu beachten, dass dies keiner Live-Sicherung entspricht, wie sie bei einem Incident-Response vorkommt und dass das Test-Image nicht so umfangreich ist, wie ein zu untersuchender Datenträger. Dennoch ist die Überprüfung der Targets und Module unter diesen Umständen sinnvoll, um festzustellen, inwiefern die Targets und Module in der praktischen Umsetzung funktionieren, um diese gegebenenfalls zu verbessern und anzupassen. Bei einer Live-Sicherung würde anstelle des Laufwerkbuchstabens des eingebundenen Images lediglich der Laufwerkbuchstabe des zu sichernden Systems angegeben. Die Ausgabe würde direkt auf den externen Datenträger, wo sich auch KAPE befindet, umgeleitet werden. So sollen unnötige Spuren auf dem zu analysierenden System verhindert werden.

4.3.1 Targets

Es wird nun überprüft, ob die Ausgabe der Targets vollständig ist. Dazu wird ein Vergleich zur Tabelle 1 vorgenommen und die Artefakte werden zu Zwecken der Übersichtlichkeit der Evaluierung wieder nach Speicherort gegliedert.

Registry

Die Artefakte aus dem Speicherort Registry werden in den Targets durch ein Compound von Registry-Hives abgebildet. Durch die Option „Recursive“ wurden die Verzeichnisstrukturen in allen Targets übernommen. Daher befinden sich die Registry-Hives `SYSTEM`, `SAM` und `SOFTWARE` unter dem Verzeichnis `\KAPE\output\tout\D\Windows\System32\config` während sich die weiteren Registry-Hives, wie die `NTUSER.dat` und `UsrClass.dat`, unter den jeweiligen Benutzer-Verzeichnissen befinden. Alle erforderlichen Registry-Hives sind vorhanden. Sie sind die Voraussetzung für das Parsen der Artefakte aus der Registry in den Modulen. Von hier aus besteht auch die Möglichkeit, diese Registry-Dateien in den Registry-Explorer von Eric Zimmerman zu laden, um die einzelnen weiteren Artefakte in der grafischen Nutzeroberfläche zu sehen.

Eventlogs

Unter den Eventlogs im Pfad `\KAPE\output\tout\D\Windows\System32\winevt\Logs` konnten lediglich zehn von den 15 erwarteten Logfiles festgestellt werden. Nach einer manuellen Überprüfung des Datenträgerabbildes war ersichtlich, dass einige Eventlogs auf dem Datenträgerabbild gar nicht vorhanden sind, jedoch aber auch mehrere Dateien von KAPE nicht kopiert wurden. Um den Ursprung dieses Fehlers zu finden, wurde zunächst das Target-Compound ICS_TEST.tkape auf Schreibfehler überprüft. Dort konnten kein Fehler gefunden werden, weshalb die eigentlichen Target-Dateien überprüft wurden. Dort konnte beispielsweise für das Eventlog Windows PowerShell.evtx ein Formfehler festgestellt werden, welcher korrigiert wurde. Es handelte sich dabei um einen Fehler in der Groß- und Kleinschreibung. Insgesamt ist die verwendete Strukturierung unter YAML an sehr strenge Vorgaben gebunden (Abbildung 32). Eine weitere Target-Datei der Eventlogs wies einen Fehler in der Formatierung auf.

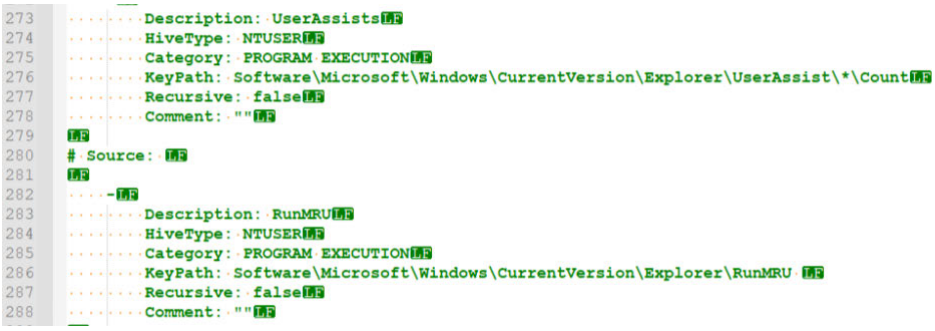


Abbildung 32 - Beispieldatei Formatierung

Wie in der Abbildung 32 zu sehen ist, bedarf es an genau vier bzw. acht Leerzeichen vor den jeweiligen Konfigurationseinstellungen. Zusätzlich muss jede Zeile mit „LF“ enden, was für einen Zeilenumbruch steht. Die Formatierungsfehler konnten ebenfalls behoben werden.

Dateisystem

Für den Speicherort Dateisystem konnten ebenfalls Abweichungen zwischen den vorgegebenen Artefakten in Tabelle 1 und der Ausgabe durch KAPE festgestellt werden. Auch in diesem Fall fand eine manuelle Überprüfung statt und es konnte bei einer händischen Überprüfung eruiert werden, dass das nicht kopierte Artefakt „RecycleBin“ tatsächlich nicht auf dem Datenträgerabbild vorhanden ist. Zusätzlich waren auch keine E-Mail-Artefakte vorhanden, welche dann entsprechend nicht kopiert werden konnten.

4.3.2 Module

Nach einer Verbesserung der Targets und dem Korrigieren der genannten Fehler wurde der Befehl aus dem Kapitel 4.2 erneut ausgeführt, um überhaupt eine vollständige Modul-Ausgabe zu ermöglichen. Nach erneuter Überprüfung der Targets wurde die Ausgabe der Module sodann mit Hilfe der Tabelle 1 auf ihre Vollständigkeit und Funktionalität überprüft. Die nicht vorhandenen Artefakte wie die „RecycleBin“, die E-Mail-Artefakte und die entsprechenden Eventlogs können natürlich auch nicht geparst werden.

Registry

Wie in Abbildung 31 zu sehen und bereits beschrieben, sind die Kategorien nach Untersuchungstyp der jeweiligen Artefakte geordnet. Dies trifft nicht auf die Registry zu, denn dieser Speicherort hat eine eigene Kategorie. Das resultiert aus den in Kapitel 3.5.2 erstellten Batch-Dateien des Tools RECmd. Die Modul-Datei greift nämlich nur auf die erstellten .reb-Dateien zu und führt diese aus.

In der Ausgabe unter `KAPE\output\mout\Registry` befindet sich neben einer Übersichtsdatei auch ein gleichnamiger Ordner. Die Übersichtsdatei enthält alle Registry-Artefakte des jeweiligen Szenarios gegliedert nach Kategorie und Artefakt Namen. Diese CSV-Datei dient zum groben Überblick über alle Artefakte, während sich im zusätzlichen, gleichnamigen Ordner detaillierte CSV-Dateien für einzelne Artefakte befinden. Benutzerbezogene Spuren tragen dabei zusätzlich den Namen des Benutzers im Dateinamen.

Es konnte für das Artefakte „TimeZoneInformation“ kein Weg gefunden werden, nur die „TimeZoneKeyName“ auszugeben, weshalb der gesamte Registry-Key mit geparst wird. Die Verwendung der Konfigurationsmöglichkeit „ValueName“ scheint dabei nicht anzuspringen. Dadurch wird mehr an Ausgabe erzeugt als notwendig. Da die Information über die Zeitzone aber sehr relevant sein kann, wird dies in Kauf genommen und das Modul bleibt so bestehen. Genauer wird die Registry-Ausgabe in der zweiten Evaluierung überprüft und beleuchtet.

Eventlogs

Unter dem Verzeichnis Eventlogs befinden sich alle geparsten CSV-Dateien zu den jeweiligen Eventlogs. Dabei werden jeweils nur die angegebenen Event-IDs in den Tabellen aufgeführt, was die Ausgabe um einiges übersichtlicher macht. Benannt werden die jeweiligen Dateien derzeit lediglich mit einem Zeitstempel. Dies ist nicht optimal, da so nicht erkennbar ist, welche Datei welchem Eventlog zuzuordnen ist. Daher soll jedes EvtxECmd-Modul zusätzlich mit der Option `--csvf` ergänzt werden. Dies ermöglicht es, einen festen Dateinamen vorzugeben, wobei jeweils einfach der Name des Eventlogs vergeben wird. So soll es möglich sein, schneller zu erkennen, um welche Datei es sich handelt.

Dateisystem

Auch bei den weiteren Modulen, bei welchen Parser von Eric Zimmerman verwendet werden, ist die Ausgabedatei nach dem jeweiligen Parser benannt. Diese sollen ebenfalls mittels der Option `--csvf` einen alternativen Namen bekommen, wofür der Artefakt-Name gewählt wird. Dies dient der Übersichtlichkeit und soll ein schnelleres Navigieren innerhalb des Ausgabeverzeichnisses ermöglichen. Das sieht beispielsweise für die LNK-Files unter der Kategorie Dateiöffnung folgendermaßen aus (Abbildung 33):

Abbildung 33 - Angepasste ICS_LECmd.tkape

Zu beachten ist, dass bei einer Veränderung auch jeweils die Versionsnummer inkrementiert wird.

Es konnte festgestellt werden, dass die Ausgabe der Browser-History unter dem Verzeichnis Webzugriffe leer ist. Zur Überprüfung wird daher mit einer Rückwärtssuche von der ICS_Test.mkape Datei über das Modul-Compound ICS_Browserhistory_Compound.mkape bis zu den einzelnen Browser-Modulen nach Fehlern geforscht. Der Fehler konnte nicht festgestellt werden und das Modul funktioniert zurzeit noch nicht. Die .mkape-Dateien weisen keine Schreib- oder Formatierungsfehler auf und es wurde sogar ein alternatives Datenträgerabbild getestet. Dennoch bleibt die geparste CSV-Datei leer. Es besteht aber die alternative Möglichkeit, die bereits durch die

Targets kopierten Dateien der Kategorie Webzugriffe manuell mit einem externen Programm auszuwerten.

4.4 Evaluierung nach Szenario

Um die erstellten Compound-Dateien der jeweiligen Szenarien zu prüfen und eine zeitliche Einordnung zu ermöglichen, werden im Folgenden die einzelnen Szenarien nach ersten genannten Verbesserungen mittels dem zur Verfügung stehenden Datenträgerabbild überprüft. Im Folgenden ist der Befehl für das Szenario Datendiebstahl zu sehen. Die weiteren zwei Szenarien haben dasselbe Schema für den Befehl:

```
.\kape.exe --tsource D: --tdest
C:\Users\DFIR\Desktop\BACH\KAPE\output\tout --tflush --target
ICS_DATATHEFT --mdest C:\Users\DFIR\Desktop\BACH\KAPE\output\mout
--mflush --module ICS_DATATHEFT --gui
```

Die Dauer für das gesamte kopieren und parsen des Szenario Datendiebstahl beträgt 106,4023 Sekunden, für das Szenario Malware sind es rund 105,9632 Sekunden und die Artefakte für das Szenario Angriff von außen lassen sich in 172,2627 Sekunden kopieren und parsen. Es kann dabei für dieses Datenträgerabbild mit einer Größe von 23,4 GB von zwei bis drei Minuten gesprochen werden.

4.5 Ergebnisse der Evaluierung nach Szenario

Die Ausgabe wird in einem externen Verzeichnis zwischengespeichert, um sie im nächsten Schritt auf ihre Vollständigkeit zu überprüfen. Dafür werden zunächst die verschiedenen Protokolldateien (Logfiles) überprüft. Im ConsoeLog.txt werden für keines der drei Szenarien Fehlermeldungen festgestellt. Im Copylog können die jeweils für die Szenarien kopierten Dateien festgestellt werden. Dabei wurden einige Artefakte weder kopiert noch entsprechend geparst, da sie auf dem gewählten Datenträgerabbild nicht vorhanden sind. Alle vorhandenen Artefakte wurden auch entsprechend der Compound-Targets der Szenarien kopiert.

Im Skiplog konnten für das Szenario Datendiebstahl 22 Dateien, für das Szenario Malware zwölf Dateien und für das Szenario Angriff von außen 14 Dateien festgestellt werden. Allesamt wurden mit der Begründung „Deduplication“ vom Kopiervorgang ausgeschlossen. Dies bedeutet, dass diese Dateien bereits vorhanden sind und Doppelungen (anhand SHA1-Werten) ausgeschlossen wurden [34]. Soll dies verhindert werden, kann man dies durch abwählen der Konfiguration „Deduplicate“ unter den Targets abwählen. Dadurch wird dem Befehl die Option `--tdd false` hinzugefügt, was den Ausschluss deduplizierter Dateien verhindert.

Die Module sind, wie auch schon in der allgemeinen Evaluierung dargestellt, jeweils nach den selbst erstellten Kategorien in Verzeichnisse gegliedert. Ein Spezialfall stellt dabei weiterhin die Registry dar. Bei den jeweiligen Szenarien befinden sich die entsprechend geparsten Dateien aus den spezifischen Batch-Dateien des Tools RECmd im Verzeichnis. Dort befindet sich eine CSV-Datei mit allen Artefakten und ein weiteres Unterverzeichnis. In der Abbildung 34 ist die CSV-Übersichtsdatei für das Beispiel Angriff von außen im Timeline Explorer dargestellt.

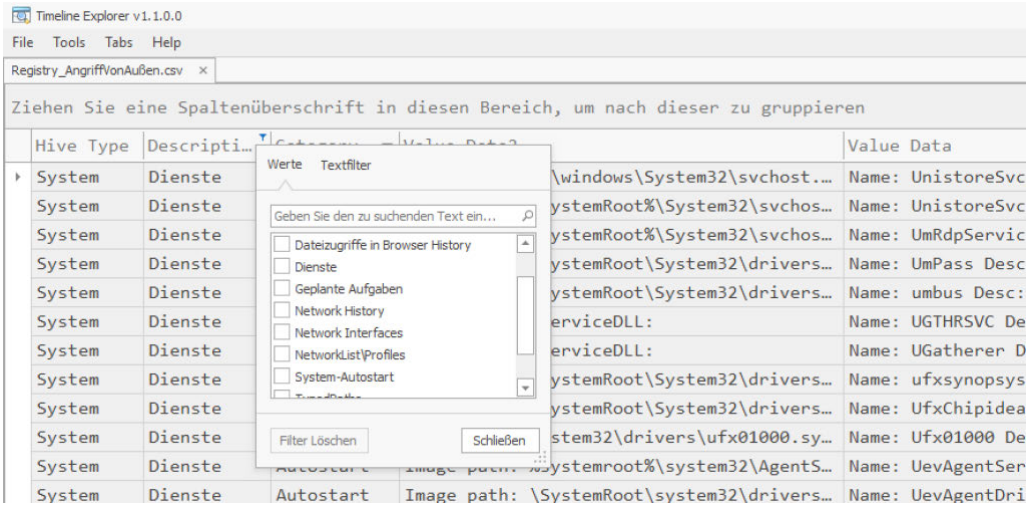


Abbildung 34 - Ausgabe Übersichtsdatei ICS_RECmd_IO.mkape

Es ist ersichtlich, dass alle entsprechenden Artefakte nach Kategorie und nochmal nach Artefakt Namen (Description) gegliedert sind. Durch Filterung können so ausschließlich gewünschte Kategorien oder sogar nur einzelne Artefakte angezeigt werden. Jedoch enthält diese Übersichtsdatei nicht alle Details zu allen Artefakten, weshalb in einem weiteren Unterverzeichnis weitere CSV-Dateien einzelner Artefakte zu finden sind. Diese weisen zusätzliche und detailliertere Informationen auf. Dazu gehört z.B. das Artefakt „UserAssist“, denn bei diesem ist zu erkennen, dass in der Übersichtsdatei die Values „Focus Time“ und „Focus Count“ fehlen, diese aber in der spezifischen Ausgabe im Unterverzeichnis in der von KAPE erstellten Datei Registry_AngriffVonAußen_UserAssist.csv vorhanden sind. Verallgemeinert kann also gesagt werden, dass die Übersichtsdatei in der Ausgabe sinnvoll für einen schnellen Überblick über alle aus der Registry geparsten Artefakte gibt. Will man aber genauere Ergebnisse und detailliertere Informationen zu einem der Artefakte, kann man die spezifischen, zusätzlichen CSV-Dateien aufrufen. Dies gilt für alle drei Szenarien.

Die restlichen Dateien unter dem Verzeichnis `\mout` sind jeweils vollständig (entsprechend der Szenarien) und wurden mit der Tabelle 3 abgeglichen. Dies setzt voraus, dass auch die Target-Compound Dateien entsprechend funktionieren, da ansonsten die Artefakte nicht geparst werden könnten. Die Größe des jeweiligen Outputs beträgt für die drei Szenarien jeweils zwischen 427 MB und 433 MB. Die Größe von dem individualisierten KAPE-Verzeichnis zuzüglich der beiden ausführbaren Dateien

`KAPE.exe` und `gkape.exe` und der selbst erstellten Targets, Module sowie Batch-Dateien beträgt lediglich 145 MB. Dies macht KAPE und seine Ausgabedateien sehr handlich und es reicht ein USB-Stick oder eine kleine Festplatte mit vorinstalliertem KAPE für die Sicherung von Dateien.

5 Diskussion

In diesem abschließenden Kapitel werden die Ergebnisse in Bezug auf die Forschungsfrage diskutiert. Weiterführend wird eine Einschätzung über die Limitierungen und das weitere Potenzial dieser Arbeit beleuchtet.

Das Ergebnis der vorangehenden Kapitel ist eine individualisierte, angepasste Version des Tools KAPE. Unter der Nutzung dieses Tools, welches sich in die Datenerhebung und Verarbeitung des Prozessschrittes Analyse einordnen lässt, wurde durch diese Arbeit eine Prozessautomatisierung bei einem Incident Response angestrebt. Dafür wurden insgesamt drei Szenarien gewählt, welche auf Grundlage der internen Arbeitsumgebung und mehreren Quellen unter Kapitel 3 am häufigsten vorkommen. Diesen Szenarien wurden jeweils 28-35 Artefakte zugeordnet. Dies entspricht ca. 56%-70% der 50 gesamten zur Auswahl stehenden Artefakte unter Tabelle 1, ausgenommen der in Kapitel 3 definierten, allgemeinen Artefakte.

Für die Umsetzung im Tool KAPE wurden sodann insgesamt 23 Targets erstellt, davon fünf Target-Compound Dateien. Zusätzlich wurden 38 Module erstellt, wovon sieben Modul-Compounds darstellen. Des Weiteren wurden drei .reb-Dateien erstellt, um die Registry-Ausgabe in einer CSV-Datei nach Szenario zu ermöglichen. Um die Übersicht zu erleichtern, wurde die Möglichkeit genutzt, nicht benötigte Target- und Modul-Dateien in das jeweilige `\!Disabled` Verzeichnis zu schieben, damit diese nicht mehr in den Reitern der grafischen Benutzeroberfläche erscheinen und nur die für die firmenintern relevanten Targets und Module sichtbar sind. So wurde auch die Übersichtlichkeit bedacht, welche sich bei einem Incident Response auch zeitsparend auswirkt.

Nach der Evaluierung und einigen Korrekturen unter Kapitel 4 steht fest, dass das individualisierte Tool funktioniert und die geschriebenen Targets und Module die erwarteten Ausgaben liefern. Im Fokus stand bei der Anpassung des Tools aber vor allem die Minimierung des zeitlichen Aufwands gegenüber des momentanen Prozessablaufs und die Einhaltung der in Kapitel 2 beschriebenen forensischen Prinzipien. Die zeitliche Reduzierung ist sowohl aus Sicht der personellen Ressourcen als auch aus Sicht der Wirtschaftlichkeit und der Schadensbegrenzung bei einem Sicherheitsvorfall sehr relevant.

5.1.1 Zeitlicher Aspekt

Bei der Evaluierung unter Kapitel 4 konnte für das Kopieren und Parsen der gesamten Artefakte eine Zeit von rund 249 Sekunden festgestellt werden, was in etwa vier Minuten entspricht. Für die einzelnen Szenarien konnte Zeiten zwischen 105 und 172 Sekunden

festgestellt werden, was zwei bis drei Minuten entspricht. Im Vergleich zum im Kapitel 2 vorgestellten aktuellen Vorgehen ist dies ein erheblicher Unterschied. Die Dauer des gesamten Vorgangs unter KAPE entspricht der zeitlichen Dauer eines Artefakts in der händischen Auswertung. Insbesondere bei der Erhebung benutzerspezifischen Daten müssen meist mehrere Benutzerkonten in Erwägung gezogen werden und einige Artefakte für jeden Benutzer einzeln geparst werden. KAPE schließt automatisch alle Benutzerkonten mit ein, für welche Informationen vorliegen, dies ist ein erheblicher Vorteil. Es kann also in Bezug auf die Forschungsfrage festgehalten werden, dass zeitlich eine erhebliche Verbesserung erreicht wurde. Die eingesparte Zeit kann so in die tatsächliche Analyse investiert werden.

Um dies zu erreichen, wurde bereits bei der Wahl der Artefakte unter Kapitel 3 darauf geachtet, nicht zu viele Artefakte für das jeweilige Szenario zu wählen, um bei diesen einen zeitlichen Vorteil gegenüber der Sicherung aller möglichen Artefakte aus Tabelle 1 zu erreichen. Zusätzlich ermöglicht eine Triage der Szenario-spezifischen Artefakte auch eine übersichtlichere Darstellung und ein schnelleres Navigieren in der jeweiligen Artefakt Ausgabe. In Abbildung 8 ist ersichtlich, dass so für die drei Szenarien nur jeweils 56%-70% der zur Verfügung stehenden Artefakte genutzt wurden. Dies entspricht einer Reduktion von beinahe 50%. Das Parsen aller Artefakte dauert bekanntermaßen etwa vier bis fünf Minuten. Das Parsen auf demselben Datenträger für die jeweiligen Szenarien dauert zwei bis drei Minuten, dies entspricht im besten Fall nochmal einer zeitlichen Reduktion von beinahe 50%.

Das unter Kapitel 2 beschriebene Vorgehen ohne die Nutzung des Tools KAPE beansprucht von der Datenerhebung bis zur ersten Analyse im Schnitt einen ganzen Arbeitstag. Durch KAPE ist es möglich, vor der Sicherung des kompletten Datenträgers eine Triage für das entsprechende Szenario zu ziehen, um dann möglichst schnell erste Ergebnisse auswerten zu können. Dies führt dazu, dass eine Entscheidung darüber getroffen werden kann, ob ein vollständiges Datenträgerabbild des untersuchten oder auch weiteren Systemen notwendig ist und welche Maßnahmen zur Eindämmung getroffen werden können. Zusätzlich müssen bei dem Vorgehen ohne KAPE alle Artefakte händisch durch die Zimmerman Tools geparst werden. Dies entspricht bei 25-30 Artefakten, welche zum Teil für mehrere Benutzer ausgewertet werden müssen in etwa der Arbeit von bis zu zwei Stunden.

5.1.2 Aspekt der forensischen Prinzipien

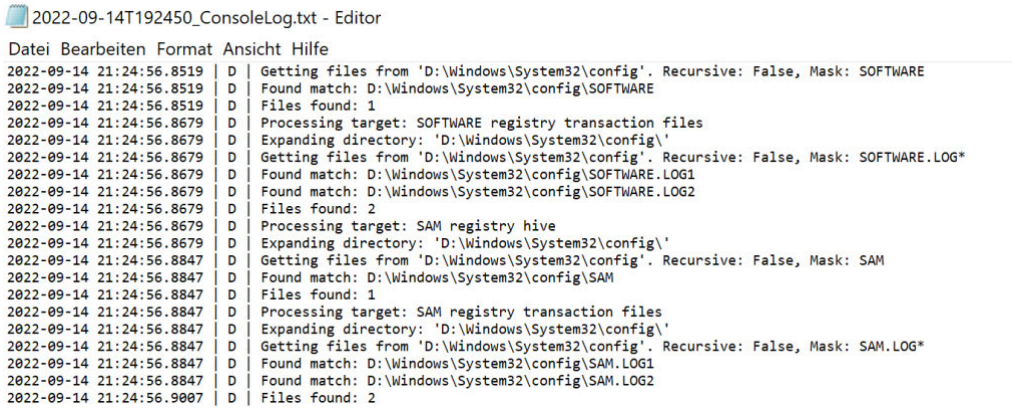
Durch die Evaluierung konnte in Bezug auf die Forschungsfrage festgestellt werden, dass die forensischen Prinzipien unter der Beachtung einiger Vorschriften entsprechend umgesetzt werden können. Die Akzeptanz der Methode ist aus dem gegebenen Anlass, dass es sich um ein Tool von Eric Zimmerman handelt, gegeben. Es konnte durch die Recherche in Kapitel 2 festgestellt werden, dass das verwendete Tool in der forensischen

Wissenschaft weit verbreitet ist und entsprechende Dokumentationen über sein Vorgehen vorhanden sind.

Bei der in Kapitel 2.1.2 erläuterten Live-Analyse ist darauf zu achten, nicht zu viele Spuren zu hinterlassen. Daher wird empfohlen, einen Datenträger mit vorinstalliertem KAPE an das Analysesystem anzuschließen und nicht mit der grafischen Benutzeroberfläche zu arbeiten, sondern eine Befehlszeile vorzubereiten und diese dann entsprechend auszuführen. Zusätzlich sollen die Artefakte nicht auf dem zu analysierenden System geparst werden, sondern lediglich durch die Targets auf dem entsprechenden Datenträger gesichert werden. Aber auch dadurch können Spuren der Sicherung nicht vollständig verhindert werden. Jedoch ist dies bei entsprechender Dokumentation akzeptabel. Zusätzlich ist es wichtig das Vier-Augen-Prinzip einzuhalten, während KAPE gestartet wird und durchläuft.

Die entsprechende Dokumentation übernimmt KAPE durch das ConsoleLog.txt selbst. Dort werden unter Verwendung von Zeitstempeln alle auf der Konsole ausgegebenen Befehle und Informationen gespeichert. Unter Aktivierung der Debug- und Trace-Nachrichten, kann dies noch weiter ins Detail geführt werden. Es wird sowohl das verwendete Kommando als auch die Version von KAPE und das Verzeichnis, aus dem es gestartet wurde, vermerkt. Zusätzlich beinhaltet das ConsoleLog.txt auch Systeminformationen über das laufende System. Durch die Option `--flush` ist es auch möglich, sicherzugehen, dass das jeweilige Ausgabe-Verzeichnis der Targets und Module vorher komplett geleert wird. Dazu wird es gelöscht und neu angelegt. Auch der Schritt der Laufwerkbuchstabenänderung ist ersichtlich, denn KAPE passt den Laufwerkbuchstaben während der Laufzeit auf allen Targets und Modulen an.

Außerdem sind durch das ConsoleLog.txt die verwendeten Targets und Module, die jeweils gefundenen Dateien und deren Namen zu erkennen. Dies macht die Wiederholbarkeit möglich. Ein Ausschnitt des ConsoleLog.txt für die Evaluierung unter 4.2 ist in folgender Abbildung 35 zu sehen.


2022-09-14T192450_ConsoleLog.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
2022-09-14 21:24:56.8519 | D | Getting files from 'D:\Windows\System32\config'. Recursive: False, Mask: SOFTWARE
2022-09-14 21:24:56.8519 | D | Found match: D:\Windows\System32\config\SOFTWARE
2022-09-14 21:24:56.8519 | D | Files found: 1
2022-09-14 21:24:56.8679 | D | Processing target: SOFTWARE registry transaction files
2022-09-14 21:24:56.8679 | D | Expanding directory: 'D:\Windows\System32\config\'
2022-09-14 21:24:56.8679 | D | Getting files from 'D:\Windows\System32\config'. Recursive: False, Mask: SOFTWARE.LOG*
2022-09-14 21:24:56.8679 | D | Found match: D:\Windows\System32\config\SOFTWARE.LOG1
2022-09-14 21:24:56.8679 | D | Found match: D:\Windows\System32\config\SOFTWARE.LOG2
2022-09-14 21:24:56.8679 | D | Files found: 2
2022-09-14 21:24:56.8679 | D | Processing target: SAM registry hive
2022-09-14 21:24:56.8679 | D | Expanding directory: 'D:\Windows\System32\config\'
2022-09-14 21:24:56.8847 | D | Getting files from 'D:\Windows\System32\config'. Recursive: False, Mask: SAM
2022-09-14 21:24:56.8847 | D | Found match: D:\Windows\System32\config\SAM
2022-09-14 21:24:56.8847 | D | Files found: 1
2022-09-14 21:24:56.8847 | D | Processing target: SAM registry transaction files
2022-09-14 21:24:56.8847 | D | Expanding directory: 'D:\Windows\System32\config\'
2022-09-14 21:24:56.8847 | D | Getting files from 'D:\Windows\System32\config'. Recursive: False, Mask: SAM.LOG*
2022-09-14 21:24:56.8847 | D | Found match: D:\Windows\System32\config\SAM.LOG1
2022-09-14 21:24:56.8847 | D | Found match: D:\Windows\System32\config\SAM.LOG2
2022-09-14 21:24:56.9007 | D | Files found: 2
```

Abbildung 35 - ConsoleLog.txt Beispiel

Das wohl wichtigste forensische Prinzip, welches es bei einer IT-forensischen Untersuchung zu beachten gibt, ist die Wahrung der Integrität der Daten. Da KAPE die Dateien mitsamt den Metadaten kopiert, werden zum einen keine Zeitstempel verändert und zum anderen wird ab diesem Zeitpunkt nicht mehr an der Originaldatei gearbeitet. Durch die Targets wird eine 1:1 Kopie der jeweiligen Dateien unter Beibehaltung der Struktur des Originalpfades in die Ausgabedatei geschrieben. Außerdem existiert unter KAPE auch ein Hash-Verfahren, denn es wird für jede kopierte Datei im CopyLog.csv der SHA-1-Wert gespeichert. Dieser kann dann mit den Hash-Werten der Originaldateien abgeglichen werden, um sicherzugehen, dass an den Dateien während des Kopiervorgangs keine Veränderungen vorgenommen wurden.

5.2 Limitation und Potenzial

Die Limitation der vorliegenden Arbeit ergibt sich vor allem durch den Mangel an testbaren Datenträgerabbilder, welche echte Sicherheitsvorfälle repräsentieren. Aufgrund der Vielfältigkeit und Individualität, die ein forensischer Vorfall aufweist, bedarf es mehrmaliger Testläufe des individualisierten Tools an echten Fällen. Die Targets und Module sowie deren Compound-Dateien sollen möglichst mehrmals mit verschiedenen Images und insbesondere auch an Live-Fällen getestet werden. Nicht jedes Datenträgerabbild enthält alle Artefakte und auch nicht jeder Datenträger ist exakt gleich aufgebaut. In dieser Arbeit lag der Fokus darauf, herauszufinden, ob das Tool KAPE bei der Prozessautomatisierung unterstützen kann und inwiefern dies umzusetzen ist. Dafür wurde beispielhaft auch eine Evaluierung durchgeführt. Die weitere Testung der erstellten Targets und Module soll in naher Zukunft erfolgen und bietet Potenzial für Weiterentwicklungen und Ergänzungen. Potenziell können weitere Targets und Module hinzugefügt werden oder sogar weitere Szenarien erstellt werden. Die bereits erstellten Targets und Module können dabei nach Belieben zusammengefügt oder auch einzeln ausgewählt werden.

Im Hinblick auf die erstellten Targets und Module ist es vorstellbar, einige noch weiter zu verbessern. Da die Artefakte unter „Webzugriffe“ durch das Tool browsinghistoryview.cmd nicht geparst werden konnten, besteht die Alternative zum jetzigen Zeitpunkt darin, die durch die Targets kopierten Artefakte durch ein externes Tool einzuspeisen und dort zu analysieren. Dahingehend besteht bei weiteren Forschungen in diesem Bereich Potenzial, auch diese Artefakte vollständig in die Automatisierung einzubinden. Auch die Ausgabe der Dateien, welche durch das Tool RECmd geparst wurden, können noch verbessert werden. Die erstellten .reb-Batchdateien erzeugen noch immer sehr viele Störfaktoren in den Ausgabedateien. Mit Störfaktoren sind vorwiegend Artefakt-Ausgaben gemeint, welche zwar geparst wurden, für die Analyse aber keine Relevanz haben. Darunter lässt sich auch das unter Kapitel 4.3.2 beschriebene „TimeZoneKey“-Problem einordnen.

Nach einer ausreichenden Testung im Bereich einer IT-forensischen Analyse eines Arbeitsplatzrechners oder Server ist es vorstellbar, dass die erstellte Version des Tools

auch dort anwendbar ist. Dabei kann KAPE auf dem gesicherten Datenträgerabbild angewendet werden. Für die anschließende Erstellung eines forensischen Berichts werden in der internen Firmenumgebung die CSV-Dateien in Excel-Arbeitsmappen umgewandelt. Ein weiterer Optimierungspunkt ist es, auch diese Arbeit durch ein ergänzendes Skript noch zu automatisieren, um die durch KAPE geparsten CSV-Dateien automatisiert in Excel-Arbeitsmappen mit entsprechend vorgegebenem Layout umzuwandeln.

Da Memory-Forensik bei einem Incident-Response relevante Informationen liefern kann, ist es durchaus möglich, sich in Zukunft bei der weiteren Individualisierung des Tools KAPE mit dieser zu beschäftigen. Dabei kann sich auf das Hinzufügen und entsprechende Parsen von Memory-Artefakten konzentriert werden. Die Forschung kann dahin geführt werden, auch die Erhebung und das Parsen der Memory-Artefakte bei einem Incident Response zu automatisieren.

KAPE bietet unter der Enterprise-Lizenz auch die Möglichkeit einer Remote-Sicherung an. Auch die Umsetzung dieser unter dem Aspekt der forensischen Prinzipien stellt eine zukünftige Forschungsfrage dar.

6 Fazit

Es ist festzuhalten, dass die Automatisierung sowohl das Problem, als auch die Lösung darstellt. Denn genau so vorteilhaft wie sich eine Automatisierung einiger Teile des Incident Response im Arbeitsprozess erwiesen hat, wird sie auch von Angreifern als Angriffsmethode genutzt. Das Verlangen nach zunehmender Automatisierung von Prozessen und Methoden steigt mit dem Stand der Technik.

Das Ziel der vorliegenden Arbeit war es, den Prozess des Incident Response effektiver zu gestalten und eine Basis für die interne Arbeitsumgebung zu bieten, schneller und effektiver zu arbeiten. Dazu sollte das in der vorangegangenen Praxisarbeit evaluierte Tool KAPE genutzt werden. Der Fokus lag dabei vorrangig auf der Reduzierung des zeitlichen Aufwands der IT-Forensischen Analyse, was durch eine Automatisierung erreicht werden sollte. Denn diese zeitliche Reduktion ist sowohl aus Sicht der Personalressourcen als auch aus wirtschaftlicher Sicht sehr relevant. So nahm die Forschungsfrage ihre Form an: Kann die Automatisierung eines Incident Response mit dem Tool KAPE schneller und effektiver gestaltet werden, ohne die forensischen Prinzipien außer acht zu lassen?

Zur Beantwortung dieser, wurden beginnend Grundlagen zu den IT-forensischen Prinzipien und dem Prozess eines Incident Response erläutert. Zusätzlich wurden relevante Tools und Artefakte sowie das momentane Vorgehen vorgestellt. Die Umsetzung und das methodische Vorgehen gestaltete sich durch die Entscheidung für drei häufig vorkommende Vorfalls-Szenarien und der Wahl der Artefakte unter entsprechender Begründung der Relevanz. Anschließend wurden in KAPE verschiedene Targets und Module für die jeweiligen Artefakte erstellt um diese, nach den zuvor recherchierten Szenarien, zu Compounds zusammen zu fassen. Schließlich wurden die erstellten Targets und Module erst allgemein und dann nach Szenario an einem Test-Image evaluiert. Dabei ging es hauptsächlich um den zeitlichen Aspekt und den Aspekt der forensischen Prinzipien. Nach der ersten Evaluierung wurden erneut Optimierungen und Verbesserungen vorgenommen.

Zu den wichtigsten Ergebnissen zählt dabei das individualisierte KAPE mit eigenen Targets und Modulen sowie Compounds für die jeweiligen Vorfalls-Szenarien. Das Tool ist funktionsfähig und hält die forensischen Prinzipien ein. Außerdem erreicht KAPE eine enorme zeitliche Verbesserung des Prozesses und schafft durch das automatische Kopieren und Parsen der Artefakte Zeit für andere Tätigkeiten. Es handelt sich dabei um eine Zeitersparnis für die Datenerhebung und Vorverarbeitung von bis zu 2 Stunden.

Die Forschungsfrage, ob die Automatisierung eines Incident Response mit dem Tool KAPE schneller und effektiver gestaltet werden kann, ohne die forensischen Prinzipien

außer acht zu lassen, kann somit mit ja beantwortet werden. Durch diese Arbeit ist es gelungen, mittels KAPE einen alternativen, automatisierten Prozess für das Notfallmanagement und einen Incident Response zu entwickeln welcher sowohl zeitliche Vorteile bringt, als auch die forensischen Prinzipien einhält.

Dennoch bedarf es an dem Tool noch an ständigen Verbesserungen und Weiterentwicklungen, denn es funktionieren noch nicht alle Module optimal. Zum Teil muss deshalb auf alternative und externe (händische) Verarbeitung zurückgegriffen werden um einige der Artefakte zu analysieren. Andere Artefakt-Ausgaben sollen in Zukunft noch übersichtlicher gestaltet werden. Dennoch bietet KAPE einen sehr guten Fortschritt zum momentanen Vorgehen und hat noch viel Potenzial sich weiterzuentwickeln.

Literatur

[1] T. Roy und A. Jain, „Windows registry forensics: An imperative step in tracking data theft via USB devices“, *International Journal of Computer Science and Information Technologies,*, Vol. 3, Nr. 3, S. 4427–4433, 2012.

[2] *Computer Security Incident Handling Guide*, NIST.SP.800-61r2, National Institute of Standards and Technology, 2012. Zugriff am: 12. August 2022. [Online]. Verfügbar unter: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

[3] J. T. Luttgens, M. Pepe und K. Mandia, *Incident Response & Computer Forensics, Third Edition*, 3. Aufl. McGraw-Hill Education, 2014.

[4] M. Bromiley. „SANS 2022 ransomware defense report | SANS institute“. Cyber Security Training | SANS Courses, Certifications & Research. https://www.sans.org/white-papers/sans-2022-ransomware-defense-report/ (Zugriff am 12. August 2022).

[5] B. Schneier, „The future of incident response“, *IEEE Security & Privacy*, Bd. 12, Nr. 5, S. 96, September 2014. Zugriff am: 14. August 2022. [Online]. Verfügbar unter: https://doi.org/10.1109/msp.2014.102

[6] A. Torres, „Incident response: How to fight back“, Survey, SANS, 2014. Zugriff am: 9. August 2022. [Online]. Verfügbar unter: https://dsimg.ubm-us.net/envelope/342373/288172/1412628284_3_INCIDENT_RESPONSE_SURVEY.pdf

[7] „Leitfaden IT-Forensik“. Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/dok/6620610 (Zugriff am 8. August 2022).

[8] D. Labudde und M. Mohaupt, *Bioinformatik im Handlungsfeld der Forensik*. Berlin, Heidelberg: Springer, 2018. Zugriff am: 18. August 2022. [Online]. Verfügbar unter: https://doi.org/10.1007/978-3-662-57872-8

[9] A. Klick-Strehl. „IT-Forensik: Relevante Daten am Tatort identifizieren und sichern“. Dr. Datenschutz. https://www.dr-datenschutz.de/it-forensik-relevante-daten-am-tatort-identifizieren-und-sichern/ (Zugriff am 22. Juli 2022).

[10] F. Y. W. Law *et al.*, „Protecting digital data privacy in computer forensic examination“, in *2011 IEEE Sixth International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE)*, Oakland, CA, USA, 26. Mai 2011. IEEE, 2011. Zugriff am: 29. Juli 2022. [Online]. Verfügbar unter: https://doi.org/10.1109/sadfe.2011.15

[11] R. Bodach, *Skript Computerforensische Methoden*, 3. Aufl. Hochschule Mittweida, 2021

[12] G. Johansen, *Digital Forensics and Incident Response: A Practical Guide to Deploying Digital Forensic Techniques in Response to Cyber Security Incidents*. Packt Publishing, 2017.

[13] B. P. Kondapally, „Forensically important artifacts in windows operating systems“, Paper, TCS Enterprise Security And Risk Management. Zugriff am: 22. Juli 2022. [Online]. Verfügbar unter: https://docplayer.net/29282798-Forensically-important-artifacts-in-windows-operating-systems.html

[14] D. Waltermire, K. Scarfone und M. Casipe, „Specification for the open checklist interactive language (OCIL) version 2.0“, National Institute of Standards and Technology, Gaithersburg, MD, 2011. Zugriff am: 19. September 2022. [Online]. Verfügbar unter: https://doi.org/10.6028/nist.ir.7692

[15] C. Crowley, „Scoping an intrusion using identity, host, and network indicators“, Whitepaper, SANS, 2021. Zugriff am: 20. Juli 2022. [Online]. Verfügbar unter: https://www.sans.org/white-papers/40250/

[16] A. Ashcraft, K. Sharkey, D. Coulter, D. Batchelor und M. Satran. „Master file table (local file systems) - win32 apps“. Microsoft Learn: Build skills that open doors in your career. https://learn.microsoft.com/en-us/windows/win32/fileio/master-file-table (Zugriff am 2. August 2022).

[17] R. Lee. „Windows forensic analysis | SANS poster“. Cyber Security Training | SANS Courses, Certifications & Research. https://www.sans.org/posters/windows-forensic-analysis/ (Zugriff am 9. Juli 2022).

[18] J. Davis, „How to use kape for fast and flexible incident response“, GIAC (GCIH) Gold Certification, SANS, 2020. Zugriff am: 16. Juli 2022. [Online]. Verfügbar unter: https://www.giac.org/paper/gcih/34611/kape-fast-flexible-incident-response/152146

[19] S. White, K. Sharkey, D. Coulter, D. Batchelor und M. Satran. „Structure of the registry - win32 apps“. Microsoft Learn: Build skills that open doors in your career. https://learn.microsoft.com/en-us/windows/win32/sysinfo/structure-of-the-registry (Zugriff am 15. Juli 2022).

[20] N. A. Hassan, *Digital Forensics Basics*. Berkeley, CA: Apress, 2019. Zugriff am: 17. Juli 2022. [Online]. Verfügbar unter: https://doi.org/10.1007/978-1-4842-3838-7

[21] A. Davis, „Leveraging the application compatibility cache in forensic investigations“, Whitepaper, Mandiant, 2012. Zugriff am: 13. August 2022. [Online]. Verfügbar unter: https://www.fireeye.com/content/dam/fireeye-www/services/freeware/shimcache-whitepaper.pdf

[22] K. Bridge *et al.* „Event logging (event logging) - win32 apps“. Microsoft Learn: Build skills that open doors in your career. https://docs.microsoft.com/en-us/windows/win32/eventlog/event-logging (Zugriff am 16. Juli 2022).

[23] „Eventlog Key - Win32 apps“. Microsoft Learn: Build skills that open doors in your career. https://docs.microsoft.com/en-us/windows/win32/eventlog/eventlog-

key (Zugriff am 16. Juli 2022).

[24] „Windows Eventlogs: Welche Ereignisse sie verraten“. Dr. Datenschutz. https://www.dr-datenschutz.de/windows-eventlogs-welche-ereignisse-sie-verraten/ (Zugriff am 14. August 2022).

[25] K. Bridge, K. Sharkey, D. Batchelor, D. Coulter, M. Jacobs und M. Satran. „Event identifiers (event logging) - win32 apps“. Microsoft Learn: Build skills that open doors in your career. https://docs.microsoft.com/en-us/windows/win32/eventlog/event-identifiers (Zugriff am 16. Juli 2022).

[26] „Windows event log analysis software, view and monitor system, application and security event logs — FSPro Labs“. FSPro Labs. https://eventlogxp.com/ (Zugriff am 7. August 2022).

[27] M. Bromiley. „OAlerts — The Microsoft Office Event Log“. Medium. https://bromiley.medium.com/oalerts-the-microsoft-office-event-log-ad164e1eec0f (Zugriff am 30. Juli 2022).

[28] R. Lee und M. Pilkington. „Hunt Evil | SANS Poster“. Cyber Security Training | SANS Courses, Certifications & Research. https://www.sans.org/posters/hunt-evil/ (Zugriff am 16. Juli 2022).

[29] E. Zimmerman. „KAPE Documentation“. Eric Zimmerman's tools. https://ericzimmerman.github.io/KapeDocs/#!index.md (Zugriff am 11. Juli 2022).

[30] E. Zimmerman. „Kroll Artifact Parser and Extractor - KAPE“. Kroll. https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape (Zugriff am 11. Juli 2022).

[31] E. Zimmerman. „KAPE Documentation - Targets“. Eric Zimmerman's tools. https://ericzimmerman.github.io/KapeDocs/#!Pages\2.1-Targets.md (Zugriff am 11. Juli 2022).

[32] E. Zimmerman. „KAPE Documentation - Modules“. Eric Zimmerman's tools. https://ericzimmerman.github.io/KapeDocs/#!Pages\2.2-Modules.md (Zugriff am 11. Juli 2022).

[33] E. Zimmerman. „KAPE Documentation - Getting started“. Eric Zimmerman's tools. https://ericzimmerman.github.io/KapeDocs/#!Pages\2.-Getting-started.md (Zugriff am 11. Juli 2022).

[34] E. Zimmerman. „KAPE Documentation - Using KAPE“. Eric Zimmerman's tools. https://ericzimmerman.github.io/KapeDocs/#!Pages\3.-Using-KAPE.md (Zugriff am 11. Juli 2022).

[35] E. Zimmerman. „KAPE Documentation - Using gkape“. Eric Zimmerman's tools. https://ericzimmerman.github.io/KapeDocs/#!Pages\5.-gkape.md (Zugriff am 11. Juli 2022).

[36] B. Ingerson, C. C. Evans und O. Ben-Kiki. „Yet Another Markup Language (YAML) 1.0“. The Official YAML Web Site. https://yaml.org/spec/history/2001-12-10.html (Zugriff am 3. August 2022).

[37] E. Zimmerman. „KAPE Documentation - Log-files“. Eric Zimmerman's tools. https://ericzimmerman.github.io/KapeDocs/#!Pages\4.-Log-files.md (Zugriff am 11. Juli 2022).

[38] E. Zimmerman. „EZ Tools | SANS Institute“. Cyber Security Training | SANS Courses, Certifications & Research. https://www.sans.org/tools/ez-tools/ (Zugriff am 13. Juli 2022).

[39] E. Zimmerman. „Eric Zimmerman's tools“. Eric Zimmerman's tools. https://ericzimmerman.github.io/#!index.md (Zugriff am 13. Juli 2022).

[40] E. Zimmerman. „AmcacheParser“. ericzimmerman.github.io. https://ericzimmerman.github.io/documentation/Amcach

eParser/ (Zugriff am 20. Juli 2022).

[41] E. Zimmerman. „JLECmd“. ericzimmerman.github.io. https://ericzimmerman.github.io/documentation/JLECmd/ (Zugriff am 20. Juli 2022).

[42] E. Zimmerman. „RBCmd“. ericzimmerman.github.io. https://ericzimmerman.github.io/documentation/RBCmd/ (Zugriff am 20. Juli 2022).

[43] “Das Recht im Fokus der digitalen Forensik“. Dr Datenschutz. https://www.dr-datenschutz.de/das-recht-im-fokus-der-digitalen-forensik/ (Zugriff am 29. Juli 2022).

[44] J. Grier, „Detecting data theft using stochastic forensics“, *Digital Investigation*, Volume 8, S. 71–77, August 2011. Zugriff am: 2. September 2022. [Online]. Verfügbar unter: https://doi.org/10.1016/j.diin.2011.05.009

[45] C. Tilbury. „OpenSaveMRU and LastVisitedMRU | SANS Institute“. Cyber Security Training | SANS Courses, Certifications & Research. https://www.sans.org/blog/opensavemru-and-lastvisitedmru/ (Zugriff am 12. August 2022).

[46] „Typed Paths“. Forensafe. https://forensafe.com/blogs/typedpaths.html (Zugriff am 26. September 2022).

[47] „Thumbnails - Forensics Wiki“. Forensics Wiki. https://forensicswiki.xyz/wiki/index.php?title=Thumbnails (Zugriff am 11. August 2022).

[48] M. Brand, C. Valli und A. Woodward, „Malware Forensics: Discovery of the Intent of Deception“, *Journal of Digital Forensics, Security and Law*, Bd. 5, Nr. 4, Artikel 2, 2010. Zugriff am: 8. August 2022. [Online]. Verfügbar unter: https://doi.org/10.15394/jdfsl.2010.1082

[49] „Schadprogramme | Universität Mannheim“. Universität Mannheim. https://www.uni-mannheim.de/informationssicherheit/sicherheitstipps/schadprogramme/#c125596 (Zugriff am 2. September 2022).

[50] Bundeskriminalamt, Hrsg., „Cybercrime Bundeslagebild 2019“, Wiesbaden, September 2020. Zugriff am: 9. Juli 2022. [Online]. Verfügbar unter: https://www.bka.de/SharedDocs/Downloads/DE/Publikationen/JahresberichteUndLagebilder/Cybercrime/cybercrimeBundeslagebild2019.html;jsessionid=57690F3440A11F7E76F04DD16C8035A0.live601?nn=28110

[51] A. Podile, K. Gottumukkala und K. S. Pendyala, „Digital forensic analysis of malware infected machine- case study“, *International Journal of Scientific & Technology Research*, Volume 4, Issue 09, S. 346–349, September 2015. Zugriff am: 8. August 2022. [Online]. Verfügbar unter: https://www.ijstr.org/final-print/sep2015/Digital-Forensic-Analysis-Of-Malware-Infected-Machine-Case-Study.pdf

[52] *Internet Security Glossary, Version 2*, RFC 4949, Internet Engineering Task Force, 2007. Zugriff am: 17. Juli 2022. [Online]. Verfügbar unter: https://www.rfc-editor.org/rfc/rfc4949

[53] O. Babaoglu, „Cybersecurity: Intrusion Detection and Cyber Forensics“, Zusammenfassung der Präsentation, ALMA MATER STUDIORUM – UNIVERSITA’ DI BOLOGNA. Zugriff am: 2. September 2022. [Online]. Verfügbar unter: https://www.cs.unibo.it/~babaoglu/courses/security/lucidi/pdf/IDS.pdf

[54] „Configuration Recommendations for Windows 10 Logging“. BSI - Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Cyber-Security/SiSyPHuS/AP10/Logging_Configuration_Guideline.pdf?__blob=publicationFile&v=5 (Zugriff am 19. August 2022).

[55] E. Skoudos. „Intrusion Discovery Cheat Sheet for Windows | Cheat Sheet“. Cyber Security Training | SANS Courses, Certifications & Research. https://www.sans.org/posters/intrusion-discovery-cheat-sheet-for-windows/ (Zugriff am 9. August 2022).

[56] D. J. Chaboya, R. A. Raines, R. O. Baldwin und B. E. Mullins, „Network Intrusion Detection: Automated and Manual Methods Prone to Attack and Evasion“, *IEEE Security and Privacy*, Bd. 4, Nr. 6, S. 36–43, November 2006. Zugriff am: 26. Juli 2022. [Online]. Verfügbar unter: https://doi.org/10.1109/msp.2006.159

[57] G. Wegberg, „KAPE-Einführung, Teil 2: Autoruns-Artefakte auswerten und verstehen“, *iX Magazin*, Nr. 08/2021, S. 100–105, August 2021. Zugriff am: 12. August 2022. [Online]. Verfügbar unter: https://www.oneconsult.com/wp-content/uploads/2021/07/ix.2021.08.100_105.pdf

[58] „7-Zip“. 7-Zip. https://www.7-zip.org/ (Zugriff am 31. August 2022).

[59] G. Wegberg, „KAPE-Einführung, Teil 3: Browserhistorie auswerten und verstehen“, *iX Magazin*, Nr. 09/2021, S. 132–136, September 2021. Zugriff am: 31. Juli 2022. [Online]. Verfügbar unter: https://www.oneconsult.com/wp-content/uploads/2021/08/ix.2021.09.132_136.pdf

[60] L. Pixley und C. Elwell, „Computer Forensics CCIC Training, Chapter 7: Recycle Bin“, Version 3, Mai 2017. Zugriff am: 22. Juli 2022. [Online]. Verfügbar unter: https://cci.calpoly.edu/sites/default/files/2021-06/2020_autopsy_training_manual_FULL.pdf

[61] D. Coulter *et al.* „The Transaction Log (SQL Server) - SQL Server“. Microsoft Learn: Build skills that open doors in your career. https://docs.microsoft.com/de-de/sql/relational-databases/logs/the-transaction-log-sql-server?view=sql-server-ver16 (Zugriff am 24. September 2022).

[62] D. Via. „Digging Up the Past: Windows Registry Forensics Revisited | Mandiant“. Mandiant. https://www.mandiant.com/resources/blog/digging-up-the-past-windows-registry-forensics-revisited (Zugriff am 19. Juli 2022).

[63] E. Kutcher. „Thumbcache Viewer - Extract thumbnail images from the thumbcache_*.db and iconcache_*.db database files.“ Thumbcache Viewer. http://thumbcacheviewer.github.io (Zugriff am 9. September 2022).

[64] N. Sofer. „View the browsing history of your Web browser“. NirSoft. https://www.nirsoft.net/utils/browsing_history_view.html (Zugriff am 9. September 2022).

[65] A. Rathbun. „DFIRPowerShellScripts“. GitHub. https://github.com/AndrewRathbun/DFIRPowerShellScripts (Zugriff am 9. September 2022).

[66] A. Fortuna. „RECmd: command line tool for Windows Registry analysis“. Andrea Fortuna. https://andreafortuna.org/2020/03/04/recmd-command-line-tool-for-windows-registry-analysis/ (Zugriff am 20. August 2022).

[67] E. Zimmerman. „EricZimmerman/evtx: C# based evtx parser with lots of extras“. GitHub. https://github.com/EricZimmerman/evtx (Zugriff am 26. August 2022).

[68] „CFReDS Portal“. cfreds.nist.gov. https://cfreds.nist.gov/all/MagnetForensics/2019WindowsMagnetCTF (Zugriff am 17. September 2022).

[69] „ArsenalRecon/Arsenal-Image-Mounter“. GitHub. https://github.com/ArsenalRecon/Arsenal-Image-Mounter (Zugriff am 11. September 2022).

Anlagen

Anlagen, Tabelle 1 – Artefakte

Artefakte	Speicherort	Pfad	Kurzbeschreibung
Windows Registry:			
SYSTEM	Dateisystem	C:\Windows\System32\Config\SYSTEM	Windows Einstellungen, werden z.T. schon beim Start benötigt (Treiber und Dienste)
SOFTWARE	Dateisystem	C:\Windows\System32\Config\SOFTWARE	Systemweit geltende Einstellungen für Windows und Anwendungen
SAM	Dateisystem	C:\Windows\System32\Config\SAM	"Security Accounts Manager", enthält Anmeldenamen und Zeitstempel
NTUSER.dat	Dateisystem	C:\Users\<user>\NTUSER.dat	Wichtige benutzerdefinierte Einstellungen
USRCLASS.dat	Dateisystem	C:\Users\<user>\AppData\Local\Microsoft\Windows\USRCLASS.dat	Benutzerspezifische Informationen
Systemanalyse:			
$MFT	Dateisystem	C:\$MFT	Zeichnet alle Dateieinträge auf einem Volume auf
CurrentVersion	Registry	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion	Installiertes Betriebssysetm
ComputerName	Registry	HKLM\SYSTEM\CurrentControlSet\Control\computername\ComputerName	Name des Computers
TimeZoneInformation	Registry	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation	Systemzeitzone
Users	Registry	HKLM\SAM\Domains\Account\Users	Benutzerkonten + SID
Users	Registry	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList	Benutzerkonten und ihre letzten Logins
Dateiöffnung und -erstellung:			
WordWheelQuery	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery	Zuletzt verwendete Suchbegriffe in der Startmenüleiste
Shellbags	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\Shell\Bags und …\Shell\BagMRU	zuletzt aufgerufene Dateien/Verzeichnisse, sowohl lokal als auch im Netzwerk oder auf entfernbaren Datenträgern
		HKCU\USRCLASS.dat\Local Settings\Software\Microsoft\Windows\Shell\Bags und …\Shell\BagMRU	
ComDlg32: LastVisitedMRU	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU	erfasst .exe-Dateien, die von Anwendungen verwendet werden, um Dateien zu öffnen
ComDlg32:	Registry	HKCU\NTUSER.dat\Software\Microsoft\Wind	erfasst Dateien, die

Open/SaveMRU		ows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU	innerhalb Windows-Shell-Dialogfelds geöffnet/gespeichert werden
RecentDocs	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs	Zuletzt geöffnete Dateien und Verzeichnisse (im Startmenü eingepflegt)
OfficeUserMRU	Registry	HKCU\NTUSER.dat\Software\Microsoft\Office\<Version>\UserMRU\LiveID_####\FileMRU (Excel, Word und PowerPoint Keys)	Zuletzt geöffnete Dateien und Verzeichnisse aus MSOffice
Shortcuts/LNK-Files	Dateisystem	C:\Users\<user>\AppData\Roaming\Microsoft\Windows\Recent für Office: \Windows\Office\Recent	Automatisch erstellte Verknüpfungen von kürzlich geöffneten Dateien
Dateizugriffe in BrowserHistory	Registry	HKLM\SOFTWARE\Microsoft\Internet Explorer	Enthält neben Browserspuren auch lokale, entfernbare und remote Dateizugriffe
Dateizugriffe im Microsoft Office Log	Eventlogs	C:\Windows\System32\winevt\Logs\OAlerts.evtx, (Event-ID: 300)	Inhalte von Dialogen des Programms (Name des Programms, Meldungen zu Speicherungen)
Typed Paths	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\TypedPaths	Zuletzt eingegebene Pfade im File Explorer
Thumbcache	Dateisystem	C:\User\<user>\AppData\Local\Microsoft\Windows\Explorer\thumbcache_*.db	Verkleinerte Versionen von Bildern, Dokumenten (Vorschaubilder)
Recycle Bin	Dateisystem	C:\$Recycle.Bin\	Gelöschte Dateien
Webzugriffe:			
Edge-Browser History	Dateisystem	C:\Users\<user>\AppData\Local\Microsoft\Edge\User Data\Default\	Informationen über Webzugriffe und Internetnutzung
Internet-Explorer Browser History	Dateisystem	C:\Users\<user>\AppData\Local\Microsoft\Windows\History\	
Chrome-Browser History	Dateisystem	C:\Users\<user>\AppData\Local\Google\Chrome\User Data\Default\History	
Firefox-Browser History	Dateisystem	C:\Users\<user>AppData\Roaming\Mozilla\Firefox\Profiles\	
E-Mail-Artefakte:			
Outlook-Datendatei	Dateisystem	C:\Users\<user>\AppData\Local\Microsoft\Outlook*.pst	Personal-Storage-Table: enthält E-Mails, Termine, Aufgaben, Notizen und archivierte Mailboxdateien
		C:\Users\<user>\Documents\Outlook Files*.pst	
Outlook-Offlinedatendatei	Dateisystem	C:\User\<user>\AppData\Local\Microsoft\Outlook*.ost	Offline-Storage-Table: enthält E-Mails, Termine, Aufgaben, Notizen und Offline-Postfachdateien
		C:\Users\>user>\Documents\Outlook Files*.ost	
Autostart:			
Geplante Aufgaben / Scheduled Tasks	Registry	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree	Informationen über geplante Aufgaben (in verschiedenen Zeitabständen)
		HKLM\SOFTWARE\Microsoft\Windwos NT\CurrentVersion\Schedule\TaskCache\Tasks	
Benutzer Autostart	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Run	Benutzerspezifische Autostarts, werden

		HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\RunOnce	ausgeführt wenn sich der Benutzer anmeldet
System Autostart	Registry	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	Systemspezifische Autostarts, werden beim Bootvorgang des Systems ausgeführt
		HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce	
		HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\Run	
Dienste	Registry	HKCC\SYSTEM\CurentControlSet\Services	Dienste, die beim Starten des Geräts automatisch ausgeführt werden
USB-Nutzung:			
USBSTOR	Registry	HKCC\SYSTEM\Controlset001\Enum\USBSTOR	Liste aller jemals angeschlossenen USB-Geräte inklusive Namen, Hersteller, Seriennummer
USB	Registry	HKCC\SYSTEM\ControlSet001\Enum\USB	
EMDMgmt	Registry	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\EMDMgmt	Volume Serial Number des Dateisystems auf dem USB-Gerät (Nicht die Seriennummer!!)
Devices	Registry	HKLM\SOFTWARE\Microsoft\Windows Portable Devices\Devices	Letztes USB-Gerät was zu spezifischem Laufwerkbuchstaben zugeordnet werden kann
MountPoints2	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2	Angemeldeter Benutzer zurzeit als ein USB-Gerät angeschlossen wurde
Mounted Devices	Registry	HKCC\SYSTEM\Mounted Devices	Datenbank, welche Seriennummer eines USB-Geräts einem Laufwerkbuchstaben/Volume zuordnet
USB-Spuren in Eventlogs	Eventlogs	C:\System32\winevt\logs\System.evtx Event ID: 20001	Installierte Plug & Play Treiber und ihre Zeitstempel sowie Informationen über das Gerät
Programmausführung:			
Prefetch	Dateisystem	C:\Windows\Prefetch	Lädt Codepages vor (Erhöhung der Performance) gibt Einblick über ausgeführte Programme
UserAssist	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{F4E57C4B-*}\Count	Vom Desktop aus gestartete grafische Programme (.lnk-Dateien)
		HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-*}\Count	Vom Desktop aus gestartete grafische Programme (.exe-Dateien)
Jumplists	Dateisystem	C:\User\<user>\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations	Werden vom Betriebssystem erstellt, um zu Elementen zu springen
SRUM	Dateisystem	C:\Windows\System32\SRU\SRUDB.dat	Zeichnet die Systemperformance auf

BAM	Registry	HKCC\SYSTEM\CurrentControlSet\Services\bam\UserSettings\<SID>	Hintergrundaktivitäten in Windows
RunMRU	Registry	HKCU\NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU oder Policies\RunMRU	Zuletzt ausgeführte Befehle
Powershell	Dateisystem	C:\User\<user>\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history. txt	Zuletzt ausgeführte Befehle in der Powershell
AppCompatCache (Shimcache)	Registry	HKCC\SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatCache\AppCompatCache	Identifizierung möglicher Kompatibilitätsproblemen bei .exe-Dateien
AmCache	Dateisystem	C:\Windows\AppCompat\Programs\Amcache.hve	Informationen über Ausführung/Installation von Anwendungen
Netzwerk:			
Network Interfaces	Registry	HKCC\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces	Informationen über IP-Adresse und Gateway
Network History	Registry	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Managed oder …\Signatures\Unmanaged	Identifizierung von Netzwerken mit welchen der Computer verbunden war
NetworList\Profiles	Registry	HKLM\SOFTWARE\Microsoft NT\CurrentVersion\NetworkList\Profiles	Erste und letzte Verbindung (in lokaler Zeit)
Eventlogs:			
Security.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Security.evtx (Event-ID: 4624,4625,4672,4720)	Konten, deren Anmeldeversuche (Zeitstempel, Art der Anmeldung), neu angelegte Benutzer
RDP: Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-RDPClient%4Operational.evtx (Event-ID: 1024,1102)	Ausgehender Remote-Access (Ziel-Hostname und Ziel-IP-Adresse)
RDP: Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx (Event-ID: 21,22,25,41)	Eingehender Remote-Access (Quell-IP-Adresse, Anmeldename des Benutzers)
RDP: Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational.evtx (Event-ID: 131,98)	Eingehender Remote-Access (Verbindungsversuche und erfolgreiche Verbindungen)
RDP: Microsoft-WindowsTerminalServices-RemoteConnectionManager%4Operational.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-WindowsTerminalServices-RemoteConnectionManager%4Operational.evtx (Event-ID: 1149)	Eingehender Remote-Access (Quell-IP-Adresse, Anmeldename des Benutzers)
SMB: Microsoft-Windows-SmbClient%4Security.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-SmbClient%4Security.evtx (Event-ID: 31001)	Ausgehende Map Network Shares (fehlgeschlagene Logons + Benutzername sowie Fehlercode)
SMB: Microsoft-	Eventlog	C:\Windows\System32\winevt\Logs\Microsoft	Ausgehende Map

Windows-SmbClient%4Connectivity.evtx	s	-Windows-SmbClient%4Connectivity.evtx (Event-ID: 30803, 30800,30804, 30816)	Network Shares (Informationen über Netzwerkverbindungen)
SMB: Microsoft-Windows-SMBServer%4Operational.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Operational.evtx (Event-ID: 1001,1003,1004)	Informationen zur Überwachung des Windows Server Message Block (SMB) Servers
SMB: Microsoft-Windows-SMBServer%4Security.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-SMBServer%4Security.evtx (Event-ID: 551,1006,1009)	Fehlgeschlagene Authentifizierungen, Unauthorisierte Zugriffsversuche
WMI: Microsoft-Windows-WMI-Activity%4Operational.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-WMI-Activity%4Operational.evtx (Event-ID: 5857, 5860, 5861)	Registrierung von temporären und permanenten Events
PS: Microsoft-Windows-PowerShell%4Operational.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-PowerShell%4Operational.evtx (Event-ID: 4103,4104, 53504)	Eingehendes PowerShell-Remoting (Skriptblock-Logging, speichert auffällige Skripte)
PS: Windows PowerShell.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Windows PowerShell.evtx (Event-ID: 400,403,800)	Eingehendes PowerShell-Remoting (Start und Ende einer Remote Session + Skriptcode)
PS: Microsoft-Windows-WinRM%4Operational.evtx	Eventlogs	C:\Windows\System32\winevt\Logs\Microsoft-Windows-WinRM%4Operational.evtx (Event-ID: 6, 8, 15, 16, 33, 91,168)	Ein- und Ausgehendes PowerShell-Remoting (WSMan Session, Benutzer)

[13][16][17][20][21][27][28][45][46][47][54][55]

Anlagen, Tabelle 2 – EZ-Toolsammlung

Toolname	**Funktion**
AmcacheParser	Amcache.hve Parser mit vielen zusätzlichen Funktionen
AppCompatCacheParser	AppCpmpatCache / ShimCache Parser
EvtxECmd	Eventlog Parser mit standardisierter CSV-Ausgabe
JLECmd	Jumplist Parser
LECmd	Parser für LNK-Dateien
MFTECmd	Parst die $MFT, $Boot, $J und weitere Dateien
PECmd	Parst Prefetch-Dateien
RBCmd	Parst Recycle Bin Artefakt (Papierkorb-Einträge)
RECmd	Befehlszeilensuche für die Registry und ihre Hives
RLA	Fügt Transaction-Logs zusammen, damit die Hives sauber sind
SBECmd	Exportiert und parst Shellbag-Daten
Shellbags Explorer	Grafische Nutzeroberfläche für Shellbag-Daten
SrumECmd	Verarbeitet SRUB.dat und optional den Software-Hive für Netzwerk-, Prozess-, und Energieinformationen
Timeline Explorer	Kann CSV- und Exceldateien anschauen, gruppieren, filtern und sortieren

[38][39]

Anlagen, Tabelle 3 – Artefakte zu Szenarios

Artefakte	Speicherort	Daten-diebstahl	Malware analyse	Angriff von aussen	Verwendetes Tool
Windows Registry:					
SYSTEM	Dateisystem	✓	✓	✓	Kann nur durch Targets gesichert werden
SOFTWARE	Dateisystem	✓	✓	✓	“ “
SAM	Dateisystem	✓	✓	✓	“ “
NTUSER.dat	Dateisystem	✓	✓	✓	“ “
USRCLASS.dat	Dateisystem	✓	✓	✓	“ “
Systemanalyse:					
$MFT	Dateisystem	✓	✓	✓	MFTECmd
CurrentVersion	Registry	✓	✓	✓	RECmd
ComputerName	Registry	✓	✓	✓	RECmd
TimeZoneInformation	Registry	✓	✓	✓	RECmd
Users	Registry	✓	✓	✓	RECmd
Dateiöffnungen und -erstellung:					
WordWheelQuery	Registry	✓			RECmd
Shellbags	Registry	✓	✓		SBECmd
ComDlg32: LastVisitedMRU	Registry	✓	✓		RECmd
ComDlg32: Open/SaveMRU	Registry	✓	✓		RECmd
RecentDocs	Registry	✓	✓		RECmd
Office User MRU	Registry	✓	✓	✓	RECmd
Shortcuts\LNK-Files	Dateisystem	✓	✓	✓	LECmd
Dateizugriffe in Browserhistory	Registry	✓		✓	RECmd
Dateizugriffe in OAlerts	Eventlogs	✓	✓	✓	EvtxECmd
Typed Paths	Registry	✓	✓	✓	RECmd
Thumbcache	Dateisystem	✓			thumbcache_viewer_cmd.exe
Recycle Bin	Dateisystem	✓	✓	✓	RBECmd
Webzugriffe	Dateisystem	✓	✓	✓	browsinghistoryview.exe
E-Mail-Artefakte	Dateisystem	✓	✓	✓	kann nur durch Targets gesichert werden
Autostart:					
Geplante Aufgaben	Registry		✓	✓	RECmd
Benutzer Autostart	Registry		✓	✓	RECmd
System Autostart	Registry		✓	✓	RECmd
Dienste	Registry		✓	✓	RECmd

USB-Nutzung:					
USBSTOR	Registry	✓			RECmd
USB	Registry	✓			REcmd
Mounted Devices	Registry	✓			RECmd
Devices	Registry	✓			RECmd
EMDMgmt	Registry	✓			RECmd
MountPoints2	Registry	✓			RECmd
USB-Spuren in Eventlogs	Eventlogs	✓			EvtxECmd
Programmausführung:					
Prefetech	Dateisystem		✓	✓	PECmd
UserAssist	Registry	✓	✓	✓	RECmd
Jumplists	Dateisystem	✓	✓	✓	JLECmd
SRUM	Dateisystem		✓		RECmd
BAM	Registry		✓		RECmd
RunMRU	Registry		✓	✓	RECmd
Powershell (Console History)	Dateisystem		✓	✓	Move-KAPEConsoleHost_History.ps1
AppCompatCache	Dateisystem	✓	✓	✓	AppCompatCache Parser
AmCache	Dateisystem	✓	✓	✓	AmcacheParser
Netzwerke:					
Network Interfaces	Registry			✓	RECmd
Network History	Registry			✓	RECmd
NetworkList\Profiles	Registry			✓	RECmd
Eventlogs:					
Security.evtx	Eventlogs	✓	✓	✓	EvtxECmd
RDP: Microsoft-Windows-TerminalServices-RDPClient%4Operational	Eventlogs			✓	EvtxECmd
RDP: Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational	Eventlogs			✓	EvtxECmd
RDP: Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational	Eventlogs			✓	EvtxECmd
RDP: Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational	Eventlogs			✓	EvtxECmd
SMB: Microsoft-Windows-SmbClient%4Security	Eventlogs	✓		✓	EvtxECmd
SMB: Microsoft-Windows-SmbClient%4Connectivity	Eventlogs	✓		✓	EvtxECmd
SMB: Microsoft-Windows-SMBServer%4Operational	Eventlogs			✓	EvtxECmd
SMB: Microsoft-Windows-SMBServer%4Security	Eventlogs			✓	EvtxECmd
WMI: Microsoft-Windows-	Eventlogs		✓	✓	EvtxECmd

WMI-Activity%4Operational					
PS: Microsoft-Windows-PowerShell%4Operational	Eventlogs		✓	✓	EvtxECmd
PS: Windows PowerShell	Eventlogs		✓	✓	EvtxECmd
PS: Microsoft-Windows-WinRM%4Operational	Eventlogs		✓	✓	EvtxECmd

[39][63][64][65]

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

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

Hamburg, den 29.09.2022

Cécile Hofmann