

Vergleich und Analyse des privaten Modus verschiedener Browser

Computer-Forensik und Vorfallsbehandlung

Carl Schünemann

Christoph Sell

29.08.2025

Inhaltsverzeichnis

1	Einleitung	1
2	Theoretischer Hintergrund	4
2.1	Private Browsing	4
2.2	Angreifermodell	6
2.3	Private Browsing Artefakte	8
3	Ziel der Arbeit	13
4	Methodik	16
4.1	Vorbereitung	16
4.1.1	Browserauswahl	17
4.1.2	Browsing Szenario	17
4.1.3	Analysewerkzeuge	18
4.2	Datensammlung	20
4.3	Datenanalyse	22
4.3.1	Common Locations	23
4.3.2	Uncommon Locations	26
4.3.3	Registry	29
5	Ergebnisse	31
5.1	Firefox	31
5.2	Tor	45
5.3	Chrome	53
5.4	Brave	53
6	Vergleich der Browser	54
7	Diskussion	55
8	Fazit	56
	Appendices	57
	Literaturverzeichnis	60
	Literatur	60

1 Einleitung

Steigende Beliebtheit private Browsing: [11] ■ Die Verwendung von PB wurde als die beliebteste Form der Online-Privatsphäre weltweit identifiziert. ■ Aufgrund der gestiegenen Sensibilität und Öffentlichkeit für den Schutz der Privatsphäre und die Regulierung des eigenen digitalen Fußabdrucks im Internet werden PB-Technologien wahrscheinlich häufiger auf den Geräten der Nutzer eingesetzt. ■ Auch wenn es schwierig ist, endgültige Nutzungsstatistiken für solche Aktionen zu erstellen, bietet der Konsens über den Online-Datenschutz einen Einblick. Im Jahr 2016 wurde die Verwendung eines PB-Fensters als die weltweit beliebteste Form der Online-Datenschutzmaßnahme identifiziert [1]. Allein in den USA nutzen Berichten zufolge rund 33 % der Nutzer ein PB-Fenster, wobei über 70 % zugeben, ihren Internetverlauf zu löschen [2]. - Eine umfassende Studie von Montasari und Peltola (2015) ergab, dass der Erfolg des privaten Modus bei verschiedenen Browsern sehr unterschiedlich ist

Vermeintliche Privatheit beim Browsen: [19] > Verschlüsselung ■ Datenschutz und Datenverwendung sind Hauptbedenken der Internetnutzer geworden [5]. ■ Fragen wie welche Daten von Unternehmen genutzt werden, mit wem sie geteilt werden und wie wertvoll sie sind, sind heute wichtige Themen. ■ Daher versuchen Benutzer, sich so weit wie möglich zu schützen, insbesondere durch Begrenzung der Datenweitergabe. ■ Lösungen wie Verschlüsselung auf HTTP-Ebene [6] und auf DNS-Ebene [7,8] sind Standard geworden und werden den Großteil des Datenverkehrs in den nächsten Jahren abdecken. ■ Sie können jedoch nur End-to-End-Konversationen verschlüsseln, d.h. IP- und TCP- oder UDP-Informationen sind immer noch verfügbar. > VPNs ■ Eine weitere beliebte Methode zum Schutz der Privatsphäre und zur Vermeidung von Datenverwendung ist die Verwendung von Virtual Private Networks (VPNs). ■ Obwohl VPNs immer beliebter geworden sind und die meisten von ihnen den IP-Verkehr verschlüsseln und tunneln können, kann der Datenverkehr tatsächlich am Endpunkt des VPNs überwacht werden. ■ Dies bedeutet, dass Akteure zwischen dem VPN-Servernetzwerk und dem Website-Server die Daten sehen und nutzen können. ■ Der VPN-Anbieter kann sogar noch weiter gehen, da er auch die Identität des Clients kennt. > Tor und Brave: 1. Die Endpunkte der verschlüsselten Verbindungen, die von Tor und Brave hergestellt werden, nicht vollständig verschlüsselt sind. Daher können einige Informationen, wie z.B. die IP-Adresse des Benutzers, an den letzten Servern in der Kette sichtbar sein. 2. Einige Tor-Ausgangsknoten haben in der Vergangenheit die Aktivität ihrer Benutzer ausspioniert, um Daten zu sammeln und möglicherweise zu verkaufen. 3. Obwohl die Verwendung von Brave und Tor dazu beitragen kann, dass Benutzer online nicht nachverfolgt werden, werden sie nicht vor Verfolgung durch andere Methoden wie Standortverfolgung oder Geräte-Fingerprinting geschützt. 4. Schließlich können auch andere Schwachstellen in der Implementierung oder Konfiguration von Tor oder Brave dazu führen, dass Daten durchsickern und somit die Privatsphäre der Benutzer kompromittiert wird.

Immer mehr Kriminelle im Internet [13]: > Das Internet und seine Nutzer wachsen ständig, aber auch die Anzahl organisierter Verbrechen und illegale Aktivitäten nehmen zu.

“Webbrowser immer beliebter bla bla ...“ [12] > Webbrowser sind heutzutage ein wichtiger Werkzeug für Online-Aktivitäten wie Online-Banking, Online-Shopping und soziale Netzwerke.

Immer mehr Internet-Nutzer:[12] ■ Im Jahr 2019 gab es laut [13] fast 4,5 Milliarden Internetnutzer.

Zunehmende Bestrebungen nach Privatheit erschwert forensische Ermittlungen [16] > Zunehmende Verwendung von verschlüsselten Daten in der Dateispeicherung und Netzwerkkommunikation erschwert Ermittlungen. > Besonders schwierig ist das Tor-Protokoll, das sich auf den Schutz der Privatsphäre des Nutzers konzentriert. > Tor-Browser hinterlässt digitale Artefakte, die von Ermittlern genutzt werden können.

Motivation Portable Browser [8] ■ Die Beliebtheit von tragbaren Webbrowsern nimmt aufgrund ihrer bequemen und kompakten Natur sowie des Vorteils, dass Daten einfach über einen USB-Stick gespeichert und übertragen werden können, zu. ■ Entwickler arbeiten an Webbrowsern, die tragbar sind und zusätzliche Sicherheitsfunktionen wie den privaten Modus-Browsing, eingebaute Werbeblocker usw. bieten. ■ Die erhöhte Wahrscheinlichkeit, tragbare Webbrowser für schädliche Aktivitäten zu nutzen, ist das Ergebnis von Cyberkriminellen, die der Ansicht sind, dass bei der Verwendung von tragbaren Webbrowsern im privaten Modus keine digitalen Fußabdrücke hinterlassen werden. ■ Das Forschungspapier zielt darauf ab, eine vergleichende Studie von vier tragbaren Webbrowsern, nämlich Brave, TOR, Vivaldi und Maxthon, zusammen mit verschiedenen Speichererfassungstools durchzuführen, um die Menge und Qualität der aus dem Speicherauszug wiederhergestellten Daten in zwei verschiedenen Bedingungen zu verstehen, nämlich wenn die Browser-Tabs geöffnet und geschlossen waren, um forensische Ermittler zu unterstützen.

Private Browsing Motivation und Ausnutzen von Kriminellen: [15] ■ Webbrowser werden täglich genutzt, um verschiedene Online-Aktivitäten durchzuführen. ■ Webbrowser speichern eine große Menge an Daten über Benutzeraktivitäten, einschließlich besuchter URLs, Suchbegriffen und Cookies. ■ Private Browsing-Modi wurden entwickelt, um Benutzern das Surfen im Internet zu ermöglichen, ohne Spuren zu hinterlassen. ■ Dies kann von Kriminellen ausgenutzt werden, um ihre Aktivitäten zu verschleiern. ■ Experimente werden auf jeder Browser-Modus durchgeführt, um zu untersuchen, ob sie Spuren auf der Festplatte oder im Arbeitsspeicher hinterlassen.

Motivation Private Browsing mit Portablen Browsern: [18] ■ Das Internet ist ein unverzichtbares Werkzeug für alltägliche Aufgaben. ■ Neben der üblichen Nutzung wünschen sich Benutzer die Möglichkeit, das Internet auf private Weise zu durchsuchen. ■ Dies kann zu einem Problem führen, wenn private Internetsitzungen vor Computerermittlern verborgen bleiben müssen, die Beweise benötigen. ■ Der Schwerpunkt dieser Forschung liegt darauf, verbleibende Artefakte aus privaten und portablen Browsing-Sitzungen zu entdecken. ■ Diese Artefakte müssen mehr als nur Dateifragmente enthalten und ausreichend sein, um eine positive Verbindung zwischen Benutzer und Sitzung herzustellen. ■ In den letzten 20 Jahren ist das Internet für alltägliche Aufgaben, die mit stationären und mobilen Computergeräten verbunden sind, drastisch unverzichtbar geworden. ■ Benutzer wünschen sich neben der üblichen Internetnutzung auch Privatsphäre und die Möglichkeit, das Internet auf private Weise zu durchsuchen. ■ Aus diesem Grund wurden neue Funktionen für das private Browsen entwickelt, die von allen gängigen Webbrowsern unterstützt werden. ■ Unsere Forschung konzentriert sich auf die Entdeckung von Informationen von lokalen Maschinen, da die meisten Computeruntersuchungen auf der Suche und Beschlagnahme von lokalen Speichergeräten beruhen. ■ Artefakte aus privaten und portablen Browsing-Sitzungen wie Benutzernamen, elektronische Kommunikation, Browsing-Verlauf, Bilder und Videos können für einen Computerermittler signifikante Beweise enthalten. ■ Wir

werden auch flüchtige Daten analysieren, die in einer gängigen Incident-Response-Umgebung verfügbar wären.

Schwachstellen in Browsern, durch die Daten "lecken" [24] ■ Private browsing ist seit 2005 eine beliebte Datenschutzfunktion in allen gängigen Browsern. ■ Laut einer Studie (-> TODO: welche?) leiden alle Browser unter einer Vielzahl von Schwachstellen, von denen viele zuvor nicht bekannt waren. ■ Die Probleme werden hauptsächlich durch eine laxere Kontrolle von Berechtigungen, inkonsistente Implementierungen der zugrunde liegenden SQLite-Datenbank, die Vernachlässigung von Cross-Mode-Interferenzen und eine fehlende Beachtung von Timing-Angriffen verursacht. ■ Alle Angriffe wurden experimentell verifiziert und Gegenmaßnahmen vorgeschlagen.

Private Browsing Motivation und Ausnutzen von Kriminellen [21] ■ Fast alle Aspekte des Lebens nutzen bereits das Internet, um auf das Internet zugreifen zu können, wird ein Webbrowser verwendet. ■ Die Einführung des Internets hat das Leben der Menschen in vielen Bereichen verändert, darunter auch im Bereich der Kriminalität, insbesondere in der Verwendung von Webbrowser-Software für Transaktionen und Prozesse im Internet. ■ Webbrowser speichern normalerweise Informationen wie URL-Verlauf, Suchbegriffe, Passwörter und andere Nutzeraktivitäten. ■ Aus Sicherheitsgründen wurden einige Funktionen von Webbrowsern entwickelt, um den privaten Modus zu ermöglichen. ■ Leider wird diese Funktion von einigen skrupellosen Menschen für kriminelle Aktivitäten durch die Anti-Forensik genutzt, um digitale Beweise in kriminellen Fällen zu minimieren oder zu verhindern.

Auswirkung von Darknet und Tor auf Forensiker [20] ■ Personen, die Inhalte aus dem Darknet abrufen möchten, müssen nicht nur in einem regulären Browser Schlüsselwörter eingeben, sondern müssen es anonym über den TOR-Browser zugreifen, um ihre Identität wie IP-Adresse oder physische Lage zu verbergen. ■ Aufgrund dieser Tatsachen ist es für Strafverfolgungsbehörden oder digitale forensische Experten schwierig, den Ursprung des Datenverkehrs, den Standort oder die Eigentümerschaft eines Computers oder einer Person im Darknet zu lokalisieren. ■ Die Auswirkungen des Darknets traten auf, als das Federal Bureau of Investigation (FBI) im Oktober 2013 die Website Silk Road abschaltete, die ein Online-Schwarzmarkt und der erste moderne Darknet-Markt für den Verkauf illegaler Drogen war. ■ Silk Road war nur über das TOR-Netzwerk zugänglich und vom Mainstream-Web verborgen. ■ Da die meisten Darknet-Sites Transaktionen über anonyme digitale Währungen wie Bitcoin durchführen, die auf kryptografischen Prinzipien basieren, ist es für digitale forensische Experten sehr schwierig, solche Transaktionen zu verfolgen, da Benutzer und Dienste anonym sind. ■ Das Ziel dieser Arbeit besteht darin, digitale forensische Techniken zu diskutieren, um solche Darknet-Verbrechen zu behandeln.

2 Theoretischer Hintergrund

Einleitend werden Struktur, Motivation und die abgeleiteten Forschungsfragen diskutiert.

2.1 Private Browsing

Definition Web Browser: > [21] ■ Der Webbrowser ist eine Softwareanwendung zum Abrufen, Präsentieren und Durchsuchen von Informationsressourcen im Internet oder World Wide Web (WWW).
■ Eine Informationsquelle wird durch einen Uniform Resource Identifier (URI) identifiziert und kann Webseiten, Bilder, Videos oder andere Inhalte enthalten.

> [12] ■ Ein Webbrowser ist eine Software, die es Benutzern ermöglicht, das Internet über den von ihrem Dienstanbieter bereitgestellten Zugang zu nutzen. ■ Die bekanntesten Webbrowser sind Google Chrome, Mozilla Firefox, Microsoft Edge und Brave. ■ Webbrowser werden für alltägliche Aktivitäten wie das Anschauen von Videos, das Durchsuchen von Webseiten, das Posten von Bildern oder Videos in sozialen Medien und das Herunterladen und Hochladen von Dateien genutzt. ■ Browser-Modi: Es gibt zwei verschiedene Browser-Modi: den normalen Browser-Modus und den privaten Browser-Modus.

Definition „Normal Browsing“: > [12] ■ Der normale Browser-Modus speichert alle Browser-Aktivitäten wie Caches, Cookies, Suchbegriffe, Login-Daten und URL-Verlauf auf dem Computer. ■ Die Cookies speichern Details des Benutzers wie z.B. Browsing-Muster, die anzeigen können, welche Websites der Benutzer häufig besucht oder welche Videos er/sie regelmäßig ansieht.

Definition "Private Browsing": > [22] - Deshalb wurde eine neue Funktion in die Webbrowser aufgenommen, die den Internetnutzern eine größere Kontrolle über ihre Privatsphäre ermöglicht. Diese Funktion ist als "Private Browsing" bekannt und soll es den Nutzern ermöglichen, im Internet zu surfen, ohne Datenspuren auf ihrem Computer zu hinterlassen.

> [11] - Private Browsing"(PB) ist ein allgemeiner Begriff, der sich auf Mechanismen, die verhindern sollen, dass ein Nutzer Beweise für sein Web-Browsing-Verhaltens auf seinem lokalen Gerät gespeichert werden. - Von Anfang an muss betont werden, dass sich privates Surfen in diesem Zusammenhang nur auf Plattformen bezieht, die lokale Privatsphäre bieten, und dass diese von Anwendungen wie Tor (siehe <https://www.torproject.org/>) zu unterscheiden sind, die sich ebenfalls auf die Online-Privatsphäre konzentrieren, sowie von Einrichtungen, die die Verfolgung und Überwachung aus der Ferne verhindern, wie z. B. der Tracking Preference Expression des W3C (auch bekannt als "Do Not Track"). - Je nach Browser des Nutzers wird eine zugehörige PB-Funktion mit unterschiedlichen Begriffen bezeichnet: Inkognito-Modus in Chrome, InPrivate in Edge und dem inzwischen nicht mehr unterstützten Internet Explorer sowie ein "privates Fenster" in Firefox.

Geschichte Private Browsing: > [22] - Die ADbC-Funktion "Privater Browsing-Modus" wurde erstmals 2005 mit Apple Safari 2.0 eingeführt. Drei Jahre später folgte Google Chrome 1.0 (Inkognito). Später,

im Jahr 2009, führten Microsoft Internet Explorer 8 und Mozilla Firefox 3.5 ihre Versionen von privaten Browsing-Modi ein, die als InPrivate bzw. Private Browsing bekannt sind (Dan, 2010).

> [15] ■ Private Browsing-Modi haben je nach Browser unterschiedliche Namen, z.B. Incognito-Modus in Chrome, InPrivate Browsing in Internet Explorer, "Private Browsing" in Firefox und Safari. ■ erstmals 2005 von Apple Safari eingeführt, gefolgt von Google Chrome und Microsoft in 2008 und Mozilla in 2009.

Grund des privaten Modus: > [12] ■ Private Browsing Mode wurde entwickelt, um die Privatsphäre und Anonymität beim Surfen im Internet zu verbessern, indem keine Spuren und Informationen von Browsing-Aktivitäten hinterlassen werden. ■ Alle neuen Caches, die während des Surfens gespeichert wurden, werden entfernt, sobald der Browser geschlossen wird. ■ Jeder Webbrowser bietet einen privaten Browser-Modus mit unterschiedlichen Bezeichnungen an, wie InPrivate Browsing für Internet Explorer und Microsoft Edge, Incognito-Modus für Google Chrome und "Private Browsing" für Mozilla Firefox. > [1] zwei wesentliche Ziele des privaten Browsing: 1. (local) Besuchte Websites sollten im privaten Modus keine Spuren auf dem Computer des Benutzers hinterlassen. Wenn ein Familienmitglied den Browserverlauf überprüft, sollte keine Evidenz von im privaten Modus besuchten Websites gefunden werden können. 2. (website) Benutzer möchten möglicherweise ihre Identität vor den Websites, die sie besuchen, verbergen, indem sie es beispielsweise für Websites schwierig machen, die Aktivitäten des Benutzers im privaten Modus mit seinen Aktivitäten im öffentlichen Modus zu verknüpfen. Dies wird als Datenschutz vor einem Webangreifer bezeichnet. > [15] ■ Private Modus Browser sollten in der Lage sein, die von besuchten Websites hinterlassenen Artefakte auf dem Computer des Benutzers zu verhindern. ■ Browser sollten es Websites unmöglich machen, herauszufinden, ob ein bestimmter Benutzer sie zuvor besucht hat, indem sie verhindern, dass Websites die Aktivitäten von Benutzern im privaten und öffentlichen Modus verknüpfen.

Stakeholder Private Browsing: > Forensiker - [13] ■ Die Entwicklung von Datenschutzfunktionen in Browsern stellt eine Herausforderung für digitale Forensiker dar, die Beweismittel sammeln möchten, um Kriminelle zu überführen. - [11] ■ Durch die Möglichkeit des privaten Browsens besteht eine erhöhte Gefahr für illegale und schädliche Online-Aktivitäten. ■ Die meisten privaten Browsing-Modi sind so konzipiert, dass sie lokal privat sind und Daten, die auf das Surfverhalten des Benutzers hinweisen, nicht auf dem Gerät gespeichert werden. ■ Diese Handlungen können die Verfügbarkeit von Beweismaterial beeinträchtigen und stellen eine Herausforderung für Untersuchungen dar. - [11] ■ Private browsing (PB) ist eine Funktion, die seit langem auf dem Radar von forensischen Praktikern steht. ■ Risiko: PB kann dazu führen, dass potenziell beweiskräftiger Inhalt nicht auf einem lokalen Gerät gespeichert wird, was zu Untersuchungsbedenken führt. ■ PB selbst hat viele legitime Anwendungen und ist nicht per se anti-forensisch, kann aber mit anti-forensischer Absicht verwendet werden. ■ Fehlende Internetinhalte stellen ein Problem für Beweissammlung ■ Private Browsing-Modi sollten die Aktivität des Nutzers vor forensischen Tools verbergen

> Kriminelle: - [13] ■ Kriminelle nutzen vermehrt private Browser, um ihre Spuren zu verwischen und ihre illegalen Handlungen zu verbergen. ■ Cyberkriminelle nutzen Private Browsing-Modi, um digitale Spuren auf dem Gerät zu verwischen und forensische Ermittler mit leeren Händen dastehen zu lassen. > Nutzerperspektive: - [11] ■ Die Verwendung von PB wurde als die beliebteste Form der Online-Privatsphäre weltweit identifiziert. ■ Aufgrund der gestiegenen Sensibilität und Öffentlichkeit für den Schutz der Privatsphäre und die Regulierung des eigenen digitalen Fußabdrucks im Internet werden PB-Technologien wahrscheinlich häufiger auf den Geräten der Nutzer eingesetzt. ■ Auch wenn es schwierig ist, endgültige Nutzungsstatistiken für solche Aktionen zu erstellen, bietet der Konsens

über den Online-Datenschutz einen Einblick. Im Jahr 2016 wurde die Verwendung eines PB-Fensters als die weltweit beliebteste Form der Online-Datenschutzmaßnahme identifiziert [1]. Allein in den USA nutzen Berichten zufolge rund 33 % der Nutzer ein PB-Fenster, wobei über 70 % zugeben, ihren Internetverlauf zu löschen [2].

- [11] ■ Die PB-Technologie wird aufgrund der gesteigerten Sensibilität und öffentlichen Aufmerksamkeit für den Schutz der Privatsphäre voraussichtlich häufiger auf Geräten verwendet. - [22] In den letzten Jahren (2010) haben jedoch viele der bekannten Webbrowser-Hersteller ihre Besorgnis über die Privatsphäre der Nutzer beim Surfen im Internet verstärkt. - Tatsächliche Gründe: [15] in [1] Experiment von Aggarwal et al.: Werbung auf Ad-Netzwerken geschaltet wurde, um verschiedene Kategorien von Websites einschließlich Erwachsenen- und Geschenk-Websites zu bewerben, um die Nutzung des privaten Modus mit der Art der besuchten Website zu korrelieren. -> Browsing-Modus auf Erwachsenen-Websites beliebter war als auf Geschenk-Websites. > Herstellerperspektive: - [15] Angeblich lt. Hersteller: o Einkaufen von Überraschungsgeschenken auf einem Familien-PC o Planung von Überraschungspartys

Stakeholder Private Browsing: > "Forensischer Ermittler [15] ■ forensischer Ermittler kann forensische Browsing-Artefakte mit forensischen Tools und Techniken wiederherstellen > "Nutzer": - [22] > Tatsächlich ergab eine Studie, dass Private Browsing auf Websites für Erwachsene beliebter ist als auf Websites für den Geschenkekauf oder für Nachrichten. Dies deutet darauf hin, dass die Anbieter von Webbrowsern den Hauptnutzen dieses Tools möglicherweise falsch einschätzen, wenn sie es als ein Tool zum Kauf von Überraschungsgeschenken beschreiben (Aggarwal, Boneh, Bursztein, und Jackson, 2010). > "Browser Entwickler [13] Die Entwickler von Browsern haben den Mangel an Benutzerdatenschutz erkannt und einen privaten Browsermodus eingeführt, der das Schreiben von Browserdaten auf die Festplatte einschränkt oder idealerweise verhindert. - [22] Einem Artikel zufolge (Belani, Jones, 2005) behaupten die Hersteller all dieser Webbrowser, dass keine der besuchten Websites, Formularfelddaten, in die Adressleiste eingegebenen Adressen, besuchten Links und Suchanfragen auf dem lokalen Computer des Nutzers gespeichert werden (Brookman, 2010).

2.2 Angreifermodell

Definition Local Attacker nach [1]: - Z.B. Forensischer Prüfer - hat physischen Zugriff auf den Computer des Benutzers - versucht, auf dessen privaten Browserverlauf zuzugreifen. - beispielsweise ein Familienmitglied oder ein Freund sein, der den Computer des Benutzers nutzt, um auf dessen Browserverlauf zuzugreifen. - kann darauf installierte Programme verwenden, um Informationen zu sammeln. - hat keinen Zugriff auf die Maschine des Benutzers, bevor der Benutzer das private Surfen beendet hat. Ohne diese Einschränkung ist Sicherheit gegen einen lokalen Angreifer unmöglich. (z.B: Keylogger installieren, Benutzeraktionen aufzeichnen) - Durch die Beschränkung des lokalen Angreifers auf "forensische Untersuchungen nach dem Ereignis" kann man hoffen, Sicherheit zu gewährleisten, indem der Browser persistenten Zustandsänderungen während einer privaten Surfsitzung ausreichend löscht. - Der Angreifer wartet, bis der Benutzer den privaten Browsing-Modus verlässt, und erhält dann die vollständige Kontrolle über die Maschine. Dies bedeutet, dass der Angreifer auf forensische Daten angewiesen ist. - Während der aktiven Phase kann der Angreifer nicht mit Netzwerkelementen kommunizieren, die Informationen über die Aktivitäten des Benutzers im privaten Modus enthalten. Dies bedeutet, dass die Implementierung von Browser-seitigen Datenschutzmodi untersucht wird, nicht die serverseitigen Datenschutzmodi.

- Das Ziel des Angreifers besteht darin, für eine bestimmte Menge von HTTP-Anfragen, die er wählt, festzustellen, ob der Browser eine dieser Anfragen im privaten Browsing-Modus ausgeführt hat oder nicht. Wenn der lokale Angreifer dieses Ziel nicht erreichen kann, gilt die Implementierung des privaten Browsers als sicher. -> Local Attacker weiß, wonach er sucht!
- Es wird darauf hingewiesen, dass die Definition impliziert, dass der Angreifer nicht feststellen kann, welche Websites der Benutzer besucht hat oder was der Benutzer auf einer bestimmten Website getan hat. Darüber hinaus wird auf die Eigenschaften des privaten Browsers nicht formal eingegangen, wenn der Benutzer den privaten Browsing-Modus nie verlässt.

Problem: Local Attacker muss überarbeitet werden: [15] ■ Es wurde festgestellt, dass das Konzept des lokalen Angreifers nicht ausreichend untersucht wurde und dass neue Experimente durchgeführt werden müssen, um ein besseres Verständnis für das Phänomen zu erlangen und herauszufinden, wie sich diese Funktion auf digitale forensische Untersuchungen auswirken könnte.

Definition Web Attacker nach [1] - Z.B. ISP - versucht Online-Aktivitäten des Benutzers im privaten Modus zu verfolgen und zu identifizieren, um diese mit seinen Aktivitäten im öffentlichen Modus in Verbindung zu bringen. - durch den Einsatz von Tracking-Tools oder das Sammeln von Informationen über die IP-Adresse des Benutzers oder andere Identifikationsmerkmale erfolgen. ■ Kontrolliert die von Benutzer besuchten Websites und kann Informationen über Benutzeraktivitäten sammeln (-> z.B. ISP), aber nicht über den Computer des Benutzers. ■ Webseiten können auch verschiedene Browser-Funktionen nutzen, um Browser zu identifizieren und sie über Privatsphäre-Grenzen hinweg zu verfolgen. ■ Die Electronic Frontier Foundation hat eine Website namens Panopticklick (-> TODO: In Demo zeigen?) erstellt, die zeigt, dass die meisten Browser eindeutig identifiziert werden können, was die Ziele (1) und (2) des privaten Surfers in allen Browsern unterbricht.

Anti-Forensische Grundsätze bei Browserentwicklung, um sich gegen Web-Attacker zu schützen nach [1] ■ Browser haben drei Ziele, um die Privatsphäre der Benutzer zu schützen. o Ziel 1: Ein Benutzer, der im privaten Modus surft, soll nicht mit demselben Benutzer verknüpft werden können, der im öffentlichen Modus surft. o Ziel 2: Ein Benutzer in einer privaten Sitzung soll nicht mit demselben Benutzer in einer anderen privaten Sitzung verknüpft werden können. o Ziel 3: Eine Website soll nicht erkennen können, ob der Browser im privaten Modus ist. ■ Ziele (1) und (2) sind schwierig zu erreichen, da die IP-Adresse des Browsers von Webseiten genutzt werden kann, um Benutzer über private Browsing-Grenzen hinweg zu verfolgen. ■ Das Torbutton Firefox-Erweiterung (ein Tor-Client) macht große Anstrengungen, um Ziele (1) und (2) zu erreichen, indem es die IP-Adresse des Clients über das Tor-Netzwerk versteckt und Schritte unternimmt, um das Browser-Fingerprinting zu verhindern.

Beispiel: Web Attacker Angriffe: > IP-Angriffe [19] ■ Obwohl Nutzer Verschlüsselung oder VPNs nutzen, ist ihre Privatsphäre oft ungeschützt, da mehrere Domains gleichzeitig besucht werden oder IP-Adressen von Cloud-Providern geteilt werden. ■ Eine neue Methode zur Identifizierung von Web-Browsing wird vorgestellt, die nur auf den IP-Adressen basiert, mit denen der Nutzer verbunden war, ohne DNS Reverse-Resolution durchzuführen. ■ Diese IP-Adresse-Sequenz wird in verschiedene Deep Learning Modelle eingespeist, um die tatsächlich besuchte Website zu identifizieren. ■ Untersucht wurden auch andere Faktoren wie Abhängigkeit vom DNS-Server, Genauigkeit bei Top-Domains, Datenverstärkung durch Paket-Sampling-Simulation, Auswirkungen auf Paket-Sampling und Skalierbarkeit der Methode. ■ Mit nur 10% der Pakete konnte die besuchte Website mit einer Genauigkeit und F1-Score von 94% bis 95% identifiziert werden. > ISP als „Web attacker“, um Kundenaktivität zu verfolgen [1] ■ ISP können unsere Ergebnisse nutzen, um den Datenverkehr ihrer Kunden zu identifizieren. ■ Dies ermöglicht

ISP, Daten für Marketingzwecke zu monetarisieren, sofern sie anonymisiert und mit Zustimmung der Kunden erfolgt. ■ ISP müssen jedoch darauf achten, wer Zugriff auf Netzwerkverkehrsdaten hat. ■ Das Weitergeben dieser Daten an Dritte kann zu potenziellen Datenschutzverletzungen bei Kunden führen. ■ Hauptaufgabe ist eigentlich einfach, aber es können viele Komplikationen auftreten ■ Hauptproblem ist das sogenannte "verwickelte Netz" ■ Beim Verbinden mit einer Website muss der Webbrowser eine Kaskade von Verbindungen zu anderen Websites öffnen ■ Grund dafür sind Bilder, Anzeigen, Banner, JavaScript-Bibliotheken, Social-Media-Links und vieles mehr

2.3 Private Browsing Artefakte

TODO: Common vs Uncommon Locations hier ansprechen

Residuale Daten > [12] ■ Überraschenderweise besteht der private Browser in Chrome und Firefox aus wenigen residuellen Daten, die jedoch Beweise für Interessen wie Suchbegriffe, E-Mail-IDs und Passwörter liefern können ■ Residuale Daten sind Daten, die von einem Gerät entfernt wurden, aber immer noch aufgespürt werden können. ■ Diese Daten können mithilfe spezieller Tools, meist in Dateiüberresten oder lokalen Ordnern, identifiziert werden. ■ Beispiele für residuale Daten sind Link-Dateien, Log-Dateien, Registry-Dateien, Prefetch-Dateien und Browser-Verlaufsdaten. ■ Digitale Forensik kann solide elektronische Beweise aus solchen Überresten und Artefakten sammeln, um sie in Gerichtsverfahren zu verwenden.

Browser Artefakte: > [12] ■ Jeder Browser hat unterschiedliche Artefakte im RAM des Geräts gespeichert ■ Im normalen Browsing-Modus werden die Browsing-History-Details des Benutzers vor und nach dem Löschen des Verlaufs im Speicher gespeichert ■ Benutzeraktivitäten und Daten beim Surfen können in normalen Browser-Modi wie Cookies, Caches, Downloads, Verlauf, anderen sensiblen Daten und temporären Dateien verfolgt und gespeichert werden, was digitalen Forensikern bei der Suche nach Beweisen hilft.

> [13] ■ Browser speichern eine Vielzahl von Nutzerdaten, die von besuchten URLs bis zu Benutzer-namen und Passwörtern reichen ■ Das Wissen, dass Browser private Surfdaten preisgeben, ist schon etwas, aber der Standort dieser Artefakte ist von größter Bedeutung

> [22] - Webbrowser sind so konzipiert, dass sie eine Vielzahl von Informationen über die Aktivitäten ihrer Benutzer aufzeichnen und speichern können. Dazu gehören Caching-Dateien, besuchte URLs, Suchbegriffe, Cookies und andere. - Diese Dateien werden auf dem lokalen Computer gespeichert und können von jeder Person, die denselben Computer verwendet, leicht aufgerufen und abgerufen werden. Dies macht es auch für forensische Prüfer relativ einfach, die Internet-Aktivitäten eines Verdächtigen in Fällen zu untersuchen, in denen fragwürdige Websites besucht oder kriminelle Handlungen über das Internet durchgeführt wurden.

> [3] ■ Bestimmte Datentypen aus HTTP-Protokoll-Transaktionen oder skriptgesteuerten Aktionen in HTML-Seiten werden separat im Dateisystem gespeichert und führen zu unterschiedlichen Datenbankeinträgen: Cookies, Web Storage und Indexed Database Storage.

Private Browsing Artefakte: > [1] 1. Änderungen, die von einer Website ohne jegliche Benutzerinteraktion initiiert werden. Beispiele hierfür sind das Setzen eines Cookies, das Hinzufügen eines Eintrags zur Verlaufsdatei und das Hinzufügen von Daten zum Browser-Cache. 2. Änderungen, die von einer Website

initiiert werden, aber eine Benutzerinteraktion erfordern. Beispiele hierfür sind das Generieren eines Client-Zertifikats oder das Hinzufügen eines Passworts zur Passwortdatenbank. 3. Änderungen, die vom Benutzer initiiert werden. Zum Beispiel das Erstellen eines Bookmarks oder das Herunterladen einer Datei. 4. Nicht benutzerspezifische Zustandsänderungen, wie das Installieren eines Browser-Patches oder das Aktualisieren der Phishing-Blockierungsliste. ■ "geschützte Aktionen- Browser Artefakt, dass beim Verlassen des privaten Surfens gelöscht werden muss

Wie entstehen "Leckagen" von privaten Browsing Artefakten? [11] 1. Ein Fehler im Design und in der Entwicklung des Browsers 2. Das Betriebssystem übernimmt mehr Kontrolle über den Browser als es sollte, was dazu führt, dass Daten von außen abgegriffen werden

Common Locations: > Ort der Browserartefakte ("common locations") ausführlich beschrieben in: [5]

> [12] ■ Die Artefakte von Webbrowsern werden in bestimmten Ordnern im Betriebssystem gespeichert. ■ Die genaue Lage variiert je nach Browser, die Dateiformate bleiben jedoch gleich. ■ Es ist wichtig zu wissen, wo die Dateien gespeichert sind, um sie während des normalen und privaten Browsing-Modus untersuchen zu können. ■ Tabelle 6 zeigt die Standorte der Artefakte von Google Chrome wie Verlauf, Caches und Cookies. ■ Tabelle 7 stellt die häufigsten Standorte von Firefox-Artefakten wie Cookies, Cache, Verlauf und Lesezeichen vor. ■ Alle Änderungen in Firefox, wie Lesezeichen, installierte Erweiterungen und gespeicherte Passwörter, werden im Profilordner gespeichert. ■ Wie in der Tabelle gezeigt, werden Cookies in cookies.sqlite gespeichert, während Cache-Dateien im cache2-Ordner zu finden sind. ■ Alle heruntergeladenen Lesezeichen, Dateien und der Verlauf werden in places.sqlite gespeichert. ■ Mögliche Informationen, die aus der Browser-Forensik extrahiert werden können, sind Browsing-Verlauf, Cache, Cookies, Lesezeichen und Download-Liste.

> [21] ■ Digitale Beweise in einem Webbrowser umfassen mindestens Caches, Verlauf, Cookies, Download-Dateilisten und Sitzungen. ■ Zumindest ein Minimum an digitalen Beweisen aus einem Webbrowser ist sehr wichtig und wird von Ermittlern genutzt, um einen Fall bei Internetnutzung zu analysieren.

Gründe für Browser-Artefakte bei Private Browsing: [11] > Fehler im Design und Entwicklung des Browsers -> führt dazu, dass Daten von innen nach außen durchsickern, d. h. Browser ist schuld > Betriebssystem übernimmt mehr Kontrolle über den Browser als es sollte, was dazu führt, dass Daten von außen abgegriffen werden, d. h. Betriebssystem ist schuld

Definition private Browsing Artefakt: =====
Strings, die Aktionen des Browsing-Protokolls zugeordnet werden können: Keywords, URLs, HTML-Fragmente, E-Mail-Adressen, Betreffzeilen etc.

Warum Computer-Forensik: [13] ■ Die Untersuchung von digitalen Beweisen ist von großer Bedeutung, um Straftäter zu identifizieren und zur Rechenschaft zu ziehen.

Definition digitale Forensik [12] ■ Digitale Forensik konzentriert sich auf die Wiederherstellung von Speichermedien, um digitale Beweise für Cybercrime-Untersuchungen zu sammeln. ■ Die gewonnenen Beweise müssen jedoch in ihrem Originalzustand erhalten bleiben, um vor Gericht zulässig zu sein. ■ Der Prozess der Erwerbung, Untersuchung, Analyse und Berichterstattung von digitalen Beweisen muss forensisch einwandfrei durchgeführt werden. ■ Daher müssen Ermittlungsteams sich an die Phasen der digitalen Forensik halten, die auf weit verbreiteten Standards basieren. ■ Digitale Forensik-Investigatoren verlassen sich auf die Artefakte, die aus diesen Browser-Records auf dem Gerät zurückbleiben, und

verwenden forensische Techniken, um die Artefakte zu erfassen, um Beweismittel zu finden. ■ Die Artefakte werden im Computer-Speicher gespeichert, nachdem alle Browser-Verläufe, Caches und Cookies gelöscht wurden, was es für digitale forensische Gutachter einfach macht, die Daten zu extrahieren.

Definition Browser Forensics > [13] ■ Web-Browser-Forensik sammelt und identifiziert Beweise und Informationen im Zusammenhang mit einem Verbrechen aus wiederhergestellten Browser-Sitzungen - Forensische Analyse des Webbrowsers beinhaltet die Wiederherstellung von Browsing-Artefakten, die Informationen über die Online-Aktivitäten eines Verdächtigen offenbaren. - Browser-Forensik wird für Ermittler immer wichtiger, da Suchverlauf, Download-Aktivität und Seitenaufrufe das Verständnis für das kriminelle Motiv verbessern können.

Ziel digitale Forensik [12] ■ Digitale Forensik hat das Ziel, verwendbare Beweise für Computerkriminalität zu sammeln. ■ Cyberkriminalität wie Hacking, betrügerische Transaktionen und Diebstahl geistigen Eigentums erhöhen den Bedarf an digitaler Forensik, um auf Cyberkriminalität mit einem digitalen Gerät zu reagieren. (2022) A Comparative Analysis of Residual Data

Live-Forensik: unterschiedliche Definitionen in Literatur > Live-Forensik als "moderner Trend" der Computer-Forensik [7] Im Gegensatz zur traditionellen (toten) digitalen Forensik wird bei der Live-Forensik versucht, flüchtige Daten aufzubewahren und Gegenmaßnahmen für verschlüsselte Dateien auf einem Live-System zu ergreifen. > [9]: TODO! > [12] ■ „Live Forensics“ wird auch als „Live System Acquisition“ bezeichnet. ■ Diese Methode wird angewendet, wenn das System in Betrieb ist, um potenzielle Artefakte im flüchtigen Arbeitsspeicher (RAM) zu finden, die als Beweismittel genutzt werden können. ■ Viele Spuren von Computer-Sitzungen und Artefakte sind nur im flüchtigen Speicher zu finden und können nicht von externem Speicher aus ausgelesen werden. ■ Die Daten können jedoch nicht gesammelt werden, da sie verloren gehen, sobald das System gestoppt oder neu gestartet wird. ■ Die RAM-Daten müssen daher mit besonderen Verfahren behandelt werden, um ihre Integrität und Zuverlässigkeit während der Analyse zu gewährleisten. ■ „Live Forensics“ ist nützlich, um auch Ereignisse zu untersuchen, die nur während der Nutzung des Systems aufgetreten sind, und um Daten effizient im flüchtigen Arbeitsspeicher zu speichern. ■ Digitale Forensik kann dazu genutzt werden, die Gültigkeit von Beweismitteln bei Gerichtsverfahren zu untersuchen. ■ Nach der Identifikation und Sammlung von potenziellen Beweismitteln wird in den meisten Fällen eine exakte Kopie der Daten erstellt, um sie als Backup zu nutzen und Veränderungen zu vermeiden. ■ Es gibt zwei Arten von forensischen Techniken, um Speicherabbilder zu erstellen: „Dead Forensics“ und „Live Forensics“. ■ Bei „Live Forensics“ hingegen wird das System im laufenden Betrieb untersucht, was oft schwieriger ist, aber auch wertvolle Informationen liefern kann. > [21] ■ Forensische Untersuchung eines Systems, während es in Betrieb ist ■ Daten gehen verloren, wenn das System heruntergefahren oder neu gestartet wird ■ Verwendung bei flüchtigem Speicher wie RAM ■ RAM-Erfassung durch RAM-Forensik-Tool ■ Ziel ist es, den normalen Betrieb des Systems nicht zu beeinträchtigen ■ Live Forensics liefert wichtige Informationen für die Analyse ■ Analyse von digitalen Beweisen aus dem RAM mit Memory Analysis Tool. ■ Eine Lösung für dieses Problem ist die Live-Forensik, um Daten aus dem Arbeitsspeicher zu extrahieren, bevor sie gelöscht werden. ■ Diese Forschungsmethode wird verwendet, um Webbrowser im Allgemeinen und insbesondere tragbare Webbrowser zu analysieren.

Beispiele Live-Forensik > [24] ■ Volatiler Speicher (Memory Inspection) kann eine wichtige Informationsquelle für forensische Untersuchungen sein ■ DNS-Caching ist eine Bedrohung für private Browsing: Diese Schwachstelle entsteht, weil das Betriebssystem DNS-Anfragen des Browsers im Cache speichert, unabhängig davon, ob der Browser im privaten Modus ist oder nicht > [13] ■ Registry

Snapshots: Um Veränderungen im System-Registry aufgrund der Browserinstallation zu verfolgen, wurde Regshot verwendet, um vor der Installation einen Snapshot der Registry aufzunehmen. - Ein zweiter Snapshot wurde nach der Installation des Browsers aufgenommen und mit dem ersten verglichen. - Regshot generiert einen Bericht über die Ergebnisse, der die neuen Dateien und Ordner zeigt, die dem Registry-Schlüssel hinzugefügt wurden.

Vorteile Live-Forensik > In Literatur bekannt: Die meisten Informationen im RAM > [11] ■ Die meisten Daten können in den RAM-Speichergeräten des Betriebssystems gefunden werden. > [16] ■ Da es wahrscheinlich ist, dass RAM-Aufnahmen Inhalte der Browsing-Session (z.B. durch Caching) aufzeigen, wurde dies in das Projekt aufgenommen, insbesondere da Warren (2017) dies aufgrund von Zeitbeschränkungen nicht tun konnte. > [16] ■ Live Analyse während der Laufzeit einer Anwendung ist besonders vorteilhaft, um zu verstehen, wie das Betriebssystem und die Anwendung interagieren. ■ Live Analyse kann potenziell mehr Informationen zur Browsing-Session liefern, da die Designbemühungen des Tor-Projekts darauf abzielten, Schreibzugriffe auf die Festplatte zu vermeiden.

Herausforderungen von Live-Forensik = Kontaminieren von Beweismitteln [7] Die größten Herausforderungen während des Datenerfassungsprozesses sind: Datenveränderung und Abhängigkeit vom Betriebssystem des verdächtigen Systems; wenn der Erfassungsprozess die Daten verändert, werden die Gerichte die Daten als forensisch untauglich abweisen

Definition Dead Forensik: > [12] ■ Bei „Dead Forensics“ wird der Computer oder das Gerät, das untersucht werden soll, zuerst heruntergefahren, bevor das Speicherabbild erstellt wird. > [11] - Physische Speichererfassung ist nicht übliche Praxis und in den meisten Fällen nicht verfügbar > [9]: TODO! > [13] - Oft einzige Option: Analysen von Festplatten-Images von ausgeschalteten Geräten - Gründe für „einzige Option“: o Verzögerungen bei der Bearbeitung o Personalmangel bei den forensischen Untersuchern - also unrealistisch und unpraktisch, beschlagnahmte Geräte eingeschaltet zu lassen. - Ausschalten eines Geräts reduziert Risiko einer Datenänderung (versehentlich oder absichtlich) - isoliert es vom Netzwerk, um etwaige Versuche, es ferngesteuert zu löschen, zu verhindern, unter anderem. > [12] -> widersprüchlich? ■ System wird heruntergefahren, bevor das Speicherabbild erstellt wird. ■ Volatile Dateien gehen verloren: versteckte Dateien, ausgetauschte Dateien, Web-Aktivitäten, Artefakte und Log-Dateien ■ Das Ziel ist es, eine genaue Kopie des nichtflüchtigen Speichers zu erstellen, bevor das System heruntergefahren wird, um die Originalität der Beweismittel zu erhalten.

Beispiele Dead Forensik: > Stichwortsuche in Festplatten-Images nach herunterfahren [24] > Timestamps von Dateien [24] > SQLite Datenbanken [24] > Unallocated Space [24] > Registry-Hives auf Festplatte, z.B. NTUSER.DAT [24]

Probleme bei Dead Forensik > [7]: TODO!

Wann Live-, wann Dead Forensik? [12] ■ Die Wahl der Methode hängt von der Art der Untersuchung und der verfügbaren Zeit ab. ■ Das Ziel ist es, eine genaue Kopie des Speichers zu erstellen, um die Integrität der Beweise zu bewahren und das Risiko von Veränderungen zu minimieren.

Definition: Darknet Forensik: [20] ■ Motivation Darknet Forensik: o Terroristen, Kriminelle, extremistische Gruppen und Hassorganisationen nutzen das Darknet, um Cybercrime zu begehen. o Die Verwendung von TOR und Bitcoin auf dem Darknet erschwert die Verfolgung von Straftaten durch digitale Forensik-Experten. o Die vorgeschlagenen forensischen Techniken können digitale Forensik-Experten helfen, mit Cybercrime-Fällen im Zusammenhang mit dem Darknet umzugehen. ■

Darknet-Forensik sind in zwei Kategorien unterteilt: 1. TOR-Browser-Forensik: ■ vier Möglichkeiten zur Extraktion von TOR-Browser Artefakten: RAM-Forensik, Registry-Änderungen, Netzwerk-Forensik und Datenbank 2. Bitcoin-Transaktions-Forensik: Extrahieren von forensischen Artefakten aus Bitcoin-Wallet-Anwendung

3 Ziel der Arbeit

Wichtig: White-Box Ansatz gemäß local Attacker in [1] - Das Ziel des Angreifers besteht darin, für eine bestimmte Menge von HTTP-Anfragen, die er wählt, festzustellen, ob der Browser eine dieser Anfragen im privaten Browsing-Modus ausgeführt hat oder nicht. Wenn der lokale Angreifer dieses Ziel nicht erreichen kann, gilt die Implementierung des privaten Browsers als sicher. - Local Attacker weiß, wonach er sucht!

Forensiker müssen Funktionsweise von Private Browsing kennen [11] ■ Die Kenntnis der Erfolgsrate der PB-Technologie unterstützt die Strafverfolgungsbehörden bei digitalen Untersuchungen von Internetinhalten ■ Internetbeweise sind oft entscheidend für Untersuchungen ■ Bestimmung des Umfangs und des Erfolgs von PB-Technologie unterstützt die Strafverfolgungsbehörden bei digitalen Untersuchungen von Internetinhalten ■ Durch die Bestimmung des Umfangs und des Erfolgs von PB-Technologie können sie unnötige Datenverarbeitung und Zeitverschwendung vermeiden, die Untersuchungseffizienz verbessern und sicherstellen, dass keine wichtigen Inhalte übersehen werden. Daher können diese Punkte dazu beitragen, die Effektivität und Effizienz von Untersuchungen zu verbessern, insbesondere in Fällen, in denen Vor-Ort-Triage stattfindet oder in denen eine SHPO angeordnet wurde. Drei Punkte wichtig: ■ Wenn der Verdacht besteht, dass PB stattgefunden hat, hilft es zu wissen, wie erfolgreich die PB-Funktion eines bestimmten Browsers ist, um unnötige Datenverarbeitung (und Zeitverschwendung) zu vermeiden, wenn tatsächlich keine Browserdaten auf einem Gerät vorhanden sind. ■ Die Kenntnis darüber, wo PB möglicherweise Informationen zu Browsing-Sitzungen preisgibt, verbessert die Effizienz von Untersuchungen und verhindert, dass wichtige Inhalte übersehen werden. Dies ist besonders wichtig bei Vor-Ort-Triage, wie sie in einigen Fällen mit einer SHPO angeordnet wird.

Ziel der Arbeit: ===== - Welche Browsing Artefakte werden beim private Browsing auf einem Rechner hinterlassen, welche zeigen, dass eine Browsing Aktion vom Browser durchgeführt wurde? - Das heißt: o Es wird nach Browsing Artefakten gesucht, welche die Zuordnung „Durchgeführte Browsing Aktion“ <-> Browser ermöglichen o Vor, während und nach private Browsing Session nach Browsing Artefakten suchen, welche dem Browser zugeordnet werden können - Negativbeispiel: Suche in Hexdump nach im Browser gesuchtem String nicht als Beweis ausreichend, dass private Browsing Artefakte gefunden wurde. - Kategorisierung nach [18]: > Browsing History > Usernames/Email Accounts > Images

=> Thematisiert in [18]: o It appeared that the overall best way to recover residual data was to obtain the evidence from RAM or working memory, o Kritik: Oft nur String Match in RAM-Hex als Nachweis für PB genannt -> ausreichend? (Evtl. Gegenexperiment mit Editor)

Warum muss String-Artefakt Browser zugeordnet werden können? [12] ■ Die Artefakte, die von den Browsing-Aktivitäten eines Kriminellen zurückgelassen wurden, können mit forensischen Tools extrahiert werden, um die Untersuchung des Ermittlers zu unterstützen. ■ Die erlangten Beweise müssen vor Gericht zugelassen werden, insbesondere digitale Beweise, da sie ohne ordnungsgemäße Verfahren leicht manipuliert werden können. ■ Es gibt bestimmte Merkmale von digitalen Beweisen, die Gerichte

nach folgenden Kriterien akzeptieren: 1. Durchsuchungsbefehle - Beweise, die ohne Genehmigung erlangt wurden, können vor Gericht nicht anerkannt werden. 2. Berichte - Alle Prozesse, Werkzeuge, Methoden, Techniken, spezifischen Zeit- und Datumsangaben sowie die Beweiskette müssen formell dokumentiert werden, um die Authentizität der digitalen Beweise vor Gericht zu demonstrieren und zu unterstützen. 3. Beweisauthentifizierung - Der ursprünglich erhaltene Beweis sollte durch Vergleich der Hash-Werte mit dem Kopiebeweis übereinstimmen. Der erworbene Beweis muss unverändert bleiben, um die Gerichte mit genauen Informationen zu überzeugen. Gerichte akzeptieren Kopien von Beweisen, wenn der ursprüngliche Beweis verloren gegangen oder zerstört wurde, die Kopie jedoch noch intakt ist.

Ziele anderer Arbeiten: ===== > [12] - Die Art der extrahierbaren Daten zu untersuchen - den Unterschied zwischen privatem und normalem Surfen zu vergleichen - zu analysieren, welcher Browser die vollständigeren residualen Daten liefert. > [15] ■ ob bestimmte Arten von Browser-Daten gefunden werden konnten (Webseiten, Verlauf, Download-Verlauf, besuchte URLs und Suchbegriffe) > [21] ■ Das Ziel dieser Studie ist es, eine Rahmenbedingung für die Analysephasen des Webbrowsers im privaten Modus und Anti-Forensik vorzuschlagen, um eine effektive und effiziente forensische Untersuchung zu ermöglichen. ■ Die Studie nutzt Live-Forensik, um detailliertere Informationen über den Computer zu erhalten, während er noch in Betrieb ist, und eignet sich daher besser für die schnelle Datenerfassung in Echtzeit. > [24] ■ umfassende Analyse der privaten Browsing-Funktion in den vier beliebtesten Webbrowsers (IE, Firefox, Chrome und Safari) vorgestellt. > [12] - digitalen Forensikern helfen, Artefakte von Geräten zu verfolgen, die Live-Memory-Erfassung verwenden > [16] - Methodik entwerfen, um folgende Fragen zu beantworten: 1. Kann Tor den Benutzer schützen, indem es Beweise für dessen Nutzung aus dem RAM löscht, wenn die Browsing-Sitzung geschlossen wird? 2. Kann die Tor-Nutzung zu vier Schlüsselmomenten erkannt werden: während das Browser-Fenster geöffnet ist, nach Schließen des Browser-Fensters, nach dem Löschen des Installationsverzeichnis/zugehöriger Dateien und nach dem Ausloggen des Benutzers? 3. Können Dateien aus dem Browsing-Protokoll in der Live-Forensik mit Tor 7.5.2 wiederhergestellt werden, der zum Zeitpunkt der Schreibens aktuellsten Version? - Die Experimente wurden im mobilen Modus mit Tor wiederholt, d.h. von einem USB-Stick ausgeführt. (!!!) zu bestätigen, dass die Existenz und Nutzung des Tor-Browsers in Windows 10 nachweisbar ist. (!!!) nachweisen, dass Artefakte des Tor-Browsing-Protokolls auf dem Zielcomputer wiederhergestellt werden können. > [12] > In dieser Studie werden die residualen Daten zwischen Google Chrome und Mozilla Firefox Webbrowsers im normalen und privaten Browsermodus mithilfe eines forensischen Tools analysiert und verglichen. > [15] ■ In dem Projekt wurden die Effektivität der "privaten"Modus von vier weit verbreiteten Webbrowsers analysiert. > [24] ■ Ziel: Bewertung der Sicherheit des privaten Surfens in den Browsern Chrome, Safari, Firefox und IE ■ Die Autoren haben eine umfassende forensische Analyse durchgeführt, die sowohl Live-Memory-Analyse als auch Post-Mortem-Analyse umfasste. > [15] ■ Vier getestet: Firefox, IE, Safari und Chrome

Keine Ziele der Arbeit: ===== - zeitl. Kontext nicht wichtig - Private Browsing Indicators": Entering/Leaving Private Browsing [18] - Zeigen, dass ein Browser gestartet/geschlossen wurde - Zeigen, dass ein Browser im privaten Modus gestartet wurde - Zeigen, wann ein Browser gestartet/geschlossen wurde - Browser-Erweiterungen: [24] > Browser-Erweiterungen und ihre Auswirkungen auf das private Surfen wurden in einer Studie von Aggarwal et al. Im Jahr 2010 untersucht. -> Siehe Punkt „Add-Ons als Leck“ > Die Chrome-Erweiterung „Incognito Inspector“ kann im privaten Modus genutzt werden, um detaillierte Informationen über die Nutzeraktivitäten zu sammeln und in Echtzeit an einen Remote-Server zu senden. > Firefox-Erweiterungen sind standardmäßig im privaten Modus aktiviert und können genutzt werden, um Nutzeraktivitäten aufzuzeichnen. > Internet

Explorer-Erweiterungen sind in der Regel deaktiviert und erfordern die manuelle Aktivierung im privaten Modus. Die von den Autoren entwickelte Erweiterung funktionierte jedoch nicht, da sie aufgrund eingeschränkter Privilegien nicht auf die BHO-Klasse zugreifen konnte - [1] > Unterschiedliche Handhabung durch Browser: Gefährliche Leckage für private Browsing Artefakte > Entwickler von Add-Ons haben möglicherweise den privaten Browsing-Modus bei der Entwicklung ihrer Software nicht berücksichtigt, und ihr Quellcode wird nicht derselben rigorosen Überprüfung unterzogen wie die Browser selbst. > Gegenmaßnahme: [1] ■ Es wurde eine Firefox-Erweiterung namens ExtensionBlocker entwickelt, um unsichere Erweiterungen im privaten Modus zu blockieren

4 Methodik

> Validation Stage (= Kapitel „Vergleich der Browser“)

Warum Methodik? > [1] Aufgrund der Komplexität moderner Browser ist eine systematische Methode erforderlich, um zu testen, ob der private Browsing-Modus ausreichend gegen die Bedrohungsmodelle aus Abschnitt 2 verteidigt. > [12] ■ Die Verfahren für die digitale Forensik für Browser-Forensik müssen angemessen befolgt werden, um dem Ermittler bei der Durchführung der Untersuchung zu helfen. Die Verfahren unterscheiden sich je nachdem, wie die Untersuchung durchgeführt werden soll. > [11] ■ Das Fehlen von Klarheit hat einen signifikanten Einfluss auf forensische Untersuchungen von Strafverfolgungsbehörden und deren Ansätze ■ Eine Kette von Beweisführung muss dokumentiert werden, um die Integrität und Zuverlässigkeit der Daten sicherzustellen. ■ Ein formaler forensischer Bericht wird dann vor Gericht präsentiert.

Bekanntes Computer Forensik Vorgehensmodell: [25]: Generic Model Computer Forensics Investigations (GCFIM) -> Daran orientieren sich alle in der Literatur

BSI: abschnittsbasierter Verlauf einer forensischen Untersuchung Vorbereitung; - Datensammlung; - Untersuchung: Extraktion von Bilddateien aus dem Image - Datenanalyse: Verbindungen zwischen mehreren Daten herstellen - Dokumentation

Phasen nach [15] ■ Die forensische Analyse erfolgt in zwei Phasen. 1. Zunächst wird die Analyse an sowohl "üblichen als auch ungewöhnlichen Speicherorten auf der Festplatte durchgeführt. 2. In der zweiten Phase wird der physische Arbeitsspeicher (RAM) untersucht.

Phasen nach [12]: ■ Es gibt verschiedene Modelle für digitale Forensik, die sich in ihren Phasen unterscheiden können. ■ Fünf Phasen sind besonders wichtig: Identifikation und Sammlung, Bewahrung, Erwerb, Analyse und Prüfung sowie Dokumentation. ■ In der Identifikations- und Sammelphase werden alle potenziellen Beweismittel identifiziert, gekennzeichnet und gesammelt, um sie in der nächsten Phase zu verwenden. ■ Beweismittel können z.B. Log-Dateien, temporäre Dateien, Netzwerkverbindungen, Browserverlauf und Cache sein. > Phasen: Preparation Phase o Versuchsplanung + Konfiguration der HW/SW + Durchführen des Experiments Acquisition Stage o Abbildung von der Festplatte (Static Forensics) und des RAMs (Live Forensics) Analysephase o Bilder der Speicherabbilder mit einem forensischen Tool untersuchen Validierungsphase o gefundenen Artefakte verglichen und dokumentiert

4.1 Vorbereitung

- [12] Versuchsplanung + Konfiguration der HW/SW + Durchführen des Experiments Acquisition Stage - BSI: alle Maßnahmen, die nach vermuteten Eintreten eines Vorfalls aber vor der eigentlichen Datensammlung erfolgen. -> z.B. Identifikation und Enumeration potentieller Datenquellen

4.1.1 Browserauswahl

- > Browserstudie [12] - Die Herstellerangaben unterschiedlicher Browser bzgl. Privatheit untersucht - Firefox 58.02: No Browsing History stored, No Cookies stored, No login Info stored, Tracking Protection Enabled: Disconnect, Download Files not Hidden - Chrome 63.0.3239: No Browsing History stored, No Cookies stored, No login Info stored, Tracking Protection Enabled: No, Download Files not Hidden
- > design aim of Tor: [16] - preventing from writing to disk (Perry et al., 2018) - enabling secure deletion of the browser (Sandvik, 2013) (hier nicht relevant)

4.1.2 Browsing Szenario

- Wichtig für White-Box-Ansatz: Browsing Szenario ist bekannt - URL X ... (TODO!) *** TODO: Automatisch geöffnete Firefox Seiten erwähnen *** *** TODO: Hier auch Browser Artefakte (Strings) auflisten ***

VM Konfiguration

Virtualisierung: > Oracle VirtualBox VM, Version 7.0.8 r156879 (Qt5.15.2) TODO: Warum Virtualisierung? > Pro Browser eine VM

VM Betriebssystem: In Literatur (wie viele?) ausschließlich Windows untersucht -> Da Ziel dieser Arbeit: keine neuen wissenschaftlichen Ergebnisse => auch Windows 10 Dazu: Windows 10 Pro, Build: 19045.2006 (nicht aktiviert) > Auf einer VM installiert, danach OVA erstellt. Von dieser ausgehend, für jeden Browser VM erstellt

VM Massenspeicher: 30 GB (VDI-Format), kein SSD-Laufwerk

VM RAM: Kaum Angaben in der Literatur: > [21]: 2 GB > [18]: 4 GB - Hier: 6 GB (maximal möglich, mit verfügbaren Speicherressourcen, um später RAM-Dumps zu speichern) -> Ausblick: Kritik an Literatur, dass RAM-Größe kaum thematisiert wird, obwohl sie Auswirkungen auf Ergebnisse hat -> Siehe Pagefile-Thematik in Analysis Stage

VM Netzwerkeinstellungen: An Literatur orientiert (Quelle?) - Netzwerkadapter mit Netzwerkbrücke = direkt mit dem physischen Netzwerk deines Hostsystems verbinden -> virtuelle Maschine eine eigene IP-Adresse im physischen Netzwerk - Netzwerkadapter erst nach Browserinstallation aktiviert

Programme auf VM: *** TODO: Shared Folder für Installation *** - Untersuchter Browser. Installationsverzeichnisse: > Firefox: > Tor: > Chrome: > Brave:

- Process Monitor "Process Monitor ist ein Dienstprogramm für das Windows-Betriebssystem, das von Microsoft entwickelt wurde. Es ermöglicht die Überwachung und Aufzeichnung aller Aktivitäten und Ereignisse, die auf einem Windows-System im Zusammenhang mit Prozessen, Dateisystemen, Registrierungseinträgen, Netzwerkverbindungen und vielem mehr stattfinden. Mit Process Monitor kannst du detaillierte Informationen über den Betrieb deines Systems erhalten, um Probleme zu diagnostizieren, Softwarekonflikte zu lösen oder verdächtige Aktivitäten zu untersuchen. Das Tool bietet eine umfassende Echtzeit-Überwachungsfunktion, die es dir ermöglicht, alle Ereignisse und Aktionen zu erfassen, die während der Ausführung von Prozessen auf deinem System auftreten."Version:

v3.93 Hier: Verwendet, um Aktivitäten des Browsers aufzuzeichnen (vgl. Wireshark Capture) -> Siehe Sammlungsphase

- Process Explorer "Process Explorer erweitert die Funktionen des Windows Task Managers und ermöglicht es Benutzern, einen umfassenden Überblick über alle aktiven Prozesse und deren Eigenschaften zu erhalten." Hier wichtig: Ermöglicht zudem alle ausgeführten Services einer PID zuzuordnen Version: v17.04

4.1.3 Analysewerkzeuge

Volatility

"Das Volatility Memory Forensics Framework ist ein Open-Source-Tool, das für die forensische Analyse von Arbeitsspeicherabbildern verwendet wird. Es ist speziell darauf ausgerichtet, Informationen und Artefakte aus dem physischen oder virtuellen Arbeitsspeicher eines Computers zu extrahieren.

Frei auf GitHub verfügbar: Version: 2.4.1 (aktuellster Release) Basierend auf python

Die Volatility Foundation wurde ins Leben gerufen, um die Interessen der Volatility-Community zu vertreten und sicherzustellen, dass das Framework frei verfügbar bleibt und kontinuierlich weiterentwickelt wird. Das Projekt wird von einer Gemeinschaft von Freiwilligen vorangetrieben, die zur Code-Entwicklung, Dokumentation und Fehlerbehebung beitragen.

Das Tool wurde entwickelt, um Forensikern und Sicherheitsexperten bei der Untersuchung von Sicherheitsvorfällen, Malware-Angriffen und anderen verdächtigen Aktivitäten zu unterstützen. Durch die Analyse des Arbeitsspeichers können wichtige Informationen wie Prozesse, Netzwerkverbindungen, Dateisystemstrukturen, Registrierungseinträge und sogar verschlüsselte Daten wiederhergestellt werden."

Volatility3: "Volatility 3 tatsächlich eine vollständige Neuschreibung des Volatility Memory Forensics Frameworks zu sein, die im Jahr 2020 veröffentlicht wurde. Diese Neuschreibung sollte viele technische und Leistungsprobleme der ursprünglichen Version beheben, die seit der ersten Veröffentlichung im Jahr 2007 aufgetreten sind.

Volatility 3 wurde mit dem Ziel entwickelt, eine robustere, flexiblere und skalierbare Plattform für die Analyse von Arbeitsspeicherabbildern bereitzustellen. Es wurden Verbesserungen hinsichtlich der Architektur, der Leistung und der Plugin-Entwicklung vorgenommen. Die Neuschreibung ermöglichte auch eine Anpassung der Lizenzierung durch die Einführung der Volatility Software License (VSL)."

Großer Vorteil von Volatility3: kein "Profile" mehr notwendig. = Konfigurationseinstellung, die angibt, um welches Betriebssystem und welche Version es sich im Arbeitsspeicherabbild handelt. Ein Profil definiert die Speicherstruktur und die Verhaltensweisen des Betriebssystems, die Volatility bei der Analyse des Arbeitsspeichers berücksichtigen muss. => Grund: kann neue Symboltabellen für die meisten Windows-Speicherabbilder generieren, basierend auf dem Speicherabbild selbst

Plugins-Liste: (Ähnlich zu [Hariharan] und [4]) > pslist > yarascan > memmap > filescan > svcscan

Genaue Beschreibung der Plugins und deren Zusammenhang: Siehe Analysephase Zusammenhang der Plugins: Siehe Analysephase

Autopsy

Autopsy ist ein Open-Source-Digital-Forensik-Tool, das auf der Sleuthkit-Bibliothek basiert. Es wurde entwickelt, um forensische Untersuchungen von digitalen Beweismitteln auf Computern und mobilen Geräten durchzuführen.

Sleuthkit ist eine Sammlung von Open-Source-Tools für die forensische Analyse von Dateisystemen. Es bietet Funktionen zum Lesen, Analysieren und Wiederherstellen von Daten aus verschiedenen Dateisystemen, darunter NTFS, FAT, exFAT, HFS+, Ext2/3/4 und mehr. Sleuthkit ermöglicht die Extraktion von Metadaten, das Durchsuchen von Dateien und die Analyse von Dateisystemstrukturen, um wichtige Informationen für forensische Untersuchungen zu gewinnen.

Autopsy baut auf der Funktionalität von Sleuthkit auf und bietet eine benutzerfreundliche grafische Benutzeroberfläche für die forensische Analyse. Es erweitert die Funktionalität von Sleuthkit, indem es zusätzliche Tools, Plug-Ins und Automatisierungsfunktionen bereitstellt, um den forensischen Untersuchungsprozess zu unterstützen.

Mit Autopsy können Forensiker die Daten auf einem Datenträger untersuchen, Dateien und Metadaten analysieren, gelöschte Dateien wiederherstellen, Keyword-Suchen durchführen, Berichte generieren und vieles mehr."

SSowohl Autopsy als auch Sleuthkit sind weit verbreitete und respektierte Tools in der digitalen Forensik-Community. Sie werden von Forensikern, Strafverfolgungsbehörden, Sicherheitsexperten und anderen Fachleuten eingesetzt, um digitale Beweise zu sammeln, zu analysieren und Berichte zu erstellen, die in rechtlichen Verfahren verwendet werden können."

Hier: Verwendet um Festplattenimages der VMs einzulesen + Interaktion mit Dateien im Image.

Sonstige Tools

Darstellung von Daten: - HxD - Notepad++ (inkl. PlugIns, bsow. JSON)

Untersuchung der Registry: - Registry Explorer: Registry Explorer ist ein leistungsstarkes forensisches Tool zur Untersuchung und Analyse der Windows-Registry. Es ermöglicht Benutzern, auf die Inhalte der Registry zuzugreifen, diese zu durchsuchen und zu bearbeiten. Das Tool wurde von Eric Zimmerman entwickelt, einem erfahrenen Forensiker und Entwickler von forensischen Tools. Mit "Registry Explorer" können Benutzer die Registry-Schlüssel und Werte anzeigen, bearbeiten und löschen, nach bestimmten Einträgen suchen und Sicherungskopien erstellen.» Wichtig: Ermöglicht effiziente Stringsuchen

SQLite: - sqldiff ***TODO***

Tools zur Dateiverwaltung: - dejsonlz4: = Hilfsprogramm des GitHub Nutzers ävih", das verwendet wird, um JSON-Dateien zu dekomprimieren, die im LZ4-komprimierten Format vorliegen. > jsonlz4 = proprietäres Firefox-Format Benutzerdaten zu speichern. Das Format basiert auf JSON (JavaScript Object Notation), einem verbreiteten Datenformat zur Repräsentation strukturierter Daten. Die Daten werden jedoch zusätzlich mit dem LZ4-Komprimierungsalgorithmus komprimiert, um Speicherplatz zu sparen und den Zugriff zu beschleunigen. > Um auf die Informationen in einer "jsonlz4Datei zuzugreifen, muss sie zuerst dekomprimiert werden. > "dejsonlz4": Dekomprimierung von lz4. Danach:

normale JSON-Datei, die gelesen und verarbeitet werden kann. - MZCacheView: = Tool zum Parsen von Firefox Cache-Dateien. Kann auf diese Dateien zugreifen und verschiedene Informationen anzeigen, wie z.B. URL, Dateigröße, MIME-Typ und Datum des Downloads. > Erwartet Firefox Cache-Dateien: > "Das Cache2-Verzeichnis ist der Speicherort, an dem Firefox zwischengespeicherte Dateien speichert FirefoxCache2: > Skript "firefox-cache2-index-parser.py» Erweitert MZCacheView, indem die sogenannte indexDatei im Cache2 Ordner analysiert wird. "Die Indexdatei fungiert als Referenz für den Cache und ermöglicht es Firefox, schnell auf die zwischengespeicherten Dateien zuzugreifen"

4.2 Datensammlung

[12] In der Identifikations- und Sammelphase werden alle potenziellen Beweismittel identifiziert, gekennzeichnet und gesammelt, um sie in der nächsten Phase zu verwenden. > Browsing Szenario durchführen

Gesammelte Daten: > Process Monitor Logfiles: - siehe Ziel der Arbeit: "White-BoxAnsatz: Versuch wird nicht aus Sicht von Forensiker durchgeführt, der nur begrenzten Zugriff auf Beweismittel hat. - Sondern: Ziel ist das Verhalten von privaten Browsing möglichst vollständig zu untersuchen, um so viele Artefakte wie möglich zu identifizieren. - Deshalb: von Fayyad-Kazan et al. vorgeschlagen [5] Aktivitäten während Browsing-Szenario aufzeichnen mit "Process Monitor". > Während Browsing Szenario Filechanges untersuchen: DaemonFS set to monitor all activity within local hard drive[18] - Aufgezeichnet werden Aktivitäten in beispielsweise der Registry, im Dateisystem und Netzwerkaktivitäten. - Mit grafischer Oberfläche: Start, Stopp und Sicherung der Aufzeichnung als PML Datei (Process Monitor Logfile) -> Kann in CSV Umgewandelt werden - Danach: Präzise Filterung der Ergebnisse. Fokus hier nach Empfehlung von Autoren in [5]: Schreibaktivitäten auf Dateien und in Registry. - Registry: [21] ■ Das Windows-Registrierungsverzeichnis enthält viele Informationen zur Nutzung des Computers, Benutzerkonfigurationen, Anwendungen und Hardwaregeräte ■ Informationen im Registrierungsverzeichnis werden nach Ausführungsreihenfolge, Suchschlüsselwörtern, zuletzt aufgerufenen Ordnern, Anwendungsprotokollen und anderen Kategorien sortiert. *** TODO: Shared Folder ***

> Speicherabbilder: > Hauptaufgabe eines Computer-Forensik-Ermittlers: Erwerb und Analyse von Speicherabbildern von Computervorrichtungen. [9] > Speicherabbild = statischer Schnappschuss der Daten auf sekundärem Speicher (HDD, SSD), angeschlossenen Speichergeräten (USB-Stick, externe Festplatte, Magnetband) oder RAM-Speicher (bei Live-Erfassung auf laufenden Systemen). [9] > Speicherabbild ist Container für Daten, der individuelle Dateien oder das gesamte Laufwerk/Live-Speicherdateien in einer Image-Datei speichern kann. [9] > Forensische Abbilder enthalten digitale Beweise, die zur Identifizierung von Sicherheitsvorfällen, Betrug und anderen illegalen Praktiken, die auf Informationssysteme abzielen, wiederhergestellt und analysiert werden müssen. [9] > Forensische Abbilder können vor Gericht verwendet werden, daher müssen die verwendeten Tools und Techniken zur Erwerbung und Analyse rechtlich akzeptiert sein. [9] > Für diesen Versuch relevant und in Literatur viel verwendet: Festplatten-Abbilder (TODO: Alle Quellen) und RAM-Abbilder (TODO: Alle Quellen) > Um Festplatten-Image zu sichern gibt es diverse Software- und Hardware Lösungen. > Da hier Virtualisierung verwendet: Festplatten Abbild entspricht einem "VM-SnapshotEin VM-Snapshot, auch virtueller Maschinensnapshot genannt, ist ein Zustandssnapshot einer virtuellen Maschine (VM). Ein Snapshot erfasst den genauen Zustand der VM zu einem bestimmten Zeitpunkt, einschließlich des Arbeitsspeichers, des Prozessorzustands und des Festplatteninhalts. Im Wesentlichen ist es eine

Momentaufnahme des virtuellen Systems. "VM Snapshot kann über grafische Oberfläche von VirtualBox durchgeführt werden: > Ausgehend vom letzten Zustand der VM > Menü "Sicherheit" > Option "Erzeugen" Kann sowohl in an- und ausgeschaltetem Zustand durchgeführt werden. *** ==> Das heißt, ein VM-Snapshot kann sowohl in Live- als auch Post-Mortem-Forensik verwendet werden <==> *** > Wichtig: es wird zwar .vdi Datei in Snapshot-Ordner erzeugt, Datei enthält aber nur differentielle Daten zum aktuellen Zustand bzw. zum vorherigen Snapshot, wenige KB groß. > Um vollständige Festplatte zum gesicherten Zeitpunkt zu erzeugen: Snapshot über Option "Klonen als eigenständige VM erzeugen. Wichtig: "vollständiger Klon auswählen > VDI Festplatte des vollständig geklonten Snapshots: enthält alle Festplatten-Daten > Problem: - Um Festplatten-Abbild später in Datenanalyse zu untersuchen: muss in Autopsy eingelesen werden. - Autopsy unterstützt jedoch nicht VDI Format. > Deshalb: Umwandlung in .img Format "Das Image-Format (.img) ist ein generisches Dateiformat, das verwendet wird, um ein binäres Abbild eines gesamten Datenträgers zu speichern. Es erfasst den Inhalt des Datenträgers, einschließlich des MBR (Master Boot Record), der Partitionstabellen, der Dateisysteme und aller Dateien und Ordner." Dazu "vboxmanage" verwendet = Kommandozeilen-Tool, das bei Installation von VirtualBox enthalten ist. Wobei: .vdi Datei das zu konvertierende VirtualBox Disk Image und .img die neue Image-Datei > *** TODO: Einlesen in Autopsy -> Zeitaufwändig! ***

> Im großen Unterschied zu VM Snapshots: Abbild des Arbeitsspeichers nur in angeschaltetem Zustand möglich (Zumindest bei virtueller Maschine, ohne spezielle HW-Werkzeuge) Auch als Ein RAM-Dump bezeichnet, ist eine Momentaufnahme des aktuellen Inhalts des Arbeitsspeichers (RAM) eines Computers zu einem bestimmten Zeitpunkt. Ein RAM-Dump erfasst den genauen Zustand des RAMs, einschließlich der im Speicher befindlichen Daten, Programme und Prozesse.» Um RAM-Dump einer laufenden virtuellen Maschine durchzuführen: > Keine weitere Verarbeitung des RAM-Abbilds notwendig, kann direkt von Analysetool "Volatility" eingelesen werden.

> Kritisch für Versuch ist die Definition der Zeitpunkte, zum Sammeln der Daten: - In Literatur kaum thematisiert: - Häufigster Fall: RAM-Dump nach Durchführung von Browsing Szenario für Live-Forensik [8, 12, 14, 18] und Festplattenabbild gesichert nach herunterfahren der VM für Post-Mortem-Forensik [3, 5, 6, 11, 13–15, 18] - Sonderfälle: > RAM-Dump nur wenn Browser noch offen ist, nach Szenario, aber kein Dump nach Schließen [13] > Mehrere RAM-Dumps zu unterschiedlichen Zeitpunkten: Vor Schließen von Browser, nach Schließen von Browser, nach Löschen der Registry [21] - Teilweise auch gar keine Zeitpunkte definiert [1, 15, 17, 22–24] - Problem haben Muir, Leimich und Buchanan erkannt und Zeitpunkte zur Datensammlung vorgeschlagen, um gesamtes Browserverhalten während des Szenarios zu überwachen und zu analysieren. - An diesen Zeitpunkten für diesen Versuch orientiert: Wichtigste

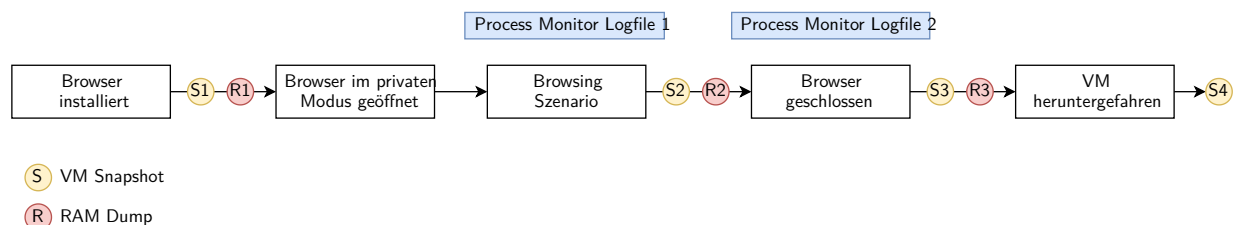


Abbildung 4.1: Datensammlung Zeitpunkte

Zeitpunkte: - Nach Browser-Installation, vor Browsing-Szenario: RAM-Dump 1, VM-Snapshot 1 => Dient als Benchmark für die Analyse: In diesen Speicherabbildern darf kein PB Artefakt gefunden

werden - Öffnen des Privaten Modus, vor Browsing Szenario: Start Logfile 1 => Bekannt: Während erstmaligem Öffnen von Browser viele Dateien initial angelegt. Um ausschließlich Schreiboperationen aufzuzeichnen, die auf private Browsing zurückzuführen sind: Start erst, nachdem Browser erstmalig im privaten Modus geöffnet wurde - Nach Durchführung des Browsing-Szenarios, Browser geöffnet: Ende Logfile 1, RAM-Dump 2, VM-Snapshot 2, Start Logfile 2 => Logfile 1 enthält nun Schreiboperationen während Browsing Szenario => Wichtig: Zweites Logfile nach sichern der Speicherabbildern gestartet, um Dateiänderungen beim Schließen der Browser aufzuzeichnen - Nach Schließen des Browsers: Ende Logfile 2, RAM-Dump 3, VM-Snapshot 3 - Nach Herunterfahren der VM: VM Snapshot 4 => Relevant für Post-Mortem Analyse

Sonderfall Tor: - Tor hat die Funktion der "Neuen Identität" im Tor-Browser ermöglicht die "Neue

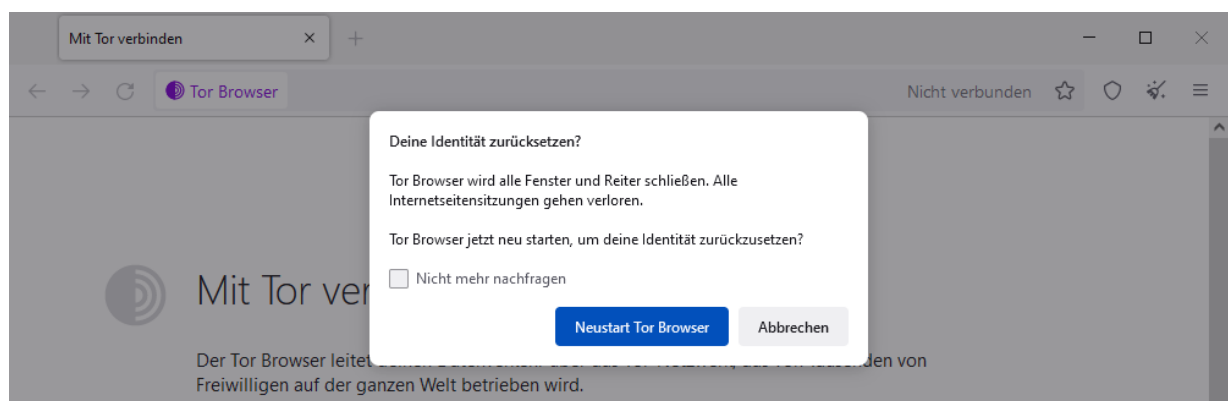


Abbildung 4.2: Tabelle mit wiederherstellbaren Dateien: Logfile 1 vs. Logfile 2

Identität eine schnelle und einfache Möglichkeit, Ihre Online-Identität zu ändern und Ihre Privatsphäre zu wahren. Wenn Sie den Identitäts-Reset im Tor-Browser durchführen, werden bestimmte Informationen und Daten gelöscht oder zurückgesetzt, um potenziell die Rückverfolgbarkeit Ihrer Online-Aktivitäten zu verringern.» Von Tor angegebene Auswirkungen: alle deine geöffneten Tabs und Fenster geschlossen, alle privaten Informationen wie Cookies und Browserverlauf. - Deshalb: Bei Tor Daten zusätzlich vor und nach Neuer Identität sammeln.

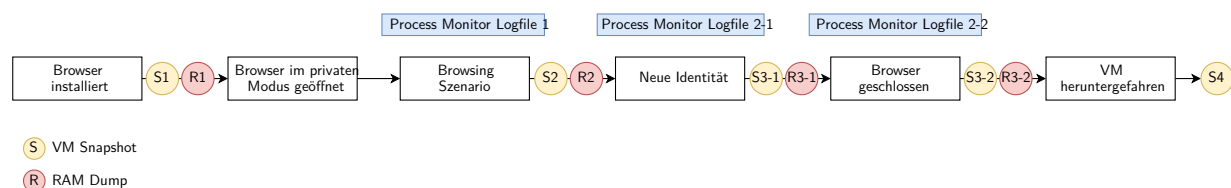


Abbildung 4.3: Datensammlung Zeitpunkte Tor

4.3 Datenanalyse

> Analysis Stage (= Kapitel „Results“) - Analyse der gesammelten Daten der vorherigen Phase: > Process Monitor Logfiles > Speicherabbilder: VM-Snapshots, RAM-Dumps - Analyse heißt hier: Suchen

nach PB Artefakten in gesammelten Daten = Suchen Nach Strings, die einem Schritt im Browsing Szenario zugeordnet werden können *** TODO: Liste von Strings ***

Wichtig dabei: (siehe Ziel der Arbeit): gefundenes Artefakt muss eindeutig Browser zugeordnet werden können. Oft kritisiert: Viele Autoren verlassen sich beispielsweise bei der Analyse der RAM-Dumps auf eine einfache Stringsuche in einem Hexadezimal-Editor. Sagt jedoch nichts darüber aus, ob gefundener String tatsächlich auf privates Browsen zurückzuführen ist. *** TODO: String in Editor Beispiel ***

Die gesammelten Daten dieses Versuchs lassen sich in drei Kategorien aufteilen: > Common Locations > Uncommon Locations > Registry

4.3.1 Common Locations

"Gängige Speicherorte beziehen sich auf die standardmäßigen Verzeichnisse oder Pfade, in denen bestimmte Dateien oder Informationen typischerweise in einem Betriebssystem oder einer Anwendung gespeichert werden. auch "Vor-Ort-Triage"[11] genannt, um bekannte Speicherorte nach PB Artefakte zu überprüfen - z.B. Profile-Ordner von Browsern, wo Nutzerdaten gespeichert werden - Befasst sich ausschließlich mit Dateien, die auf Festplatte geschrieben werden - SSuchrichtung": Bekannte Browser-Speicherorte und Dateien -> Prüfen ob Dateien PB Artefakte enthalten - Hier: Browser Speicherorte werden über die Schreiboperationen der Process Monitor Logfiles identifiziert

Eng verwandt mit den "Common Locations" ist die "Whitebox-Analyse": (gezieltes Suchen nach Dateien) [2] > Definition: "White-Box" Computer Forensik bezieht sich auf eine forensische Untersuchungsmethode, bei der der forensische Analyst über umfassende Kenntnisse und Zugriff auf das untersuchte System verfügt. Im Kontext der Computerforensik bezieht sich "White-Box" darauf, dass der Analyst über volle Transparenz und Zugriff auf alle Informationen, Ressourcen und Artefakte des Systems verfügt. > Die "White-Box" Forensik kann verschiedene Techniken und Tools umfassen (z.B. Process Monitor, Regshot, Registry Explorer, Dekomprimierungstools), um Daten wiederherzustellen, gelöschte Informationen wiederherzustellen, Metadaten zu analysieren, Netzwerkaktivitäten zu überwachen und weitere forensische Analysen durchzuführen. Der Fokus liegt darauf, das System vollständig zu verstehen und alle relevanten Beweise zu sammeln.

Process Monitor Logfiles

Grundlage für Common Locations = Process Monitor Logfiles

Datenaufbereitung für jede Logfile: - Für Common Locations Filter setzen: > Nur "File System Activity" > Process Name: contains "Browser-Prozess".exe - Firefox = firefox.exe - Tor = firefox.exe und tor.exe - Chrome = chrome.exe - Brave = brave.exe > Operation: contains "WriteFile" > Gefiltertes Logfile als CSV exportieren, dann in Excel öffnen - Irrelevante Spalten löschen: > Time of Day > Process Name (Da in Process Monitor bereits nach Namen gefiltert wurde -> Alle Prozesse haben gleichen Namen) > Operation (Da in Process Monitor bereits nach Operation gefiltert wurde -> Alle Prozesse haben gleiche Operation „WriteFile“) > Result > Detail - Gleiche Operationen (Duplikate) löschen - Gruppieren nach browserspezifischen Speicherorten

Datenanalyse für jede geschriebene Datei: 1) Dateiextraktion 1. In Autopsy prüfen, ob Datei in Image von entsprechendem Snapshot vorhanden. Siehe Abbildung X: > Logfile 1 -> Snapshot 2 > Logfile 2 ->

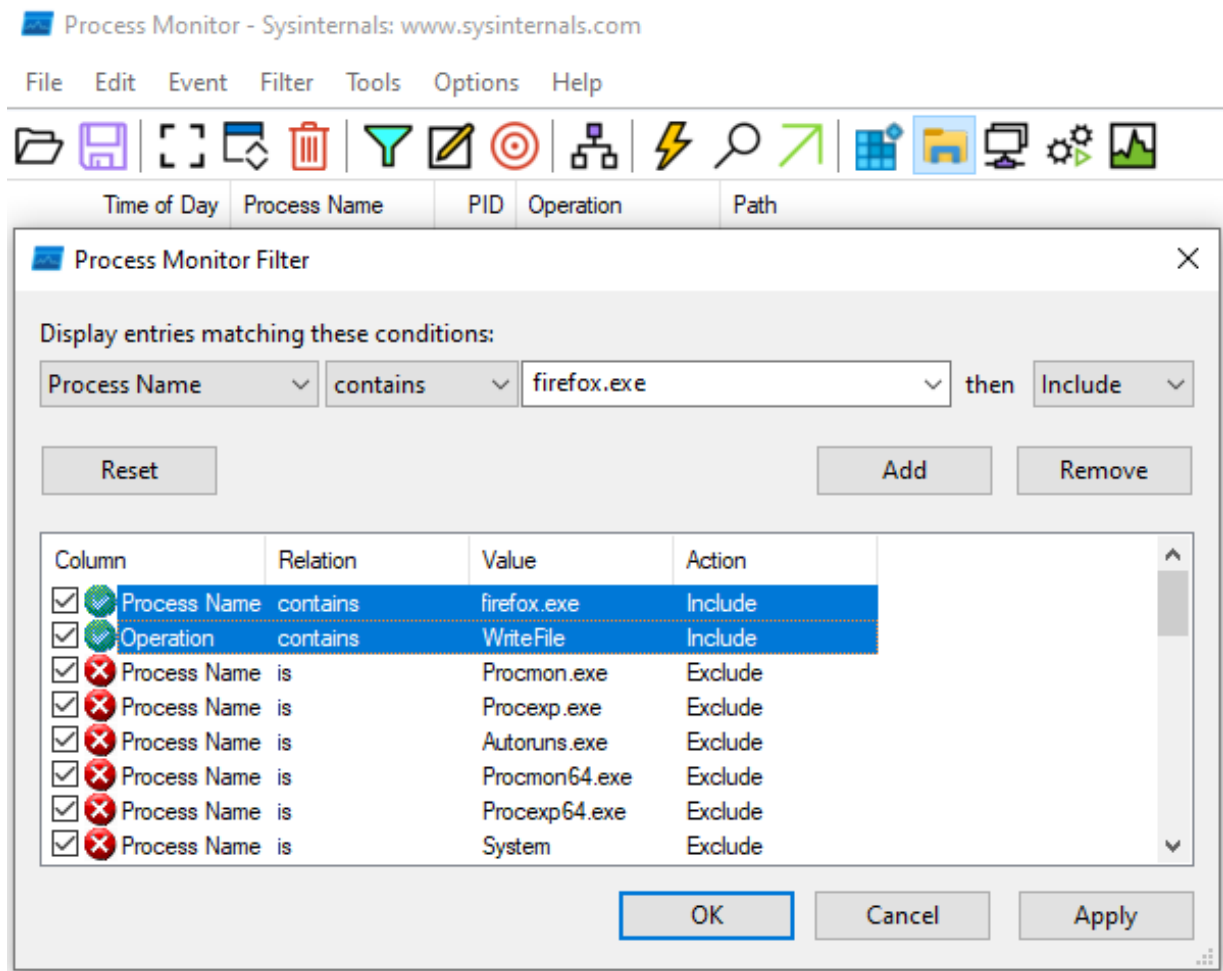


Abbildung 4.4: Tabelle mit wiederherstellbaren Dateien: Logfile 1 vs. Logfile 2

Snapshot 3 Tor: > Logfile 1 -> Snapshot 2 > Logfile 2-1 -> Snapshot 3-1 > Logfile 2-2 -> Snapshot 3-2

2. Wenn ja, Dateien mit Autopsy extrahieren

3. Wenn nein, prüfen, ob Datei in RAM gecacht: > Dazu Ausgabe von Volatility Plugin "filesScan" in entsprechendem RAM-Dump prüfen. > filesScan: Das filesScan-Plugin von Volatility durchsucht den Arbeitsspeicher nach Speicherbereichen, die Informationen über Dateien enthalten. Die Ausgabe des Plugins enthält eine Liste von Dateinamen, die im Speicher gefunden wurden. Können sein: Ausführbare Dateien, Bibliotheksdateien oder Konfigurationsdateien. Welchen Dump bei welcher Logfile? > Logfile 1 -> RAM-Dump 2 > Logfile 2 -> RAM-Dump 3

3 Tor: > Logfile 1 -> RAM-Dump 2 > Logfile 2-1 -> RAM-Dump 3-1 > Logfile 2-2 -> RAM-Dump 3-2

Wenn Datei gefunden wurde: Datei mit dumpfiles und entsprechender virtueller Dateispeicheradresse extrahieren.

4. Wenn Datei auch nicht im RAM gecacht ist: Prüfen, ob es sich bei Datei um Hilfsdatei handelt. Dabei handelt es sich meistens um temporäre Dateien mit zusätzlicher Dateiendung (meistens .tmp)

5. Wenn ja: Zur die Dateiendung der Hilfsdatei weglassen und bei Schritt 1 beginnen. z.B: Wenn Datei Bome-file.json.tmp nicht existiert, prüfen, ob Datei Bome-file.json existiert

6. Wenn nein: Datei als "nicht wiederherstellbar" markieren

II) Dateianalyse (nachdem Datei extrahiert wurde)

7. Datei mit entsprechendem Tool untersuchen

8. Wenn Datei nicht direkt untersuchbar ist: ggf. Mit

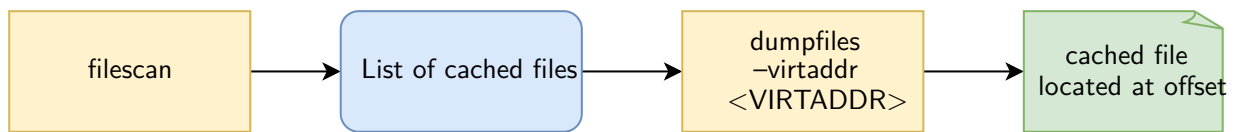


Abbildung 4.5: Datensammlung Zeitpunkte Tor

zusätzlichem Tool Datei vorbereiten. z.B. Wenn es sich um komprimierte Dateien handelt 9. In Excel-Tabelle markieren, ob Datei PB Artefakte enthält. Dabei gibt es drei Zustände: - leere Datei - neuer (nicht-leerer) Inhalt - gleichbleibender Inhalt

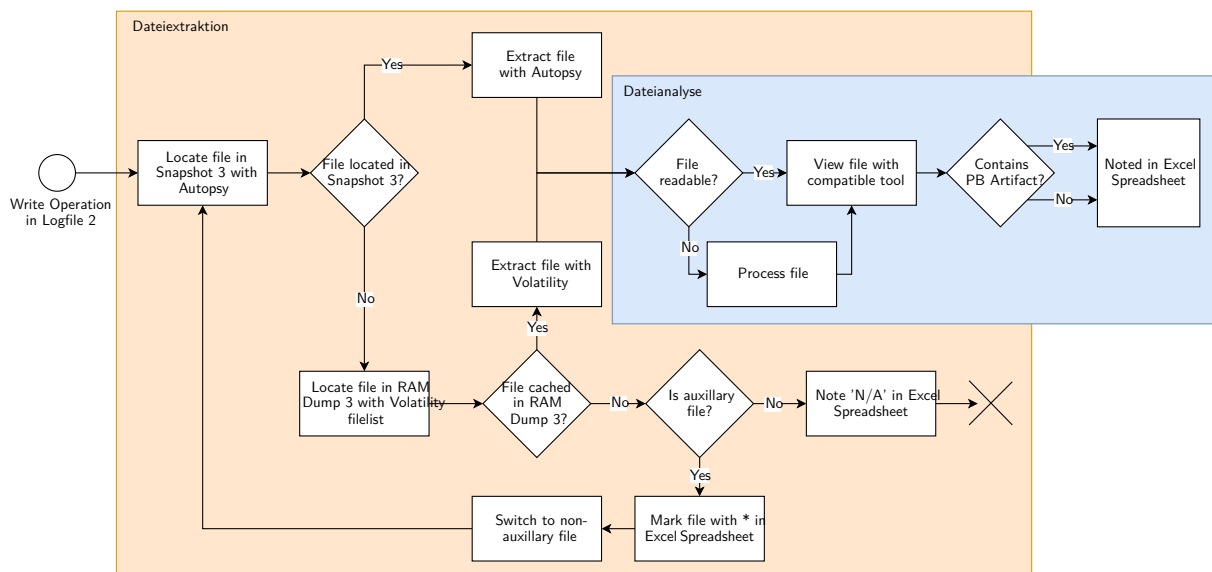


Abbildung 4.6: TODO: Process Monitor Write Operation to Excel Spreadsheet

SQLite-Datenbanken

- Besondere Rolle nehmen bei den ausgewählten Browsern SQLite Datenbanken ein.

SQLite-Datenbanken sind besonders wichtig bei Webbrowsern wie Firefox, Tor, Chrome und Brave, da sie Informationen wie Lesezeichen, Browserverlauf, Caches, Cookies und Erweiterungsdaten speichern. Diese Datenbanken bieten Benutzern eine personalisierte Browsererfahrung und sind für Forensiker und Sicherheitsexperten von Bedeutung, um relevante Informationen über die Online-Aktivitäten eines Benutzers zu analysieren."

"Die Verwendung von SQLite-Datenbanken in diesen Webbrowsern bietet Effizienz, Flexibilität und eine einfache Möglichkeit, Daten zu organisieren und abzurufen. Für Forensiker und Sicherheitsexperten sind diese Datenbanken von Bedeutung, da sie wertvolle Informationen über die Online-Aktivitäten eines Benutzers enthalten können, die bei Untersuchungen, Sicherheitsanalysen oder forensischen Untersuchungen relevant sein können."

> Deshalb: Für jeden Browser werden die SQLite Datenbanken in allen Snapshots miteinander verglichen.
 I) Dateiextraktion analog zu Dateien im Process Monitor II) Dateianalyse: - SQLite Datenbank wird

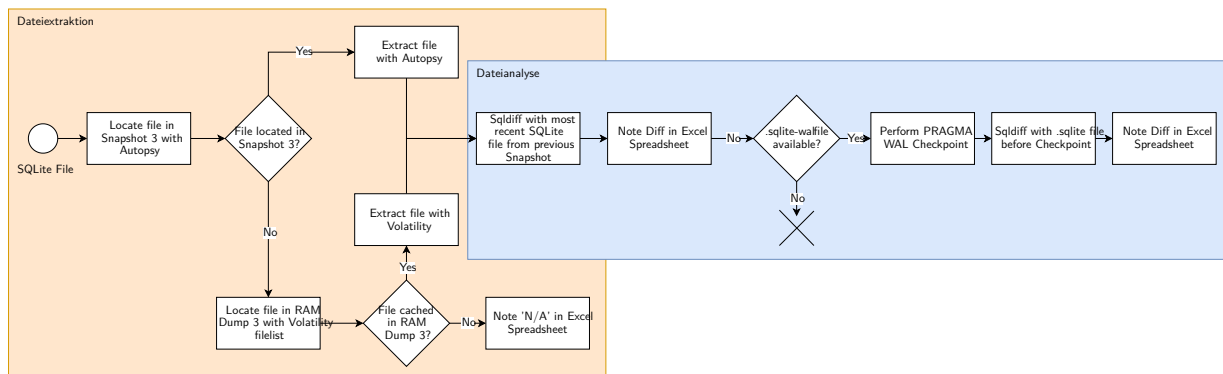


Abbildung 4.7: TODO: Process Monitor Write Operation to Excel Spreadsheet

mit gleicher SQLite Datenbank aus vorherigem Snapshot verglichen (wenn vorhanden) -> Unterschiede in Excel-Tabelle festgehalten - Wichtig: Zu jeder SQLite Datenbank gibt es normalerweise eine .sqlite-wal Datei. "Die Dateien mit der Erweiterung ".sqlite-wal bind WAL-Dateien (Write-Ahead Log) in Verbindung mit SQLite-Datenbanken. Sie werden von SQLite verwendet, um Änderungen an einer Datenbank vorübergehend zu protokollieren, bevor sie dauerhaft in die Hauptdatenbankdatei geschrieben werden. Die WAL-Dateien dienen als Transaktions-Log und ermöglichen eine effiziente und sichere Aktualisierung der SQLite-Datenbanken, insbesondere in Situationen mit gleichzeitigen Schreibvorgängen.» Inhalte werden erst in WAL Datei geschrieben, dann in SQLite Datei - Deshalb: WAL Änderungen in SQLite Datei schreiben, danach sqldiff mit gleicher Datenbank ohne WAL Änderungen -> Unterschiede in Excel-Tabelle 1) über sqlite3 Kommandozeile .sqlite Datenbank öffnen: 2) Über Pragma Befehl WAL Checkpoint durchführen:

4.3.2 Uncommon Locations

Üngewöhnliche Speicherorte beziehen sich auf Verzeichnisse oder Orte, die nicht zu den gängigen oder standardmäßigen Speicherorten gehören.- „Unbekannte Speicherorte“, nur durch tiefgehende forensische Analyse entdeckt Blackbox-Analyse: [2] (Stringsuchen im gesamten Image mithilfe von Tool) = Durchsuchung des Beweismaterials ohne Vorwissen über Browserverhalten (d.h. welche Dateien geschrieben wurden) sowie ohne Vorverarbeitung der Dateien (z.B. Entpacken von Dateien). Stattdessen: Untersuchen der Images nur mittels vordefinierter Funktionen von Forensik-Tools, da dies schnelles erstes Mittel von Forensikern, um nach Acquisition Phase Ergebnisse zu erhalten - wichtigster Unterschied zu "Common Locations": Umgekehrte Suchrichtung => PB Artefakt (String) -> Alle Daten nach PB Artefakt durchsuchen - Eigenschaften: > Wenn manuell durchgeführt, deutlich Zeitaufwändiger > Deshalb: Unterstützung durch Forensik-Tools. > Vertrauen in die Vollständigkeit der Tools - Beispiele in der Literatur: > "\$MFT", "\$LogFile", "Favicons", "etilqs", "Manifest.json", "pagefile.sys.", "unallocated space" and "slack space" [15]

Bei Browser Forensik hier oft verwendet: Stichwortsuchen über gesamtes Speicherabbild

Wichtig: String-Treffer muss Browser zugeordnet werden können > Oft wiederkehrende Negativbeispiele in Literatur: gesamten RAM-Dump in Hexadezimaeditor geöffnet, danach: Stringsuche nach PB Artefakten. [14, 15, 21] > Problem: Gefundener String kein Indiz, dass tatsächlich Artefakt in Zusammenhang mit PB aufgetaucht ist. > Gegenbeispiel: String in Textdatei in Desktop speichern. RAM Dump enthält String.

Analyse mit Autopsy

Bei White-Box Analyse: Autopsy nur zur Dateixtraktion genutzt, hier: als konkretes forensisches Werkzeug

> Stichwortsuche über gesamtes Festplatten-Image *** TODO: Screenshot von Suchfunktion *** > Zusätzlich: Autopsy Plugins indexieren und kategorisieren bereits Dateien - Web Bookmarks - Web Cookies - Web History - Web Categories

Analyse mit Volatility

Bei White-Box Analyse: Volatility nur zur Dateixtraktion genutzt, hier: als konkretes forensisches Werkzeug

Wie oben ausführlich beschrieben: gefundener String im RAM muss Browser zugeordnet werden können -> Passendes Werkzeug = Volatility Plugin "Yarascan". (siehe Kapitel Analysewerkzeuge)

*** TODO: konkrete Yara-Rules hier einfügen *** > HTML-Fragmente: [22] > Image als Hex: [18]

> yarascan: Das Plugin "yarascan" ermöglicht die Durchführung einer YARA-basierten Malware-Erkennung im Arbeitsspeicherabbild. YARA ist eine flexible Regel-Engine, die verwendet werden kann, um nach bestimmten Mustern, Signaturen oder Verhaltensmerkmalen von Malware zu suchen. Das Plugin wendet YARA-Regeln auf den Arbeitsspeicher an und gibt potenzielle Treffer aus. Wichtig: yara_rules.yara = In domainspecific Yara-Skriptsprache geschriebene Regeln, nach welchen RAM durchsucht wird. Hier: Einfache String-Pattern, die jeden Schritt des Browsing-Szenarios abdecken: *** TODO Beispielhafte Yara-Rules einfügen *** *** TODO: Erklärung Yara-Ausgaben, mit Screenshot *** => Wichtig hier: Gibt PID des Prozesses und virtuelle Adresse des gefundenen Strings im Speicherbereich des Prozesses an

> Davon ausgehend: pslist: Das Plugin "pslist" wird verwendet, um eine Liste der aktiven Prozesse im Arbeitsspeicherabbild zu erstellen. Es extrahiert Informationen wie Prozessnamen, PID (Prozess-ID), Elternprozess, Startzeitpunkt, Priorität und andere Attribute für jeden laufenden Prozess. => Wichtig hier: Prozessname

> SString Kontext"herausfinden = Prüfen, ob in Speicherseite über und unter gefundenen String weitere Zusammenhänge zu erkennen sind. > Dazu: memmap: Das Plugin "memmap" dient dazu, eine detaillierte Karte des physischen Speichers des Systems zu erstellen. Es gibt Informationen über die physischen Speicherbereiche, deren Startadressen, Größen, Schutzattribute und andere relevante Details. Dieses Plugin kann bei der Analyse von Speicherlayout und -nutzung hilfreich sein. 1) gibt Abbildung der Adressen des virtuellen Speichers auf die physikalische Adresse des Speicherbereichs des Prozesses mit der PID <PID>. Weiterhin und hier relevant: gibt Abbildung von virtueller Adresse auf

einen Datei-Offset an. Dieser bezieht sich auf die Datei, die erzeugt wird, wenn der Befehl mit dem `-dump` Flag ausgeführt wird: 2) führt dies zur physischen Extraktion des Speicherbereichs des Prozesses mit der PID <PID> in eine separate Datei. => Extrahierte Seite mit Hexeditor HxD untersuchen

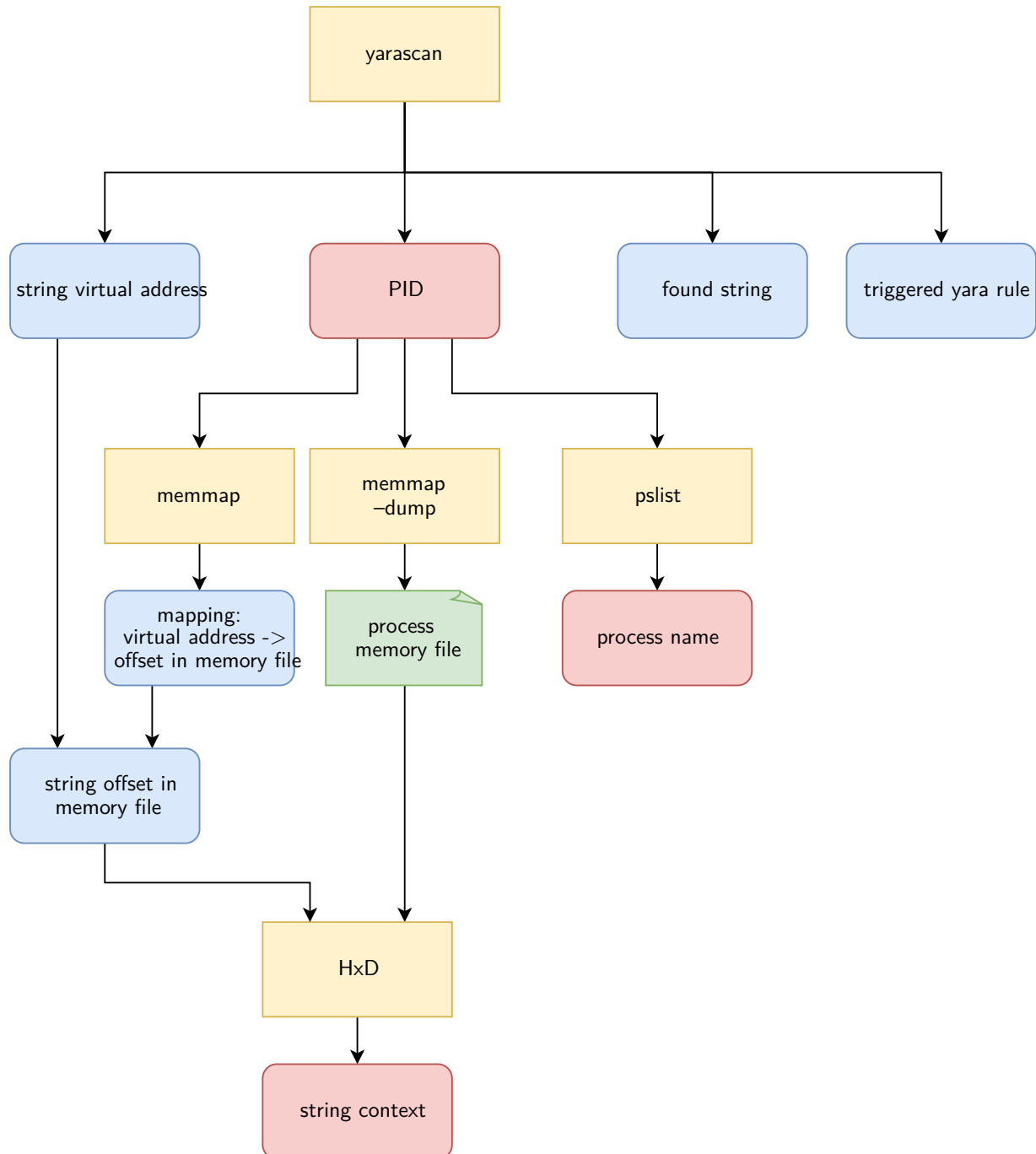


Abbildung 4.8: TODO: Process Monitor Write Operation to Excel Spreadsheet

4.3.3 Registry

- zählt sowohl Common als auch Uncommon Location -> Deshalb eigene Kategorie - Common: Analyse der Process Monitor Schreiboperationen in Registry Datenaufbereitung für jede Logfile: - Grundlage = Process Monitor Logfiles - Für Common Locations Filter setzen: > Nur "Registry Activity» Process

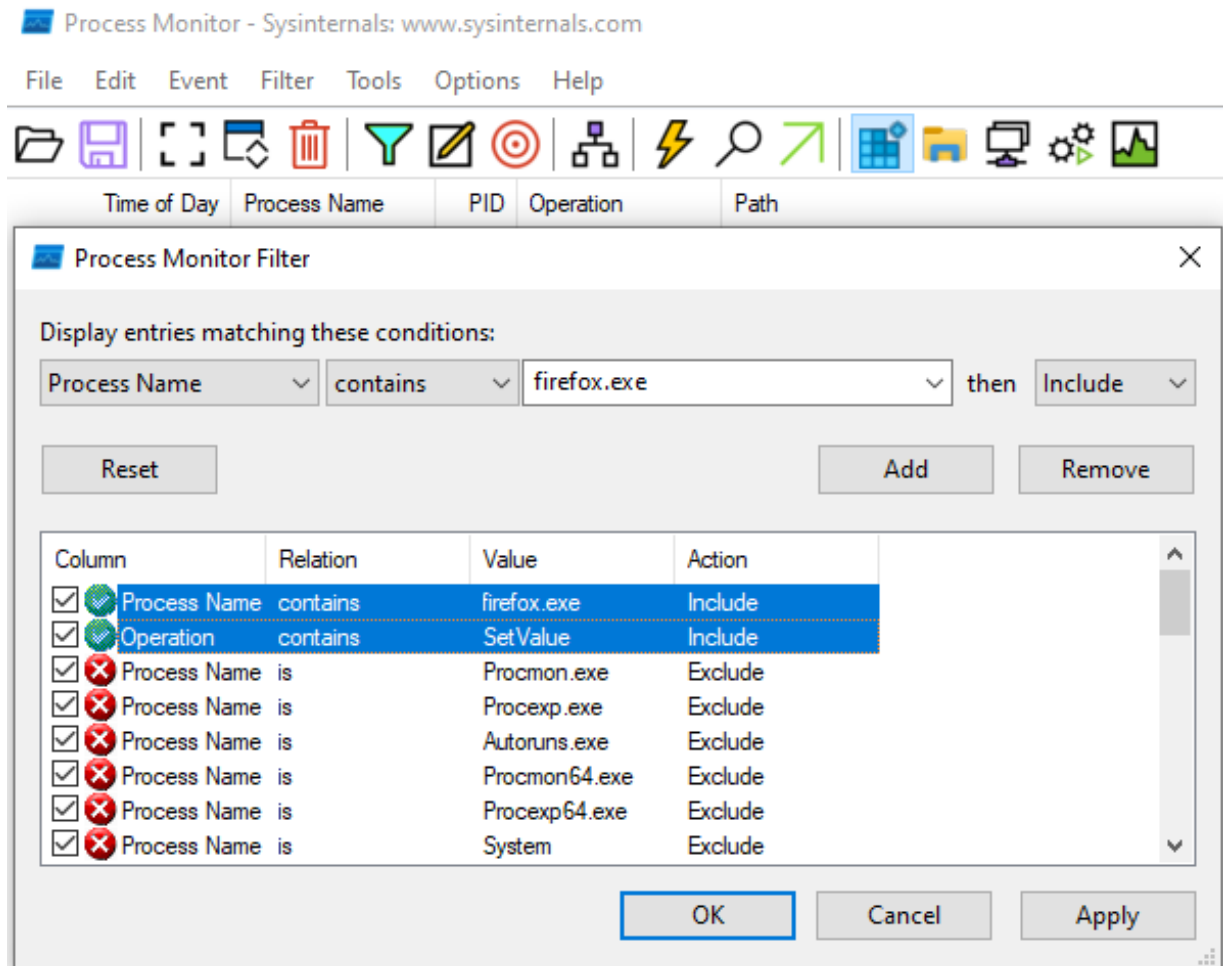


Abbildung 4.9: Tabelle mit wiederherstellbaren Dateien: Logfile 1 vs. Logfile 2

Name: contains "Browser-Prozess".exe - Firefox = firefox.exe - Tor = firefox.exe und tor.exe - Chrome = chrome.exe - Brave = brave.exe > Operation: contains SSetValue Gefiltertes Logfile als CSV exportieren, dann in Excel öffnen - Gleiche Operationen (Duplikate) löschen - Browserspezifisches Gruppieren Datenanalyse für jeden geschriebenen Registry Key: Value untersuchen (steht in CSV hinter jeder Key Schreiboperation)

- Uncommon: Stringsuche in bekannten Registry Hives mit Registry Explorer

Ein Registry Hive ist eine Datei in der Windows-Registry, die als Container für spezifische Arten von Konfigurations- und Nutzerdaten dient. Jedes Hive hat eine bestimmte Funktion, wie die Speicherung von Systemeinstellungen (System-Hives) oder individuellen Benutzerkonfigurationen (User-Hives). Die

Hives im System32

Config-Verzeichnis enthalten wichtige Informationen für das Betriebssystem, während die User-Hives unter dem jeweiligen Benutzerverzeichnis gespeichert werden. Diese Hives werden von Windows beim Start geladen und dienen als Quelle für Einstellungen und Informationen, die von verschiedenen Systemkomponenten und Anwendungen genutzt werden."

Quelle für Hives:

System-Hives im Verzeichnis ===== DEFAULT: Enthält die Standardkonfigurationseinstellungen für neue Benutzerprofile. SAM: Enthält die Sicherheitskontenverwaltungsdaten, einschließlich der Benutzerkonten und deren Kennwörter. SECURITY: Enthält Sicherheitsinformationen, die für die Zugriffssteuerung und Authentifizierung verwendet werden. SOFTWARE: Enthält Konfigurationsdaten für installierte Software und Anwendungen. SYSTEM: Enthält Systemkonfigurationseinstellungen und Gerätetreiberinformationen.

User-Hives im Verzeichnis ===== NTUSER.DAT: Enthält die individuellen Einstellungen und Konfigurationen für den angemeldeten Benutzer. USERCLASS.DAT: Enthält Informationen über die Dateizuordnungen und Registrierungseinstellungen für den angemeldeten Benutzer im Zusammenhang mit seinen benutzerdefinierten Klassen.

Analyse der Hives für jeden VM-Snapshot: Alle Hives in eine Registry Explorer Sitzung, danach Stringsuche nach PB Artefakten

5 Ergebnisse

*** TODO ***

5.1 Firefox

Im nachfolgenden Abschnitt werden die Ergebnisse der Datenanalyse für den Webbrowser Firefox detailliert beschrieben. Die Analyse ist in drei Hauptkategorien unterteilt: Common Locations, Uncommon Locations und Registry.

Common Locations

Zunächst werden die standardmäßigen Speicherorte für Browserartefakte nach potentiellen privaten Browsing Artefakten untersucht. Diese Common Locations beziehen sich ausschließlich auf Dateien, die auf die Festplatte geschrieben werden. In diesem Versuch wird gemäß Methodik in Kapitel X (TODO!) zwischen Schreiboperationen aus den Process Monitor Logfiles und SQLite Datenbanken zur Verwaltung von Nutzerdaten unterschieden.

Process Monitor WriteFile Operations

Gemäß Versuchsdurchführung in Abbildung X (TODO!) wurden für Firefox mit dem Process Monitor Tool zwei Logfiles erstellt. Diese Dateien enthalten alle aufgezeichneten Prozessaktivitäten während und nach dem Browsing Szenario. Zunächst werden die beiden Logfiles gemäß Methodik in Kapitel X (TODO!) in Excel aufbereitet. Im Anhang X (TODO!) ist dazu eine Tabelle mit allen in den gefilterten Logfiles identifizierten Dateien aufgeführt. Dabei wurde für jede Datei vermerkt ob und wie sie wiederherstellbar war, mit welchem Tool die Datei analysiert wurde und ob PB Artefakte enthalten sind.

Abbildung X (TODO!) zeigt diese Tabelle in reduzierter Darstellung. Dazu wurden ausschließlich wiederherstellbare Dateien aufgeführt. Die Dateien wurden in die fünf Kategorien "Cache", "datareporting", "SSessionstore-Backup" und "SSonstige" Dateien eingeordnet. Für jede Datei wurde vermerkt, ob in der entsprechenden Logfile PB Artefakte geschrieben wurden. Dies trifft für keine der identifizierten Dateien zu.

Bei detaillierter Untersuchung der Dateien, können zwei Pfade identifiziert werden, in die Firefox während des Versuchs Dateien schreibt. Nur die Dateien in der Cache Kategorie sind im Local Pfad gespeichert.

		Logfile 1	Logfile 2
Cache	cache2\entries\037778A55E1B7E9BED3390289866D09402D6C913	Keine PB Artefakte	Keine Schreiboperation
	cache2\entries\1223A0378B8971FA4CD25EA1731C80B2B1676B42	Keine PB Artefakte	Keine Schreiboperation
	cache2\entries\250EE2BC03AFF526F1A1C3DB212A79DE3EB60D5E	Keine PB Artefakte	Keine Schreiboperation
	jumpListCache\ZKJGVJPzPe7w4w0KwEY0jw==.ico	Keine PB Artefakte	Keine Schreiboperation
	cache2\index.log	Keine Schreiboperation	Keine PB Artefakte
Datareporting	cache2\index	Keine Schreiboperation	Keine PB Artefakte
	datareporting\glean\events\pageload	Keine PB Artefakte	Keine Schreiboperation
	*datareporting\glean\db\data.safe	Keine PB Artefakte	Keine PB Artefakte
	*datareporting\archived\2023-05\1683405837882.9102466b-e465-4ecb-810f-74ae90c64c63.new-profile.jsonlz4	Keine Schreiboperation	Keine PB Artefakte
	*datareporting\archived\2023-05\1683405837905.86f4c992-6329-415b-8c29-911a2d4b7f9d.event.jsonlz4	Keine Schreiboperation	Keine PB Artefakte
SQLite	*datareporting\archived\2023-05\1683405837939.abf8b065-41a4-4e94-a044-1cead61e396a.main.jsonlz4	Keine Schreiboperation	Keine PB Artefakte
	storage\permanent\chrome\idb\3870112724rsegmnolttet-es.sqlite	Keine PB Artefakte	Keine Schreiboperation
	storage\permanent\chrome\idb\1657114595AmcateivrtiSty.sqlite	Keine PB Artefakte	Keine PB Artefakte
	places.sqlite	Keine PB Artefakte	Keine PB Artefakte
	cookies.sqlite	Keine PB Artefakte	Keine Schreiboperation
	formhistory.sqlite	Keine PB Artefakte	Keine Schreiboperation
	webappsstore.sqlite	Keine Schreiboperation	Keine PB Artefakte
	favicons.sqlite	Keine PB Artefakte	Keine Schreiboperation
	storage.sqlite	Keine PB Artefakte	Keine Schreiboperation
Sessionstore	*sessionstore-backups\recovery.jsonlz4	Keine PB Artefakte	Keine PB Artefakte
Sonstige	prefs-1.js	Keine PB Artefakte	Keine PB Artefakte
Dateien	*xulstore.json	Keine PB Artefakte	Keine PB Artefakte

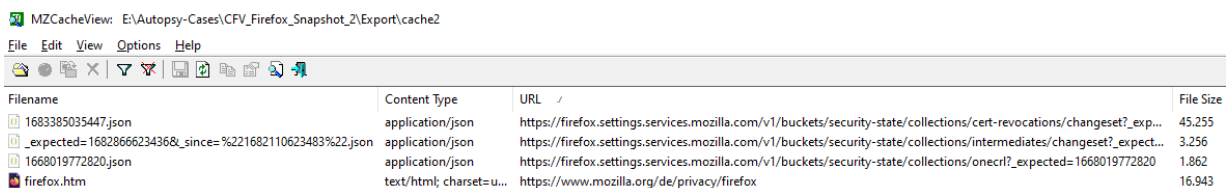
Abbildung 5.1: Tabelle mit wiederherstellbaren Dateien: Logfile 1 vs. Logfile 2

Local C:\Users\<User>\AppData\Local\Mozilla\Firefox\Profiles\<Profile>.default-release\

Roaming C:\Users\<User>\AppData\Roaming\Mozilla\Firefox\Profiles\<Profile>.default-release\

In Tabelle X (TODO!) sind die Dateien je nach Speicherort "Local"(Hellblau) oder "Roaming"(Dunkelblau) entsprechend eingefärbt.

Cache Firefox verwendet den Cache, um Webseiten und deren Ressourcen temporär lokal zu speichern. Dadurch können wiederholte Anfragen an den Server vermieden und die Ladezeiten verringert werden. Die Inhalte dieser Dateien sind binär. Die Dateien im Format \cache2\entries\<ID> werden dem Cache zugeordnet und im Local Pfad gespeichert. Wie in Kapitel X beschrieben, können diese Dateien mit dem Tool MZCacheView eingelesen werden. Wie in Abbildung X gezeigt, konnten im Cache-Ordner im zweiten Snapshot drei JSON Dateien identifiziert werden. Dabei handelt es sich um Zertifikatsdateien, die von der Öne Certificate Revocation List stammen, ein Mechanismus von Firefox zur Überprüfung von Zertifikaten. In keinem der Zertifikate konnten mit HxD private Browsing Artefakte oder besuchte Seiten gefunden werden. Weiterhin befindet sich im Cache das HTML-Dokument der Firefox Datenschutzseite, welche sich beim ersten Start des Browsers automatisch öffnete. Weitere Cache Dateien konnten in keinem Snapshot gefunden werden. Die Indexdatei



Filename	Content Type	URL	File Size
1683385035447.json	application/json	https://firefox.settings.services.mozilla.com/v1/buckets/security-state/collections/cert-revocations/changeset?_exp...	45,255
_expected=1682866623436&_since=%221682110623483%22.json	application/json	https://firefox.settings.services.mozilla.com/v1/buckets/security-state/collections/intermediates/changeset?_expect...	3,256
1668019772820.json	application/json	https://firefox.settings.services.mozilla.com/v1/buckets/security-state/collections/onecrl?_expected=1668019772820	1,862
firefox.htm	text/html; charset=u...	https://www.mozilla.org/de/privacy/firefox	16,943

Abbildung 5.2: Tabelle mit wiederherstellbaren Dateien: Logfile 1 vs. Logfile 2

\cache2\index dient als Datenbank im Cache. Sie ermöglicht dem Firefox-Browser, schnell auf die zwischengespeicherten Ressourcen zuzugreifen und diese effizient zu verwalten. Sowohl mit HxD als auch dem Tool FirefoxCache2 konnten keine PB Artefakte identifiziert werden. Schließlich enthält

die Datei `\jumpListCache\ZKJGVJPzPe7w4w0KwEY0jw==.ico` ein 64x64 Pixel großes Mozilla Logo. Dieses Logo ist keinem Schritt aus dem Browsing Szenario zuzuordnen

Datareporting Dateien im Ordner `\datareporting\glean\db` sind Teil des Glean-Systems, das für die Sammlung von Telemetriedaten und deren Übermittlung an Mozilla verwendet wird. Die Datei `data.safe.bin` enthält verschlüsselte und anonyme Informationen über die Nutzung des Browsers. In HxD konnten keine PB Artefakte gefunden werden. Dateien im Foremat `\datareporting\glean\db\<Profilname>.jsonlz4` speichern Informationen über das Firefox-Profil, das von Glean verwendet wird. Wie in Kapitel X beschrieben, lassen sich Dateien, im proprietären *jsonlz4*-Format mit dem Tool *dejsonlz4* dekomprimieren. Die entstandene JSON Datei wird mit dem Notepad++ JSON Plugin untersucht. Dabei konnten keine PB Artefakte gefunden werden.

Sessionstore Die Datei `\sessionstore-backups\recovery.jsonlz4` enthält eine Sicherungskopie der vorherigen Sitzung. Sie wird erstellt, wenn der Firefox-Browser nach einem Absturz oder einem unerwarteten Beenden neu gestartet wird." Jefferson Scher entwickelte ein Online-Tool zur Analyse von *Sessionstore-Backup* Dateien. In der Sitzungswiederherstellung konnten wie in Abbildung X gezeigt lediglich die automatisch geöffnete Seite über Firefox Datenschutzhinweise identifiziert werden.

Closed Window 1

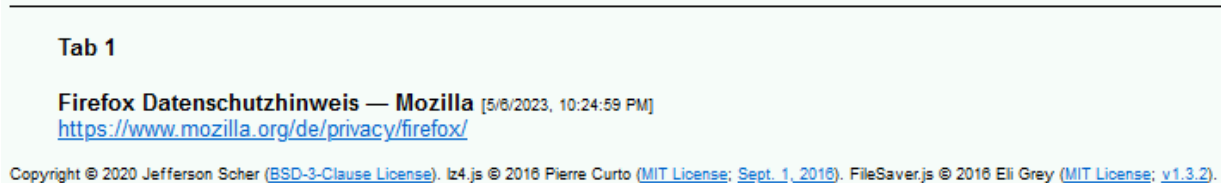


Abbildung 5.3: Tabelle mit wiederherstellbaren Dateien: Logfile 1 vs. Logfile 2

Sonstige Dateien In der Datei `prefs-1.js` werden benutzerspezifische Einstellungen und Konfigurationen für den Firefox-Browser gespeichert. Die Datei enthält Präferenzen des Benutzers in Form von JavaScript-Objekten. Es konnten mit HxD keine PB Artefakte gefunden werden. Schließlich speichert die Datei `xulstore.json` benutzerspezifische Anpassungen und Konfigurationen für den Firefox-Browser. In der Datei konnten mit Notepad++ keine PB Artefakte gefunden werden.

SQLite Datenbanken

Wie in Kapitel X (Methodik, TODO!) erwähnt, werden SQLite Datenbanken als Datenstrukturen für Nutzerdaten genauer untersucht. Mithilfe der Process Monitor Logfiles wurden die in Tabelle X dargestellten SQLite-Datenbanken für Firefox identifiziert:

Jede dieser Datenbanken wurde in allen vier Snapshots miteinander verglichen. Die Dateieextraktion und Dateianalyse erfolgte analog zur Methodik in Kapitel X (TODO!). Die Ergebnisse wurden in Tabelle X (TODO!) dargestellt.

Datenbank	Gespeicherte Daten
places.sqlite	Informationen über Lesezeichen und Verlauf. Zu jeder besuchten Webseite: URL, Seitentitel, Zeitstempel des Besuchs etc.
cookies.sqlite	Von besuchten Webseiten verwendete Cookies.
storage.sqlite	Diverse Webdaten, z. B. Indexed-Datenbanken, Offline-Cache-Daten und andere lokale Speicherinformationen.
favicons.sqlite	Enthält Favicons (kleine Symbole in der Adressleiste) um besuchte Webseiten visuell zu identifizieren.
webappsstore.sqlite	Speichert Informationen über installierte Webanwendungen im Firefox-Browser, z.B. Berechtigungen und Einstellungen.
1657114595AmcateirvtiSty.sqlite	Datenspeicher für Activity Stream, eine personalisierte Übersicht über Browser-Aktivitäten beim Öffnen eines neuen Tabs.
3870112724rsegmnoittet-es.sqlite	Datenspeicher für Remote Settings, eine zentrale Verwaltung von benutzerspezifischen Browsereinstellungen.

	File	Snapshot 1: Browser Installation	Snapshot 2: After Browsing Scenario, Browser open		Snapshot 3: After Browsing Scenario, Browser closed		Snapshot 3: Browser closed	
		Installation	Vor WAL	Nach WAL	Vor WAL	Nach WAL	Vor WAL	Nach WAL
SQLite	places.sqlite	N/A	Initialisiert, Zeilen, Einträge für exist. geöffnete Seiten	no diff	Indizes bei vorhandenem Seiten aktualisiert	no diff	no diff	no diff
	cookies.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer
	storage.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer
	favicons.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer
	webappsstore.sqlite	N/A	N/A	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer
	formhistory.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer
	1657114595AmcateirvtiSty.sqlite	N/A	Initialisiert, 1 Zeile: "origin: chrome"	no diff	Einträge (Binärdaten) eingefügt, keine PB Artefakte (HxD)	no diff	no diff	no diff
	3870112724rsegmnoittet-es.sqlite	N/A	Initialisiert, 1 Zeile: "origin: chrome"	no diff	no diff	no diff	no diff	no diff
			Leer					
			Unverändert (nicht-leer)					
			Neuer (nicht-leerer) Inhalt					

Abbildung 5.4: Comparison of found PB artifacts between RAM Dumps

Nach Browser-Installation (Snapshot 1) existierte noch keine der SQLite-Dateien.

Nach dem Browsing Szenario (Snapshot 2) wurde festgestellt, dass alle SQLite-Datenbanken initialisiert wurden, außer webappsstore.sqlite. Dabei wurden in places.sqlite die automatisch im normalen Modus geöffnete Datenschutzhinweise Seite eingetragen. In restlichen Datenbanken wurden leer initialisiert, nur die Spaltennamen wurden eingetragen. Der Inhalt aller erstellten Datenbanken blieb nach Durchführung von PRAGMA WAL Checkpoints unverändert.

Nach Schließen des Browsers (Snapshot 3) wurden in places.sqlite die Indizes bei eingetragenen Seiten aktualisiert. Die SQLite-Datenbank 1657114595AmcateirvtiSty.sqlite erhielt ein binäres Datenobjekt als Eintrag. Bei der Untersuchung mit HxD konnten keine Artefakte gefunden werden. Weiterhin wurde webappsstore.sqlite leer initialisiert. Die restlichen Daten blieben im Vergleich mit Snapshot 2 unverändert. Ebenfalls veränderte sich nicht der Inhalt nach Durchführung von PRAGMA WAL Checkpoints.

Nach herunterfahren der VM (Snapshot 4) gab es keine Änderungen in den SQLite Datenbanken, auch nach Durchführung der PRAGMA WAL Checkpoints.

Somit wurden in den SQLite Datenbanken von Firefox keine zurückverfolgbaren PB Artefakte im privaten Modus hinterlassen.

Mithilfe des Process Monitors wurde festgestellt, dass sowohl während des Browsing Szenarios (Logfile 1) als auch danach (Logfile 2) Inhalte in Dateien geschrieben wurden. Wie zusammenfassend in Abbildung X (TODO!) dargestellt, wurde mit Ausnahme der Datareporting Dateien gab es in Logfile 1 stets mehr oder genauso viele Schreiboperationen wie in Logfile 2. Keine Schreiboperation hinterließ jedoch Private Browsing Artefakte.

Uncommon Locations

Nachfolgend werden die Analyseergebnisse der Firefox Uncommon Locations beschrieben. Wie in Kapitel X erläutert, wird im Gegensatz zu Common Locations die Suchrichtung umgekehrt und es werden alle gesammelten Daten nach einem spezifischen PB Artefakt durchsucht. Somit benötigt ein

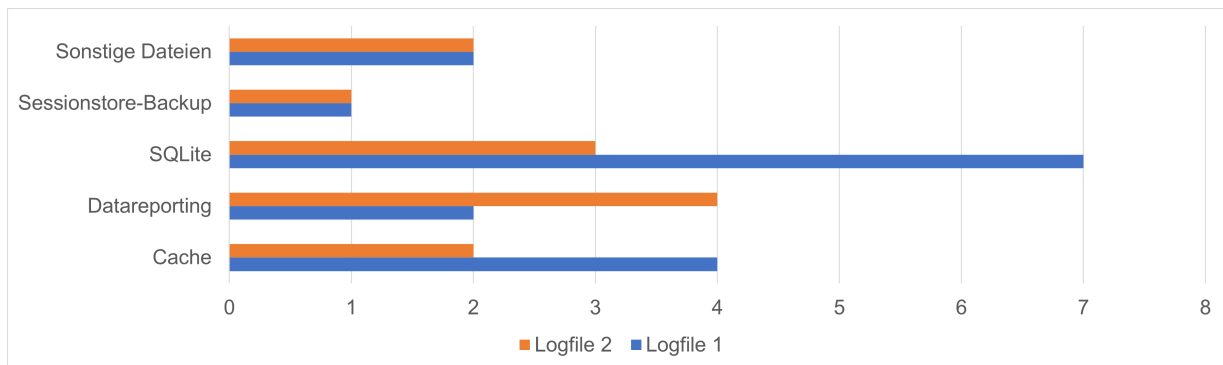


Abbildung 5.5: Comparison of found PB artifacts between RAM Dumps

Forensiker kein Wissen über das Browserverhalten. Stattdessen wird sich auf die Vollständigkeit der Funktionen von Forensik-Tools verlassen. Im Rahmen dieses Versuchs werden die Tools Autopsy und Volatility verwendet.

Analyse mit Autopsy

Bei den Common Locations in Kapitel X wird Autopsy nur zur Dateieextraktion genutzt. Im Falle der Uncommon Locations dient Autopsy als forensisches Werkzeug zur Datenanalyse.

Eine Autopsy Stichwortsuche gemäß Methodik in Kapitel X (TODO!) lieferte in allen Snapshots keine Treffer. Es wurde zusätzlich das \$Carved Verzeichnis durchsucht, in dem Autopsy alle wiederhergestellten Dateien speichert.

Anschließend wurden die automatisch von Autopsy kategorisierten Dateien untersucht. Gemäß Methodik in Kapitel X wurden dazu die Dateien der Kategorien "Web Bookmarks", "Web Cookies", "Web History" sowie "Web Categories" analysiert. Beim Vergleich der Festplattenabbilder wurde festgestellt, dass ein Snapshot stets die kategorisierten Dateien des vorherigen Snapshots enthielt. Es sind innerhalb einer Kategorie nur neue Dateien dazugekommen. Somit enthält Snapshot 4 in jeder Kategorie alle Dateien der vorherigen Snapshots.

Web Bookmarks Bereits vor Durchführung des Browsing Szenarios enthielt Firefox im ersten Snapshot die Bing Startseite als gespeichertes Lesezeichen. In den restlichen Snapshots 2 – 4 blieb diese Kategorie unverändert.

Source Name	S	C	O	URL	Title	△ Date Created	Program Name	Domain	Data Source
Bing.url			19	http://go.microsoft.com/fwlink/?LinkId=255142	Bing.url	2023-04-25 16:09:28 MESZ	Internet Explorer Analyzer	microsoft.com	CFV_Firefox_klon_Snapshot_3.img

Abbildung 5.6: Autopsy Web Bookmarks

Web Cookies Auch diese Kategorie enthält bereits vor Beginn des Browsing Szenarios zehn Cookie-Einträge in der Datei WebCacheV01.dat. Dabei handelt es sich um eine Datenbank des Microsoft Edge Browsers zur Speicherung von Nutzerdaten. Diese Datei verhält sich ähnlich wie die in diesem

Versuch relevanten SQLite-Dateien. Die Datei enthält. Bei den Einträgen handelt es sich um Cookies für Bing und die Outlook Webseite, obwohl diese Seiten nie in Microsoft Edge geöffnet wurden. In den Snapshots 2 – 4 kamen keine weiteren Einträge in dieser Kategorie hinzu.

Source Name	S	C	O	URL	Date Accessed	Name	Value	Program Name	Domain	Data Source
WebCacheV01.dat			15	bing.com	2023-05-06 19:50:17 MESZ	SUID	M	Microsoft Edge Analyzer	bing.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	www.bing.com	2023-05-06 19:51:24 MESZ	MUIDB	31708C5FC3CF47068AFADC1CB47D0111	Microsoft Edge Analyzer	bing.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	bing.com	2023-05-06 19:50:17 MESZ	SRCHD	AF=NOFORM	Microsoft Edge Analyzer	bing.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	bing.com	2023-05-06 19:50:17 MESZ	SRCHUID	V=26GJID=B2C50ADB8B94234A9FE14DB81DCB91D8dm...	Microsoft Edge Analyzer	bing.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	bing.com	2023-05-06 19:50:17 MESZ	SRCHUSR	DOB=20230506	Microsoft Edge Analyzer	bing.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	bing.com	2023-05-06 19:50:20 MESZ	SRCH#PGLSR	SRCHLANG=de&LUT=16834026192238JPMH=dee20405&I...	Microsoft Edge Analyzer	bing.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	bing.com	2023-05-06 19:50:19 MESZ	CortanaAppUID	C164AA3A4D7E127DDC66AD915CFD04C	Microsoft Edge Analyzer	bing.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	bing.com	2023-05-06 19:55:22 MESZ	ANON	A=A3B5B679A14D59B0AA027635FFFFFFF	Microsoft Edge Analyzer	bing.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	live.com	2023-05-06 19:50:30 MESZ	MUID	118A534093A9626528C5404997A966B8	Microsoft Edge Analyzer	live.com	CFV_Firefox_Klon_Snapshot_3.img
WebCacheV01.dat			15	login.live.com	2023-05-06 19:51:06 MESZ	_Host-MSAAUTHP		Microsoft Edge Analyzer	live.com	CFV_Firefox_Klon_Snapshot_3.img

Abbildung 5.7: Autopsy Web Cookies

Web History Diese Kategorie listet alle Dateien mit gespeichertem Suchverlauf auf. Vor Beginn des Browsing Szenarios (Snapshot 1) enthält die Kategorie ebenfalls zwei Einträge zur Outlook Webseite in der Datei WebCacheV01.dat. Nach Durchführung des Browsing Szenarios (Snapshot 2) wurde ein Eintrag in der places.sqlite Datenbank hinzugefügt. Dabei handelt es sich um die automatisch im normalen Browsingmodus geöffnete Firefox-Standardseite über Datenschutzhinweise. Dies deckt sich mit den Beobachtungen der Common Locations in Kapitel X. Darüber hinaus enthält dieser Snapshot für die Datei WebCacheV01.dat den Eintrag file:///Z:/Logfile_1. Dabei handelt es sich um das Process Monitor Logfile, das gemäß Methodik in Kapitel X (TODO!) über den gemeinsamen VM-Ordner zum Analyse-Rechner transportiert wurde. Ergänzt wird das in Snapshot 3 durch den Eintrag file:///Z:/Logfile_2, dem zweiten Process Monitor Logfile. In Snapshot 4 werden in dieser Kategorie keine neuen Dateien erfasst.

Source Name	S	C	O	URL	Date Accessed	Referrer URL	Title	Program Name	Domain	Data Source	Username
places.sqlite			6	https://www.mozilla.org/de/privacy/firefox/	2023-05-06 22:25:00 MESZ	https://www.mozilla.org/privacy/firefox/	Firefox Datenschutzhinweis — Mozilla	Firefox Analyzer	mozilla.org	CFV_Firefox_Klon_Snapshot_3.img	
WebCacheV01.dat			15	https://login.live.com/oauth20_desktop.aspx?lc=1031	2023-05-06 19:51:06 MESZ			Microsoft Edge Analyzer	live.com	CFV_Firefox_Klon_Snapshot_3.img	Forensik
WebCacheV01.dat			15	https://login.live.com/oauth20_authorize.aspx?client_j...	2023-05-06 19:51:08 MESZ			Microsoft Edge Analyzer	live.com	CFV_Firefox_Klon_Snapshot_3.img	Forensik
WebCacheV01.dat				file:///Z:/Logfile_1	2023-05-06 20:29:36 MESZ			Microsoft Edge Analyzer		CFV_Firefox_Klon_Snapshot_3.img	Forensik
WebCacheV01.dat				file:///Z:/Logfile_2	2023-05-06 20:44:19 MESZ			Microsoft Edge Analyzer		CFV_Firefox_Klon_Snapshot_3.img	Forensik

Abbildung 5.8: Autopsy Web History

Web Categories Diese Kategorie klassifiziert im Speicherabbild gefundene Browsing Artefakte nach Inhalt. Vor Beginn des Browsing Szenarios (Snapshot 1) werden hier bereits zwei Einträge aufgelistet. Der Eintrag bing.com wird als "Suchmaschine" klassifiziert und live.com als "Web-Email". Wie oben erwähnt, wurden beide Seiten nie aufgerufen. Es gab keine zusätzlichen Einträge in dieser Kategorie in den Snapshots 2 bis 4.

Source Name	S	C	O	Source Type	Score	Domain	Host	Name	File Path
WebCacheV01.dat			0	File	Unknown	bing.com	bing.com	Search Engine	/img_CPV_Firefox_Klon_Snapshot_3.img/vol_vol3/Users/Forensik/AppData/Local/Microsoft/Windows/WebCache/WebCacheV01.dat
WebCacheV01.dat			0	File	Unknown	live.com	login.live.com	Web Email	/img_CPV_Firefox_Klon_Snapshot_3.img/vol_vol3/Users/Forensik/AppData/Local/Microsoft/Windows/WebCache/WebCacheV01.dat

Abbildung 5.9: Autopsy Web Categories

Somit wurden in allen Kategorien ausschließlich Browsing Artefakte des Edge Browsers in der Datei WebCacheV01.dat gefunden, sowie ein Eintrag in der Firefox SQLite Datenbank places.sqlite. Die

eingetragene Firefox-Standardseite deckt sich mit den Ergebnissen der Common Locations in Tabelle X. Die aufgelisteten Einträge in der Datei WebCacheV01.dat sind nicht auf Schritte des Browsing Szenarios zurückzuführen. Die Einträge sind bereits im ersten Snapshot enthalten, obwohl vor Beginn des Browsing Szenarios keine Browseraktivitäten durchgeführt wurden. Weiterhin enthält diese Datei Einträge über die Process Monitor Logfiles, welche über einen gemeinsamen VM-Ordner zum Rechner transportiert wurde, auf dem die virtuelle Maschine läuft. In keiner der Kategorien konnten private Browsing Artefakte identifiziert werden.

Analyse mit Volatility

Nachdem die Firefox Festplattenabbilder als Uncommon Location mit Autopsy untersucht wurden, werden nachfolgend die Analyseergebnisse des RAMs als Uncommon Location beschrieben. Dazu wurde eine Stringsuche im gesamten RAM nach PB Artefakten durchgeführt. Wie in Kapitel X ausführlich beschrieben muss ein gefundener String eindeutig einem Browser zugeordnet werden können. Deshalb wurde dazu das Volatility PlugIn "Yarascan" verwendet, ein Werkzeug um nach bestimmten Mustern im RAM zu suchen. Dazu wurden die in Tabelle X aufgeführten Yara-Regeln verwendet. Wie in Kapitel Methodik (TODO!) beschrieben, wird davon ausgehend das PlugIn "pslist" verwendet, um den Prozessnamen anhand PID zu identifizieren. Die Ergebnisse dieser Stringsuche sind nachfolgend nach Kategorie geordnet.

Yararule HTML In keinem der Firefox RAM Dumps wurden HTML Fragmente der besuchten Seiten gefunden. Somit wird diese Yara-Regel nicht weiter betrachtet.

Yararule Keyword Wie in Abbildung X (TODO!) gezeigt, wurden alle Suchbegriffe "pfaffenhofen", "nanoradar", "mooserliesl" sowie "mallofamerica" identifiziert im zweiten RAM Dump, nach dem Browsing Szenario mit geöffnetem Browser, identifiziert. Die Artefakte befinden sich ausschließlich im zweiten RAM Dump. Die Suchbegriffe wurden größtenteils in den Speicherbereichen von Firefox-Prozessen gefunden. Nur in zwölf Fällen wurden Suchbegriffe in anderen Prozessen identifiziert. Am häufigsten wurde der Suchbegriff "pfaffenhofen" mit 1301 gefundenen Artefakten im zweiten Firefox RAM Dump gefunden. Dies ist vermutlich auf den Google Maps Kartenbereich zurückzuführen, einen visuellen Ausschnitt der Karte, welcher bei der Google-Suche erscheint und Informationen über die geografische Lage, Straßen, Sehenswürdigkeiten und andere relevante Orte in der gesuchten Stadt zeigt. In den RAM Dumps 1 und 3 konnten Artefakte zu den Suchbegriffen identifiziert werden.

Yararule URL Es konnten in den Arbeitsspeicherabbildern alle besuchten URLs unitree.com, mooserliesl.de, mallofamerica.com sowie donaukurier.de identifiziert werden. Dabei wurden die meisten Artefakte nach dem Browsing Szenario mit geöffnetem Browser (RAM Dump 2) gefunden. Alle besuchten URLs wurden in diesem Dump sowohl in Firefox als auch anderen Prozessen gefunden, wobei die meisten Artefakte in Firefox Prozessen zu finden sind. Dabei wurde "mooserliesl.de" mit insgesamt 390 Artefakten am wenigsten gefunden, "donaukurier.de" mit über 3600 Artefakten am häufigsten.

Bemerkenswert ist, dass URL Artefakte gefunden wurden, nachdem der Browser geschlossen wurde (RAM Dump 3). Dabei wurde kein URL Artefakte in einem Firefox Prozess gefunden. Anhand

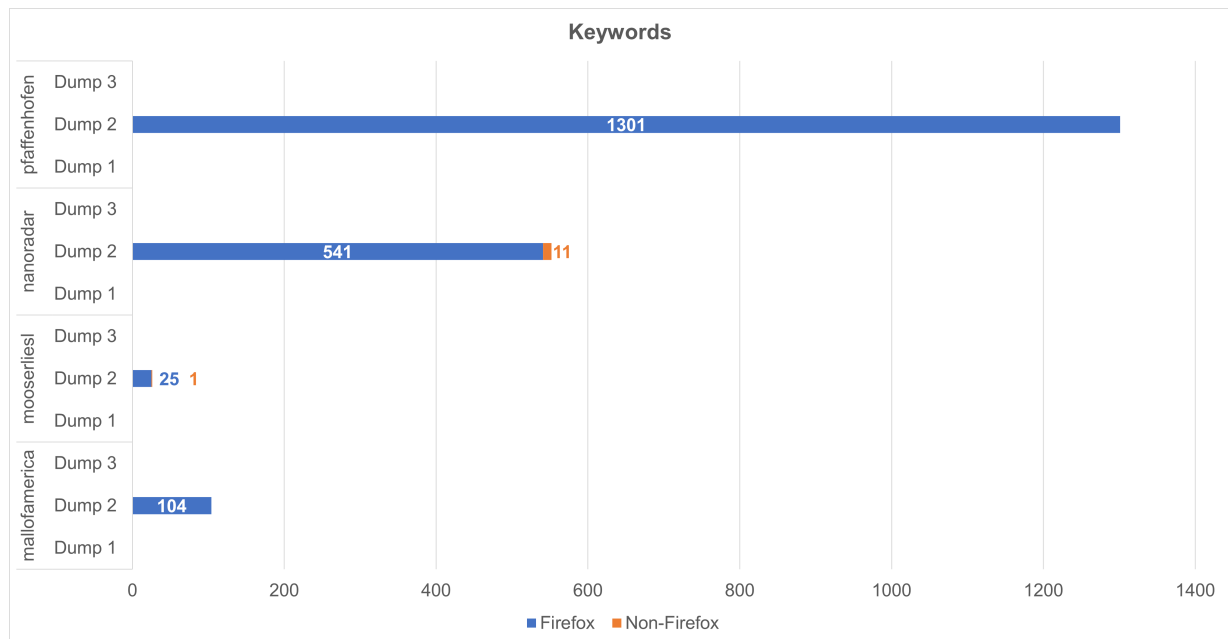


Abbildung 5.10: Keywords

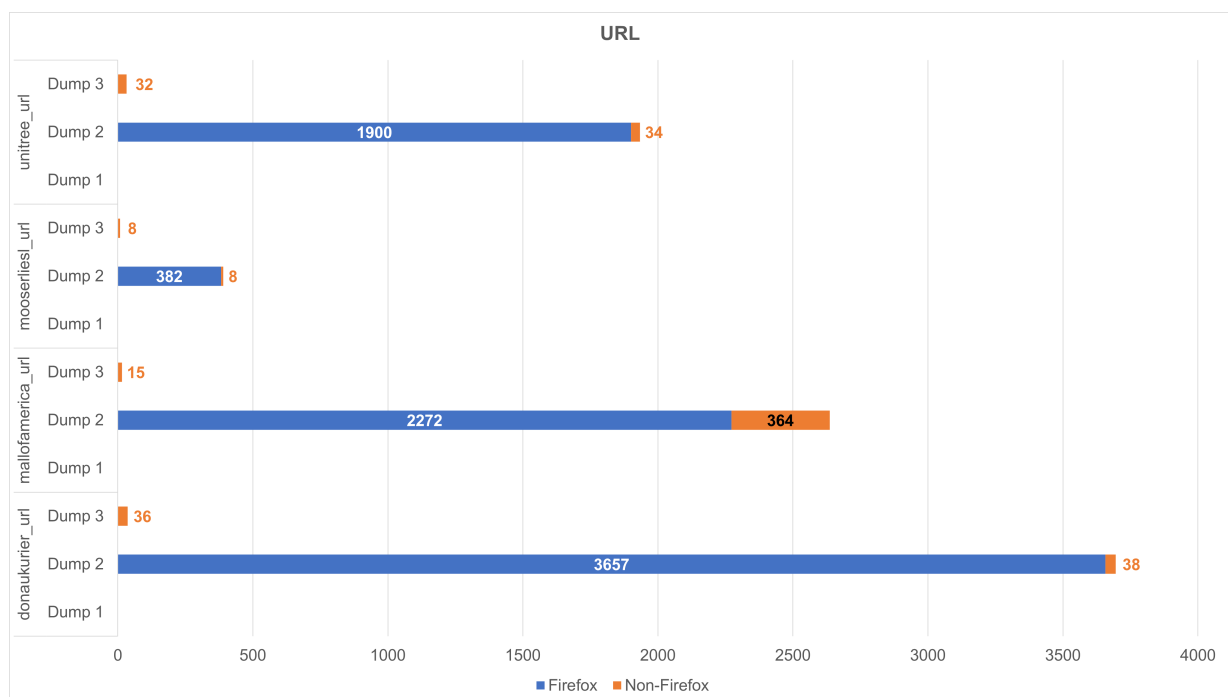


Abbildung 5.11: URL

der PID 2252 wurde festgestellt, dass sich alle URL Artefakte des dritten RAM Dumps in einem `lvchost.exe` Prozess mit der gleichen PID befinden. Unter dem `SService Host` Prozess laufen gruppierte Windows-Dienste, um Ressourcen zu sparen und die Systemleistung zu verbessern. Volatility bietet das

Virtuelle Speicheradresse	PID	Byte-Offset in extrahierter Speicherseite
0xb9ce29180c8	7420	0x11dd40c8
0x2859f4ffd4e0	7420	0x12e234e0
0x24083b41858	8424	0x583858
0x240840e5b08	8424	0x96bb08

Plugin Bvscan, mit dem alle laufenden Dienste ausgegeben werden können. Bei Anwendung auf den dritten RAM Dump wurde jedoch zu keinem Dienst eine PID angegeben, wodurch der Dienst mit den URL Artefakten nicht im RAM identifiziert werden konnte. Stattdessen wurde der dritte Snapshot aufgetaut, um im laufenden Windowsbetrieb den Dienst mithilfe des Process Explorers zu identifizieren. Wie in Abbildung X (TODO!) gezeigt, wurde anhand der PID 2252 der Dienst "DNSCache" ermittelt. Der DNSCache-Dienst unter Windows ist ein Teil des Betriebssystems, der für

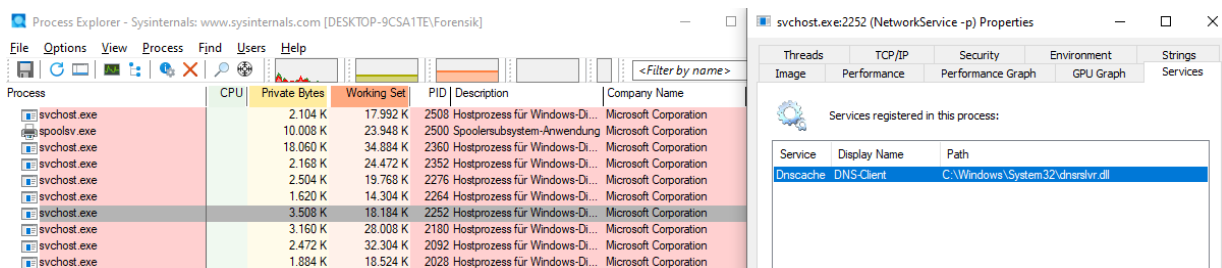


Abbildung 5.12: URL

die Übersetzung von Domainnamen in IP-Adressen verantwortlich ist. Der DNSCache-Dienst speichert DNS-Anfragen und Antworten temporär, um wiederholte DNS-Anfragen zu beschleunigen. Nach Löschen des DNSCaches mit dem Kommandozeilenbefehl `ipconfig /flushdns`, dem Schließen aller Process Monitor Instanzen sowie Beenden des DNSCaches Services wurde erneut ein RAM-Dump durchgeführt. Dort wurden keine URL Artefakte mehr gefunden.

Yararule Mail Es konnten alle E-Mail Artefakte des Browsing Szenarios gefunden werden. Die Artefakte befinden sich ausschließlich im zweiten Firefox RAM Dump, nach dem Browsing Szenario mit geöffnetem Browser. Unter den gefundenen Artefakten befindet sich mit zwölf Vorkommen am häufigsten die Absenderadresse "computerforensikvl@gmail.com". Dieses Artefakt wurde als einziges Mail-Artefakt in anderen Prozessen außer Firefox gefunden.

Bemerkenswert ist, dass das Passwort des Google-Accounts, mit dem die E-Mails verschickt wurden, vier mal als Klartext im RAM gefunden wurden. Das Passwort wurde in je zwei Firefox Prozessen mit den PIDs 7420 und 8424 zwei mal gefunden. Tabelle X zeigt die virtuellen Speicheradressen der Artefakte aus der Yarascan Ausgabe.

Zu diesen Artefakten wurde gemäß Methodik in Kapitel X der String Kontext – die Zeichen vor und nach dem gefundenen Artefakt im Speicherbereich – ermittelt. Dazu wurde gemäß Methodik in Kapitel X mithilfe des Volatility memmap-Plugins die Abbildung der virtuellen Speicheradressen auf den Byte-Offset in der extrahierten Speicherseite des Prozesses ermittelt. Wie in Abbildung X gezeigt, sind in der Speicherseite des Prozesses mit PID 7420 konnte vor und nach dem gefundenen Passwort

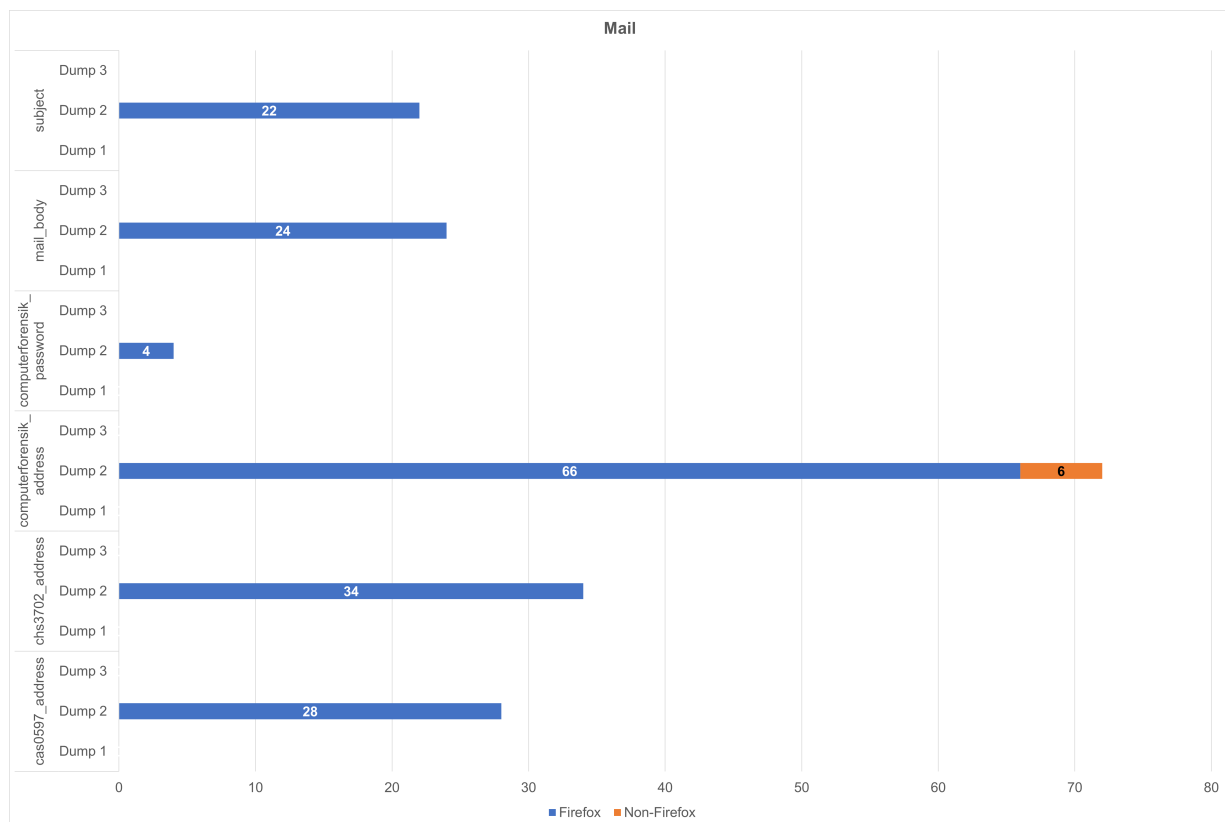


Abbildung 5.13: Mail

```

11DD4040 58 02 00 00 08 00 00 00 67 6D 70 41 64 64 6F 6E X.....gmpAddon
11DD4050 4B 4B 4B 4B 4B 4B 4B 4B DC F9 0E 50 4B 4B 4B 4B KKKKKKKKÜ.PKKKK
11DD4060 58 02 00 00 0D 00 00 00 67 6D 70 44 6F 77 6E 6C X.....gmpDownl
11DD4070 6F 61 64 65 72 4B 4B 4B 50 C3 FB EA 4B 4B 4B 4B oaderKKKPÄüKKKK
11DD4080 58 02 00 00 0D 00 00 00 47 4D 50 44 6F 77 6E 6C X.....GMPDownl
11DD4090 6F 61 64 65 72 4B 4B 4B D0 6F AE 8A 4B 4B 4B 4B oaderKKKDoöSKKKK
11DD40A0 58 02 00 00 0D 00 00 00 5F 69 73 45 4D 45 45 6E X....._isEMEEEn
11DD40B0 61 62 6C 65 64 4B 4B 4B 8F 4F 0E 73 4B 4B 4B 4B abledKKK.O.sKKKK
11DD40C0 58 02 00 00 0C 00 00 00 56 6F 72 6C 65 73 75 6E X.....Vorlesun
11DD40D0 67 32 33 21 4B 4B 4B 4B F8 35 7D 48 4B 4B 4B 4B g23!KKKK5}HKKKK
11DD40E0 58 02 00 00 0F 00 00 00 5F 69 73 41 64 64 6F 6E X....._isAddon
11DD40F0 45 6E 61 62 6C 65 64 4B 42 1D 99 C2 4B 4B 4B 4B EnabledKB.ÄÄKKKK
11DD4100 58 02 00 00 0F 00 00 00 6D 61 69 6C 2E 67 6F 6F X.....mail.goo
11DD4110 67 6C 65 2E 63 6F 6D 4B 44 47 D9 2D 4B 4B 4B 4B gle.comKDGÜ-KKKK
11DD4120 58 02 00 00 10 00 00 00 5F 75 70 64 61 74 65 4C X....._updateL
11DD4130 61 73 74 43 68 65 63 6B 43 1F 7D 4B 4B 4B 4B astCheckC.)KKKKK
11DD4140 58 02 00 00 10 00 00 00 73 65 63 6F 6E 64 73 53 X.....secondsS

```

Abbildung 5.14: Password in memory page of PID 7420 at Byte-Offset 0x11dd40c8

am Byte-Offset 0xb9ce29180c8 neben der Gmail-Url "mail.google.com" Code-Fragmente der "Gecko-Engine" finden. Dieser Teil des Firefox Browsers ist für das Rendering von Webinhalten verantwortlich, einschließlich HTML, CSS, JavaScript und anderen Medienformaten wie Bildern, Audio und Video. In der gleichen Datei konnte nach dem gefundenen Passwort am Byte-Offset 0x24083b41858 die Strings "Passwd" sowie "Sessionrestore" (siehe Common Location SSessionstore in Kapitel X) identifiziert werden. Wie in den Abbildungen X und Y (TODO!) gezeigt, können in den Byte-Offsets der gefundenen Passwörter in der Speicherseite der PID 8424 konnten kein Kontext ermittelt werden. Im Gegensatz

```

12E23470 00 00 00 00 00 00 00 00 10 02 00 00 34 00 00 00 .....4...
12E23480 00 C8 CC E5 29 02 00 00 00 00 00 00 00 00 00 00 .Eiä).....
12E23490 10 02 00 00 27 00 00 00 40 D3 CC E5 29 02 00 00 ....'...@Öiä)...
12E234A0 00 00 00 00 00 00 00 00 10 02 00 00 2A 00 00 00 .....*...
12E234B0 70 D3 CC E5 29 02 00 00 00 00 00 00 00 00 00 00 pÓiä).....
12E234C0 00 02 00 00 45 00 00 00 A0 47 7C 1A 55 23 00 00 ....E... G|.U#..
12E234D0 F8 DE FF F4 59 28 00 00 50 02 00 00 0C 00 00 00 øÞÿÖY(.P.....
12E234E0 56 6F 72 6C 65 73 75 6E 67 32 33 21 A2 1D FB FF Vorlesung23!c.üÿ
12E234F0 50 02 00 00 0A 00 00 00 69 64 65 6E 74 69 66 69 P.....identifi
12E23500 65 72 F0 B8 FA 7F 00 00 50 02 00 00 06 00 00 00 erö,ú...P.....
12E23510 50 61 73 73 77 64 F9 FF 18 96 73 E5 29 02 00 00 Passwdüÿ.-sä)...
12E23520 50 02 00 00 0E 00 00 00 73 65 73 73 69 6F 6E 72 P.....sessionr
12E23530 65 73 74 6F 72 65 00 00 10 02 00 00 2E 00 00 00 estore.....
12E23540 80 D2 CC E5 29 02 00 00 00 00 00 00 00 00 00 00 eÖiä).....
12E23550 10 02 00 00 1F 00 00 00 00 B8 CC E5 29 02 00 00 .....iä)...

```

Abbildung 5.15: Password in memory page of PID 7420 at Byte-Offset 0x12e234e0

zur Speicherseite der PID 7420 wird das Passwort dort mit 2 Bytes pro Zeichen enkodiert. Das eine Unicode-Zeichenenkodierung vermuten.

```

005837F0 02 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 .....
00583800 C0 9E EA FF 40 02 00 00 0E 00 00 00 00 00 00 00 Äzëÿ@.....
00583810 02 00 00 00 00 00 00 00 00 E5 E5 E5 E5 E5 E5 .....ääääääää
00583820 02 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 .....
00583830 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00583840 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
00583850 02 00 00 00 1A 00 00 00 56 00 6F 00 72 00 6C 00 .....V.o.r.l.
00583860 65 00 73 00 75 00 6E 00 67 00 32 00 33 00 21 00 e.s.u.n.g.2.3.!.
00583870 00 00 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 ..ääääääääääääää
00583880 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
00583890 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
005838A0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
005838B0 08 00 00 00 D7 16 71 67 01 00 00 00 00 00 00 00 ....*.qg.....
005838C0 6F 00 6E 00 44 00 51 00 30 00 4B 00 55 00 62 00 o.n.D.Q.O.K.U.b.

```

Abbildung 5.16: Password in memory page of PID 8424 at Byte-Offset 0x583858

```

0096BA80 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BA90 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BAA0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BAB0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BAC0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BAD0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BAE0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BAF0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BB00 01 00 00 00 38 00 00 00 56 00 6F 00 72 00 6C 00 .....8...V.o.r.l.
0096BB10 65 00 73 00 75 00 6E 00 67 00 32 00 33 00 21 00 e.s.u.n.g.2.3.!.
0096BB20 00 00 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 ..ääääääääääääää
0096BB30 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BB40 00 C8 93 B8 FA 7F 00 00 28 C8 93 B8 FA 7F 00 00 .E",ú... (E",ú...
0096BB50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0096BB60 04 00 00 00 00 00 00 00 80 9B 47 88 40 02 00 00 .....€>G^@...
0096BB70 2C 41 7B B8 FA 7F 00 00 00 00 00 00 00 00 00 00 ,A{,ú.....
0096BB80 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BB90 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää
0096BBA0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 .....ääääääääääääää

```

Abbildung 5.17: Password in memory page of PID 8424 at Byte-Offset 0x96bb08

Yararule Image Das im Browsing Szenario geöffnete Donaukurier Logo wurde ausschließlich im zweiten RAM Dump in drei mal in Firefox Prozessen gefunden.

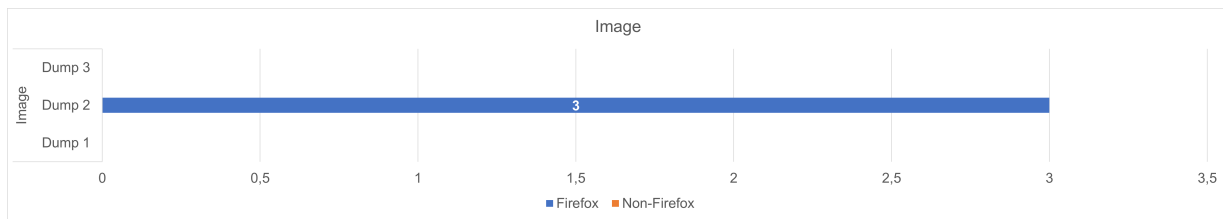


Abbildung 5.18: Image

Wie in Abbildung X zusammenfassend gezeigt wurden vor dem Browsing Szenario, keine private Browsing Artefakte im ersten RAM Dump gefunden. Nach dem Browsing Szenario mit geöffnetem Browser konnten die meisten Artefakte identifiziert werden. Dabei wurden am häufigsten URL Artefakte in Firefox Prozessen gefunden. Zudem konnte hier das E-Mail Passwort im Klartext lokalisiert werden. Nach Schließen des Browsers konnten im dritten Snapshots URLs im DNSCache Windows Service gefunden werden. Nach leeren des Caches und Beenden des DNSCache Services konnten keine Artefakte gefunden werden.

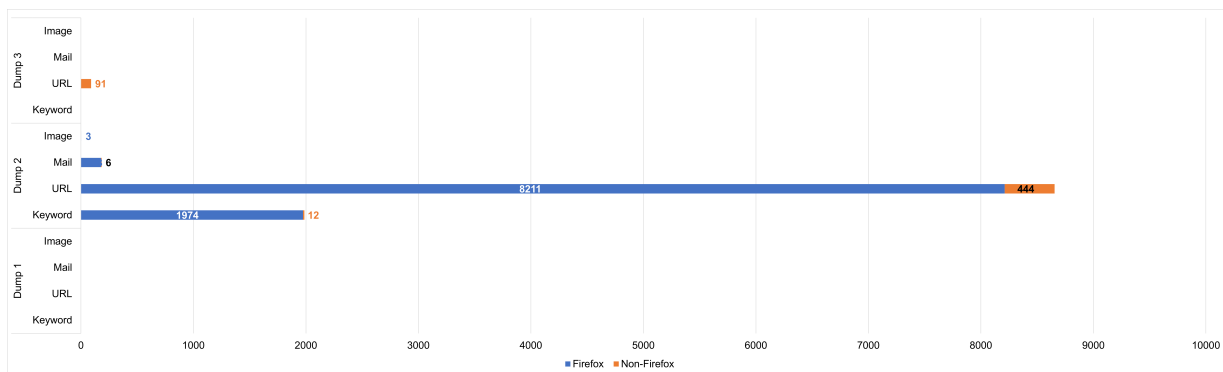


Abbildung 5.19: Summary

Registry

Die Analyse der Registry zählt gemäß Methodik in Kapitel X sowohl zu den Common als auch Uncommon Locations

Process Monitor SetValue Operations

Als Teil der Common Locations werden für Firefox alle Registry SSetValueSchreiboperationen der beiden Process Monitor Logfiles untersucht.

In beiden Logfiles wurden zwei Kategorien von Registry Keys geschrieben: "PreXULSkeletonUISettings" und "Business Activity Monitoring". In Abbildung X ist der Anteil der Schreiboperationen je Kategorie für beide Logfiles gezeigt.

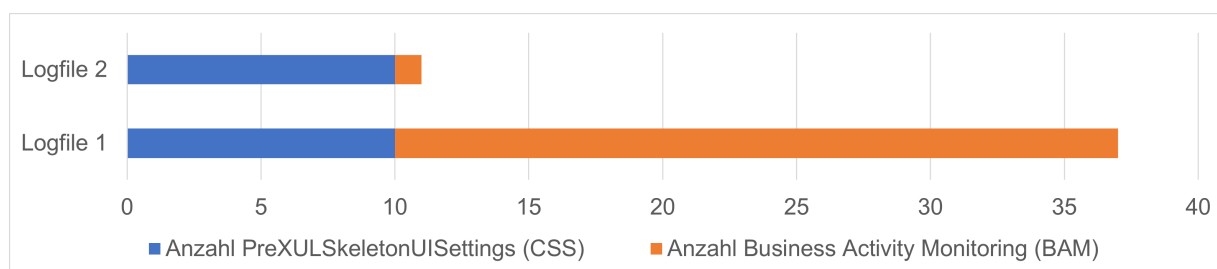


Abbildung 5.20: Comparison of found PB artifacts between RAM Dumps

PreXULSkeletonUISettings Der "PreXULSkeletonUISettings"Registry Key enthält Einstellungen für die Benutzeroberfläche (UI) des Firefox-Browsers, insbesondere für das sogenannte SSkeleton UI, eine vereinfachte Benutzeroberfläche, die während des Ladens des Browsers angezeigt wird, bevor die vollständige Benutzeroberfläche geladen ist. PreXULSkeletonUISettings Registry Keys haben das Format HKCU\SOFTWARE\Mozilla\Firefox\PreXULSkeletonUISettings\<Absoluter Firefox Installationspfad>\firefox.exe|<Skeleton UI Setting>. Somit enthält der Key den absoluten Installationspfad von Firefox gefolgt von einer Skeleton UI Einstellung. Nachfolgend sind alle möglichen UI Einstellungen aufgelistet, gefolgt vom Datentyp des Keys.

- ScreenX (DWORD)
- ScreenY (DWORD)
- Width (DWORD)
- Height (DWORD)
- Maximized (DWORD)
- Flags (DWORD)
- CssToDevPixelScaling (REG_BINARY)
- UrlbarCSSSpan (REG_BINARY)
- SearchbarCSSSpan (REG_BINARY)
- SpringsCSSSpan (REG_BINARY)

Somit enthalten die Keys nur Daten zur Formatierung und Struktur der grafischen Oberfläche. Es wurden keine PB Artefakte geschrieben

Business Activity Monitoring "Business Activity Monitoring", kurz BAM ist eine weitgehend undokumentierte Windows Funktion, die im Hintergrund ausgeführte Programme steuert. Der Registry Key hat das Format HKLM\System\CurrentControlSet\Services\bam\State\UserSettings\<SID>\Device\Harddisk\<Firefox Installationspfad>\firefox.exe und den Datentyp REG_BINARY. Jeder Schlüssel wird durch die Sicherheits-ID (SID) des Benutzers identifiziert. Ein BAM Registry Key schreibt für alle ausgeführten Programme — hier Firefox — den Zeitstempel der letzten Ausführung. PB Artefakte sind dabei nicht enthalten.

Stringsuche in Registry Hives

Gemäß Methodik in Kapitel X wird die Firefox Registry als Uncommon Location behandelt, indem über alle auf der Festplatte vorhandenen Registry Datenbanken, den Registry-Hives, eine Stringsuche durchgeführt wird, ohne die Struktur der Hives zu beachten. Dazu wurden sowohl die System-Hives als auch die User-Hives aus Tabelle X (TODO!) aus jedem Snapshot extrahiert und mithilfe des Registry Explorers nach PB Artefakten durchsucht. Dabei wurde in keinem Snapshot in keinem Hive ein PB Artefakt gefunden.

*** TODO: Zusammenfassung Firefox ***

5.2 Tor

White-Box Analyse/Common Locations

Schreiboperationen mit Process Monitor verfolgen:

Im Anhang: Tabelle mit allen geschriebenen Dateien (markiert, wenn nicht mehr wiederherstellbar + markiert, wenn Datei "verändert"(siehe oben: temp, WAL))

Aux-Dateien, welche nicht mehr vorhanden waren, aber dafür "richtige"Dateien:

Ergebnis: Tabelle mit wiederherstellbaren Dateien: Logfile 1 vs. Logfile 2 + Tool mit dem Datei untersucht wurde - Dateien, die in beiden Logfiles nicht wiederherstellbar

Kategorie	Datei	Logfile 1	Logfile 2	Logfile 3
Cache	I:\Cache\profile.default\startupCache\startupCache.8.little	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
Datareporting	I:\datareporting\gleant\db\data.safe.bin	Keine PB Artefakte	Keine Schreiboperationen	Keine PB Artefakte
	I:\datareporting\state.json	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
SQLite	I:\storage\permanent\chrome\idb\3870112724\segment-es.sqlite	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I:\storage\permanent\chrome\idb\165714555\Amcater\vti\Sky.sqlite	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\places.sqlite	Keine PB Artefakte	Keine PB Artefakte	Keine PB Artefakte
	I\cookies.sqlite	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\storage.sqlite	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\avicons.sqlite	Keine PB Artefakte	Keine Schreiboperationen	Keine PB Artefakte
	I\webappsstore.sqlite	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\formhistory.sqlite	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\addonStartup.json.gz	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\AlternateServices.txt	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
Sonstige Dateien	I\broadcast-listeners.json	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\extensions.json	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\extensions\staged\73a6fe31-535d-460b-a320-fcd088432321.xpi	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\tonion-aliases.json	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\prefs-1.p	Keine PB Artefakte	Keine PB Artefakte	Keine PB Artefakte
	I\security_state\data.safe.bin	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\setting\data.safe.bin	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\SiteSecurity\ServiceState.txt	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\SiteSecurity\ServiceState-1.txt	Keine PB Artefakte	Keine Schreiboperationen	Keine Schreiboperationen
	I\profile.default\builstore.json	Keine PB Artefakte	Keine Schreiboperationen	Keine PB Artefakte
	I\profile.default\cert_override.txt	Keine Schreiboperationen	Keine PB Artefakte	Keine Schreiboperationen
	I\profile.default\enumerated_devices.txt	Keine Schreiboperationen	Keine PB Artefakte	Keine Schreiboperationen
	I\profile.default\sessionCheckpoints.json.tmp	Keine Schreiboperationen	Keine Schreiboperationen	Keine PB Artefakte
	I\storage\defaultmoz-extension+++3041a34e-918a-4fca-8ea0-53f366d7a1f1\userContextId=4234367235\metadata-v2	Keine Schreiboperationen	Keine Schreiboperationen	Keine PB Artefakte
	Caches			
	Profile Default			

Abbildung 5.21: Tabelle mit wiederherstellbaren Dateien: Logfile 1 vs. Logfile 2

Allgemein: Tor hat nur einen "Common Pfad - Dateien tauchen in zwei unterschiedlichen Ordnern auf:

- -

- Alle Schreiboperationen von Prozess "firefox.exe"durchgeführt, nicht "tor.exe"

=> Keine der Dateien enthält PB Artefakte, trotzdem nachfolgende genauere Betrachtung der wichtigsten Dateien im Zusammenhang des Tor Browsers

Kategorien der Logs: - Cache: > Zweck: "Die Datei \startupCache.8.little\ ist eine interne Datei, die von Firefox und dem Tor Browser erstellt wird, um den Startvorgang des Browsers zu beschleunigen. Sie enthält im Wesentlichen eine Zwischenspeicherung von Daten, die beim Starten des Browsers benötigt werden.

Diese Datei enthält Informationen über bereits geladene Browser-Komponenten wie JavaScript-Code, CSS-Dateien, Bilder und andere Ressourcen. Indem der Browser diese Informationen zwischenspeichert, kann er sie beim erneuten Starten des Browsers wiederverwenden, anstatt sie erneut herunterladen und verarbeiten zu müssen. Dadurch wird die Startzeit des Browsers verkürzt und die allgemeine Leistung verbessert.Änalyse: - Tool: HxD - kein PB Artefakte

- datareporting: > Zweck: "Die Datei `state.json` im Ordner `/datareporting` enthält Informationen über den Zustand und die Konfiguration des Firefox- oder Tor Browsers. Diese Datei kann Daten über die Verwendung des Browsers, wie z.B. installierte Add-Ons, zuletzt besuchte Websites, Browser-Einstellungen und andere Informationen enthalten. Sie wird verwendet, um dem Browser bei Bedarf den Zustand und die Einstellungen wiederherzustellen. Analyse: - Tool Notepad++ mit JSON Plugin - keine PB Artefakte
- Sonstige Dateien: > Zweck: enthält onion URLs, HTTP Alternative Services is a mechanism that allows servers to tell clients that the service they are accessing is available at another network location or over another protocol. This mapping can be stored in a file in the profile folder. This allows websites that do not support HTTPS to communicate in a secure way via port 443 (Opportunistic Encryption).» Zweck: Ist "NoScript" Extension. Wenn in Firefox geöffnet, kann installiert werden-> TODO: Screenshot, wenn in Firefox per "drag-and-drop" gezogen > Zweck: Enthält SecureDrop Adressen: z.B. `sueddeutsche.securedrop.tor.onion` (z.B. > Zweck: The file containing the updated security data > Entielt früher private Browsing Artefakte (<https://gitlab.torproject.org/tpo/applications/tor-browser/-/issues/18589>), jetzt aber keine private Browsing Artefakte => Keine der Dateien enthält PB Artefakte
- SQLite: Aus Process Monitor Logfiles erkennbar: Tor verwaltet und beschreibt die exakt gleichen SQLite Datenbanken wie Firefox.

Hier ebenfalls gesondert betrachtet: Fokus auf die Entwicklung von Dateinhalt in allen Snapshots (1, 2, 3-1, 3 und 4) betrachtet

Ergebnisse: > Nach Browser-Installation noch keine SQLite-Datei angelegt (Snapshot 1) > Während

File	Snapshot 1: Browser installation	Snapshot 2: After Browsing Scenario, Browser open		(Tor only) Snapshot 3-1: After Identity reset		Snapshot 3: After Browsing Scenario, Browser closed		Snapshot 4: VM Shutdown	
		Yor WAL	Rock WAL	Tor (Diff)	Rock WAL	Tor	Rock WAL	Yor WAL	Rock WAL
places.sqlite	N/A	Initialisiert, Zitiert Onion URLs für Tor Standardseiten, wie "The Tor Blog" oder "Tor Browser Manual" und Spenden-Seite (http://www.torproject.org/docs/faq-fake-favicons.html#faq-fake-favicons)	no diff	Indizes bei vorhandenen Seiten aktualisiert	no diff	Indizes bei vorhandenen Seiten aktualisiert	no diff	no diff	no diff
cookies.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer	leer	leer
storage.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer	leer	leer
favicons.sqlite	N/A	Initialisiert, Enthält ein Präfix für alle favicons, die zu Tor Standardseiten führen, wie "The Tor Blog" oder "Tor Browser Manual" und Spenden-Seite (http://www.torproject.org/docs/faq-fake-favicons.html#faq-fake-favicons)	no diff	no diff	no diff	In allen drei Tabellen Indizes aktualisiert	no diff	no diff	no diff
webappsstore.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer	leer	leer
formhistory.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer	leer	leer
1657114355Amcshuim5ty.sqlite	N/A	Leer initialisiert (Nur Spaltennamen)	leer	leer	leer	leer	leer	leer	leer
3870112724rcgmnoottot-cs.sqlite	N/A	Initialisiert, 1 Zeile: "origins: chrome"	no diff	no diff	no diff	gleich bleibend	no diff	no diff	no diff
		Leer							
		Unverändert							
		Neuer (nicht-leerer) Inhalt							

Abbildung 5.22: Comparison of found PB artifacts between RAM Dumps

Browsing Szenario alle DBs Initialisiert, außer "webappsstore.sqlite" (Snapshot 2) - Dabei wurden in places.sqlite automatisch .onion URLs geschrieben, die zu Tor Standardseiten führen, wie "The Tor Blog" oder "Tor Browser Manual" bzw. die Tor Spenden-Seite, obwohl keine dieser Seiten aufgerufen wurde TODO: Screenshot von URLs? - in Favicons.sqlite wurden die exakt gleichen Einträge geschrieben, mit dem Präfix "Fake-favicon-uri". Ein tatsächliches Icon wurde nicht in die DB geschrieben - remote settings Datenbank enthielt den gleichen Eintrag wie es bereits bei Firefox der Fall war. Keine PB Artefakte - Restliche Dateien ohne Inhalt, nur Spaltennamen - Nach WAL Checkpoints bleiben Dateien unverändert > Nach Zurücksetzen der Browser-Identität (Snapshot 3-1) - in places.sqlite: Indizes bei eingetragenen Seiten aktualisiert - restliche Dateien unverändert > Nach Schließen des Browsers (Snapshot 3) - in places.sqlite sowie favicons.sqlite: Indizes bei eingetragenen Seiten aktualisiert - restliche Dateien unverändert - nach WAL Checkpoints bleiben Dateien unverändert > Nach herunterfahren der VM (Snapshot 4) - Alle Dateien unverändert, auch nach WAL Checkpoint

- Zusammenfassung: in keiner Datei PB Artefakte

Quantitativ: (Diagramme) > Balkendiagramm: Für jede Logfilekategorie: Anzahl Schreiboperationen Logfile 1 vs Logfile 2

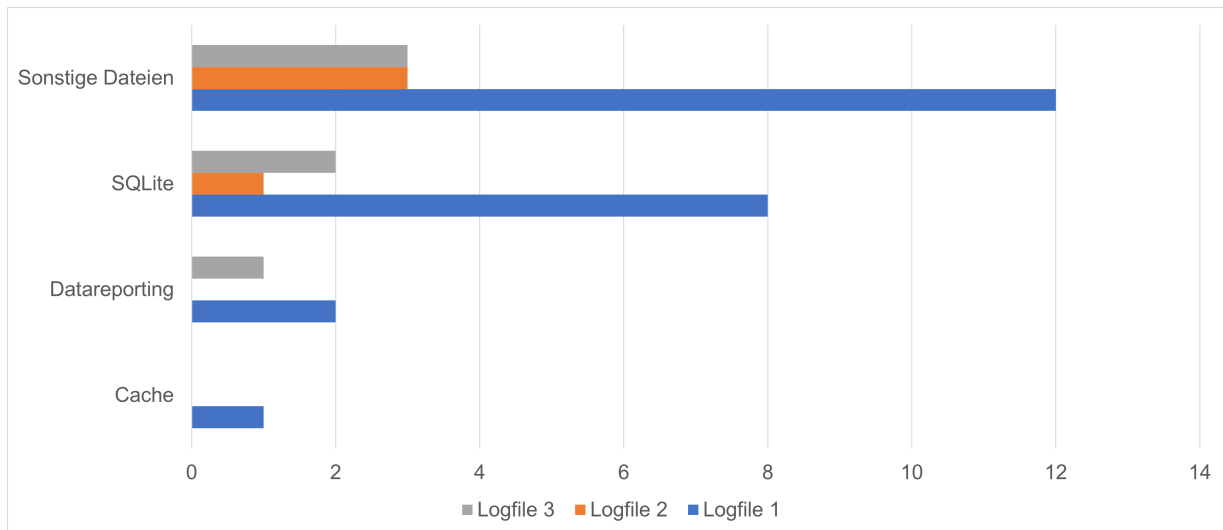


Abbildung 5.23: Comparison of found PB artifacts between RAM Dumps

Registry

> Process Monitor: SetValue Operationen von Browser Kategorien Registry Keys: Analog zu Firefox
 1) PreXULSkeletonUISettings: > Prefix: Absoluter Installationspfad von Firefox > Skeleton UI Einstellungen von Firefox Definition: > Der "PreXULSkeletonUISettings"Registry Key enthielt Einstellungen für die Benutzeroberfläche (UI) des Firefox-Browsers, insbesondere für das sogenannte "Skeleton UI". Das Skeleton UI ist eine vereinfachte Benutzeroberfläche, die während des Ladens des Browsers angezeigt wird, bevor die vollständige Benutzeroberfläche geladen ist. Es besteht aus grundlegenden Steuerelementen und Elementen, die dem Benutzer die Interaktion ermöglichen, während der Rest der Benutzeroberfläche noch geladen wird. > Der "PreXULSkeletonUISettingsSchlüssel enthielt Konfigurationsoptionen wie Farben, Positionen und andere Einstellungen für das Skeleton UI. Durch das Bearbeiten dieses Schlüssels konnten Benutzer die Darstellung des Skeleton UI anpassen. Es ist jedoch wichtig zu beachten, dass das Ändern der Registrierungseinträge ein fortgeschrittenes Verfahren ist und Fehler zu Problemen mit dem Browser führen kann.

> Struktur der Keys: > Unterschiedliche UI Einstellungen - - - - - > keine PB Artefakte unter UI Einstellungen
 2) Business Activity Monitoring > Quelle: > BAM is a mostly undocumented feature that controls the programs executed in the background. DAM is a feature for devices supporting the "Connected Standby" mode (i.e when a device is turned on, but its display will be turned off). As a result, the BAM registry keys will contain data on any devices, while DAM registry keys will only contain data on mobile devices. > The BAM registry key contains multiple subkeys under bam State

UserSettings, with one subkey per user, identified with the user SID. While the key is in the SYSTEM

registry hive, program executions can thus still be tied to a specific user using this SID. > Each user-specific key contains a list of executed programs, with their full path and timestamp of last execution. > If a file is deleted, the eventual associated entry in the BAM is deleted as well after the system reboot. Additionally, BAM entries older than 7 days are deleted upon system boot. The BAM thus provides limited information on historic execution of programs > No entries are created in the BAM keys for executables on removable media and/or on network shares. > Key:

Quantitativ: (Diagramme) - Stacked Balkendiagramm jeweils für Logfile 1 und Logfile2: Anteil Kategorie 1 bzw.2 an allen Registry-Schreiboperationen

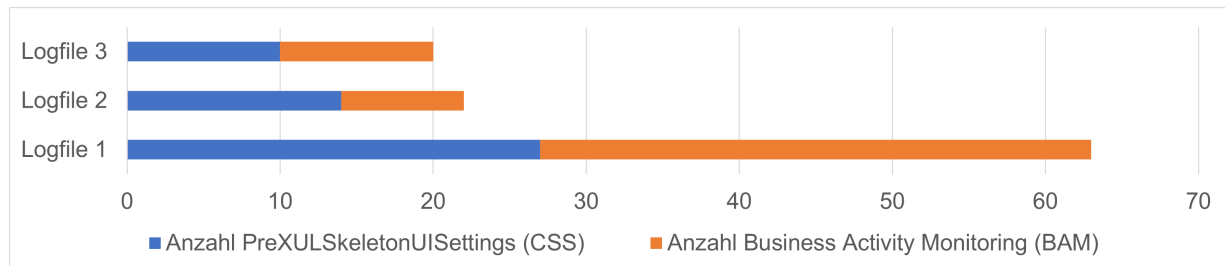


Abbildung 5.24: Comparison of found PB artifacts between RAM Dumps

> Stringsuche in Registry Hives mit Registry Explorer (Siehe Liste) In allen Hives kein Treffer für alle Suchbegriffe

Literatur: > Wie bei Firefox: Shellactivities Key existiert nicht mehr -> Nicht mehr vorhanden in aktueller Version (Verweis auf E-Mail)

Black-Box Analyse/Uncommon Locations

Analyse mit Autopsy

Bei White-Box Analyse/Common Locations: Autopsy nur zur Dateixtraktion genutzt, hier: als konkretes forensisches Werkzeug

Stichwortsuche: - In allen Snapshots keine Treffer (auch innerhalb \$Carved) - TODO: Pagefile gefunden?

Von Autopsy automatisch indexierte Dateien: In allen Fällen: keine Dateien gelöscht, nur über Zeitraum der Snapshots neue dazugekommen - Web Bookmarks: Snapshot 1: > Bing.url (Unter


Source Name	S	C	O	URL	Title	▼ Date Created	Program Name	Domain
 Bing.url			19	http://go.microsoft.com/fwlink/p/?LinkId=255142	Bing.url	2023-04-25 16:09:28 MESZ	Internet Explorer Analyzer	microsoft.com

Abbildung 5.25: Autopsy Web Bookmarks

C:/User/Forensik/Favorites/Links) enthält Bing Startseite Snapshot 2: > unverändert zu 1 Snapshot 3-1: > unverändert zu 2 Snapshot 3-2: > unverändert zu 3-1 Snapshot 4: > unverändert zu 3-2 - Web Cookies: Snapshot 1: > 9 Einträge in WebCacheV01.dat (= DB des Internet Explorers zum speichern

Source Name	S	C	O	URL	Date Accessed	Name	Value	Program Name	Domain
WebCacheV01.dat			15	bing.com	2023-05-07 13:56:11 MESZ	SUID	A	Microsoft Edge Analyzer	bing.com
WebCacheV01.dat			15	bing.com	2023-05-07 13:47:01 MESZ	ANON	A=D8530A977A1F5DAD20B78D8CFFFFFFF	Microsoft Edge Analyzer	bing.com
WebCacheV01.dat			15	login.live.com	2023-05-07 12:26:40 MESZ	Host-MSAAUTHP		Microsoft Edge Analyzer	live.com
WebCacheV01.dat			15	live.com	2023-05-07 12:25:56 MESZ	MUID	0EDC58E500A76FCE180F4BEF04A768D6	Microsoft Edge Analyzer	live.com
WebCacheV01.dat			15	www.bing.com	2023-05-07 12:25:44 MESZ	MUIDB	31708C5FC3CF47068AFADC1CB47D0111	Microsoft Edge Analyzer	bing.com
WebCacheV01.dat			15	bing.com	2023-05-07 12:25:44 MESZ	SRCHD	AF=NOFORM	Microsoft Edge Analyzer	bing.com
WebCacheV01.dat			15	bing.com	2023-05-07 12:25:44 MESZ	SRCHUID	V=28GUID=62F5FD78E9D9446BAFDF9DEC81881038dm...	Microsoft Edge Analyzer	bing.com
WebCacheV01.dat			15	bing.com	2023-05-07 12:25:44 MESZ	SRCHUSR	DOB=20230507	Microsoft Edge Analyzer	bing.com
WebCacheV01.dat			15	bing.com	2023-05-07 12:25:44 MESZ	SRCHHPGUSR	SRCHLANG=de	Microsoft Edge Analyzer	bing.com

Abbildung 5.26: Autopsy Web Cookies

von Browserdaten): Cookies für bing.com und live.com (= outlook) Snapshot 2: > unverändert zu 1 Snapshot 3-1: > unverändert zu 2 Snapshot 3-2: > unverändert zu 3-1 Snapshot 4: > unverändert zu 3-2 - Web History: Snapshot 1: > 2 Einträge in WebCacheV01.dat: - 2x live.com (= outlook) Snapshot

Source Name	S	C	O	URL	Date Accessed	Program Name	Domain	Username
WebCacheV01.dat				file:///Z:/Logfile2-2	2023-05-07 14:28:06 MESZ	Microsoft Edge Analyzer		Forensik
WebCacheV01.dat				file:///Z:/Logfile2-1	2023-05-07 14:17:05 MESZ	Microsoft Edge Analyzer		Forensik
WebCacheV01.dat				file:///Z:/Logfile1	2023-05-07 13:55:54 MESZ	Microsoft Edge Analyzer		Forensik
WebCacheV01.dat			15	https://login.live.com/oauth20_authorize.srf?client_id=000...	2023-05-07 12:26:43 MESZ	Microsoft Edge Analyzer	live.com	Forensik
WebCacheV01.dat			15	https://login.live.com/oauth20_desktop.srf?lc=1031	2023-05-07 12:26:40 MESZ	Microsoft Edge Analyzer	live.com	Forensik

Abbildung 5.27: Autopsy Web History

2: > 1 neuer Einträge in WebCacheV01.dat: - file:///Z:/Logfile_1 (= Process Monitor Logfile, die in shared-Folder geladen wurde) -> Erklärung? Snapshot 3-1: > 1 neuer Eintrag in WebCacheV01.dat: - file:///Z:/Logfile_2-1 (= Process Monitor Logfile, die in shared-Folder geladen wurde) -> Erklärung? Snapshot 3-2: > 1 neuer Eintrag in WebCacheV01.dat: - file:///Z:/Logfile_2-2 (= Process Monitor Logfile, die in shared-Folder geladen wurde) -> Erklärung? Snapshot 4: > unverändert zu 3-2 - Web Categories: Snapshot 1: > 2x WebCacheV01.dat aufgelistet => Mit HxD untersucht, keine

Source Name	△ S	C	O	Source Type	Score	Conclusion	Configuration	Justification	Domain	Host	Name
WebCacheV01.dat			0	File	Unknown				bing.com	bing.com	Search Engine
WebCacheV01.dat			0	File	Unknown				live.com	login.live.com	Web Email

Abbildung 5.28: Autopsy Web Categories

PB Artefakte Snapshot 2: > unverändert zu 2 Snapshot 3-1: > unverändert zu 3 Snapshot 3-2: > unverändert zu 3-1 Snapshot 4: > unverändert zu 3-2

Zusammenfassung: - keine PB Artefakte - Keine neuen Erkenntnisse vgl. mit intensiver Analyse mittels Process Monitor in Kapitel X - .onion URL Einträge in places.sql nicht erkannt

Analyse mit Volatility

Vorgehen: Siehe "Methodik" Kapitel - Ausgangslage: Volatility Yarascan Treffer - Für jeden Treffer: virtueller Offset des Strings, PID, getriggerte Yarasrule, getriggerte Yara Component z(= Variablenname des gesuchten Strings), gefundener String - Neue Spalte: "Prozessname" zu jeder PID Prozessnamen - Ergebnisse Aufbereitet nach folgendem Schema: > Für jeden RAM Dump > Für jede Yarasrule > Für

jede Component > Filter: Prozessname = Firefox -> Anzahl zählen > Filter: Prozessname = Alle Prozesse außer Firefox -> Anzahl zählen

Wie bei Firefox: HTML Artefakte wurden in keinem RAM Dump gefunden => Nicht aufgeführt

Yararule "Keyword": Analyse: > Ausschließlich in RAM Dump 2 und RAM Dump 3-1 Keyword Artefakte

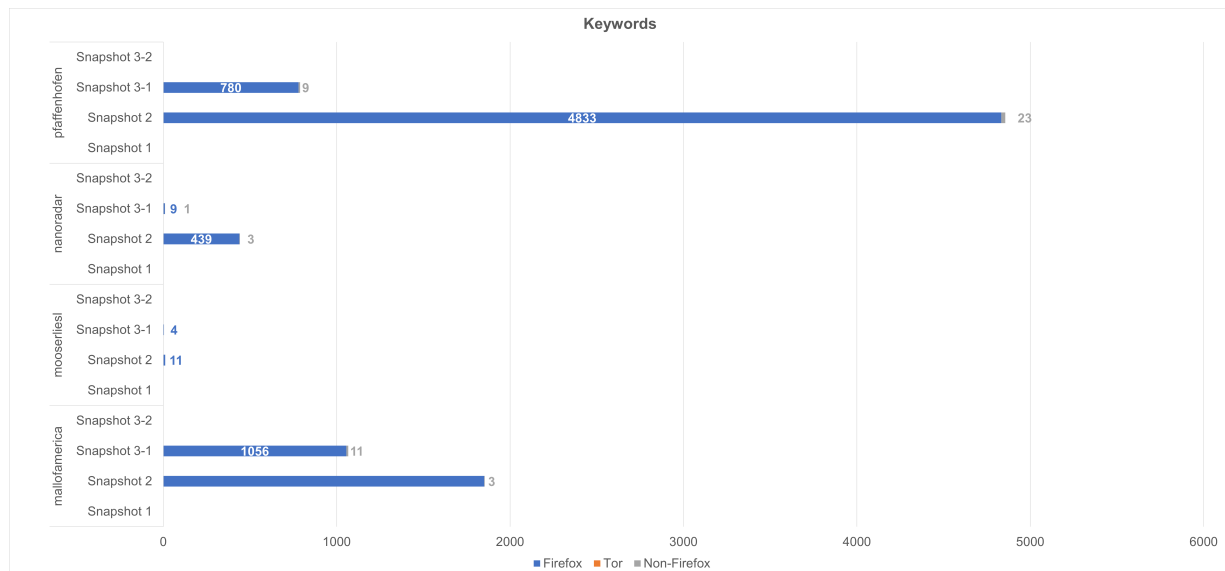


Abbildung 5.29: Keywords

gefunden > In RAM Dump 3-1 bei jedem Keyword deutlich weniger Artefakte als in RAM Dump 2 => Identitäts-Reset reduziert Keyword Artefakte deutlich > Hauptsächlich in Firefox Prozess, kein Artefakt in Tor.exe Prozess > Mit 4833 Artefakten in RAM Dump 2 am häufigsten "pfaffenhofen" vertreten. Vermutung: Evtl. weil Google Maps viele zusätzliche Artefakte lädt. > Nach Schließen von Tor Browser: keine Keyword Artefakte mehr in RAM

Yararule "URL": Analyse: > Wie bei Yararule "Keyword": Ausschließlich in RAM Dump 2 und RAM Dump 3-1 Keyword Artefakte gefunden > In RAM Dump 3-1 bei jedem Keyword deutlich weniger Artefakte als in RAM Dump 2 => Identitäts-Reset reduziert URL Artefakte deutlich > Hauptsächlich in Firefox Prozess, danach am häufigsten Tor.exe Prozess und am wenigsten Artefakte in anderen Prozessen > Bemerkenswert: "mallofamerica.com" ist mit 26.505 mal in RAM Dump 2 am häufigsten als Artefakt gefunden worden. Vergleich: "mooserliesl.de" wurde nur 508 mal in RAM Dump 2 gefunden > Nach Schließen von Tor Browser: keine URL Artefakte mehr in RAM

> TODO: DNSCache?

Yararule "Mail": Analyse: > Alle Mail Artefakte gefunden > Artefakte ausschließlich in Firefox Prozess gefunden > Artefakte fast ausschließlich in RAM Dump 2 Mail gefunden > Nur die Absenderadresse "computerforensikvl@gmail.com" wurde nach Identitäts-Reset in RAM Dump 3-1 gefunden > Absenderadresse ist häufigstes Mail Artefakt > Bemerkenswert: Passwort wurde 2x als Klartext im RAM gefunden! String Kontext: Offsets: PIDs: 0xb9c29180c8 7420 0x2859f4ffd4e0 7420 0x24083b41858 8424 0x240840e5b08 8424

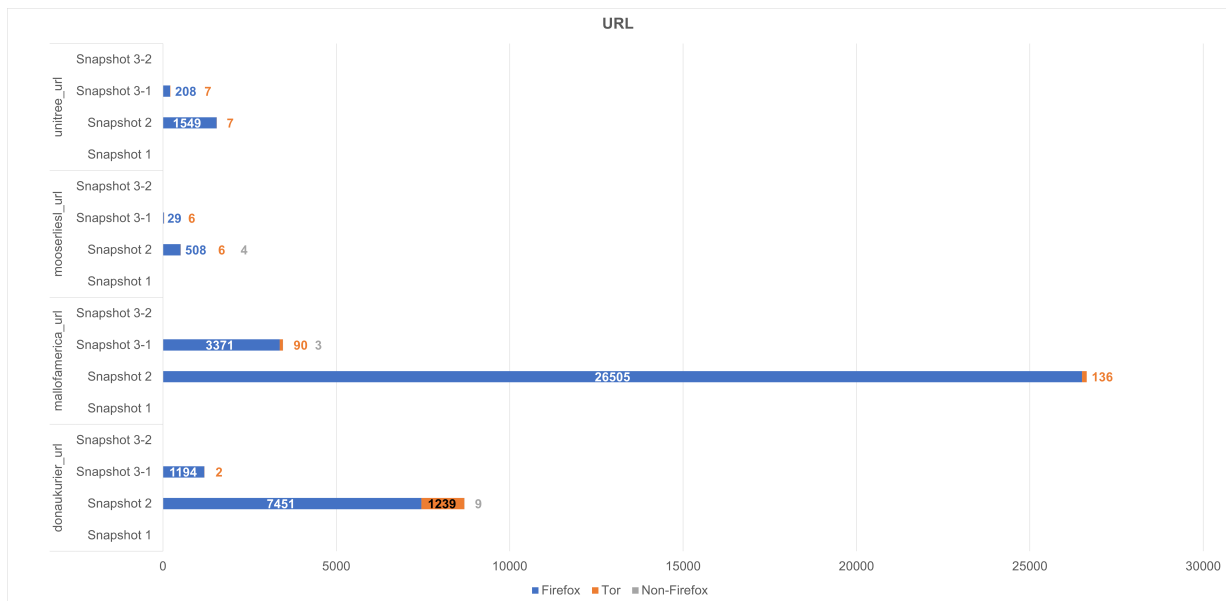


Abbildung 5.30: URL

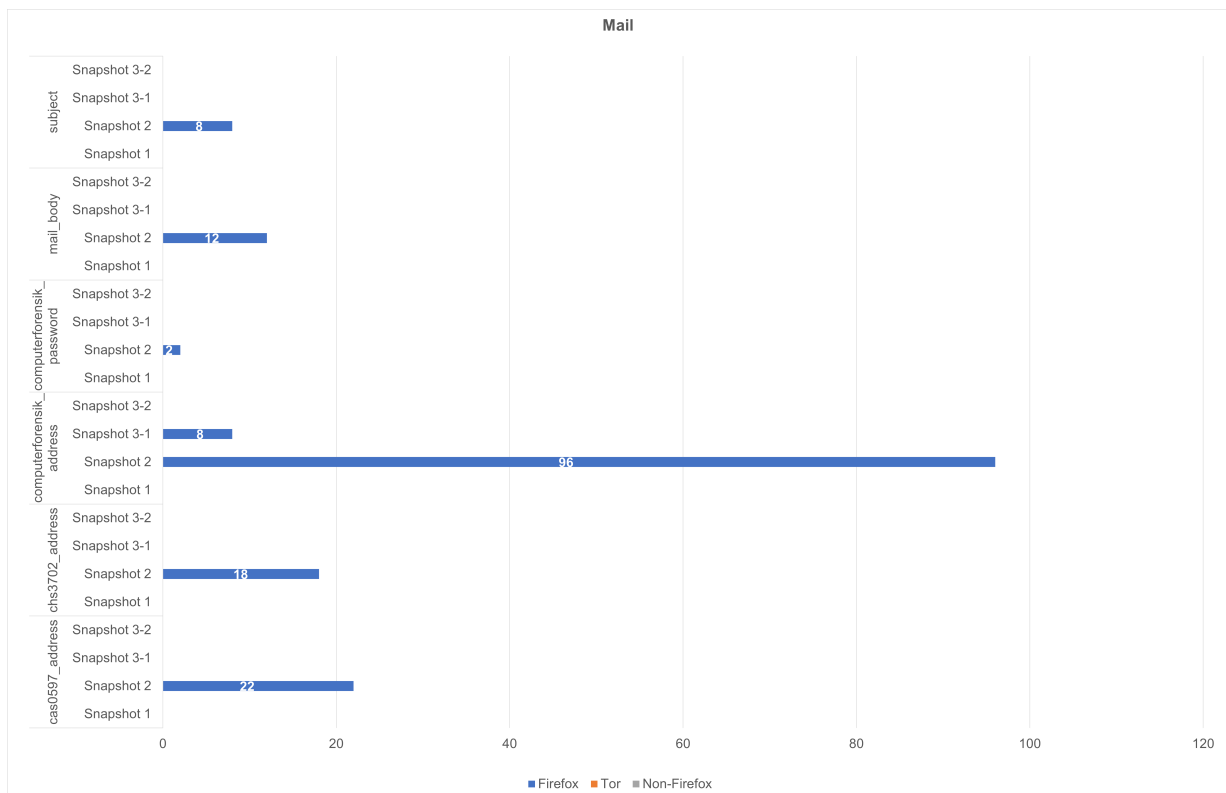


Abbildung 5.31: Mail

Yararule Image": Analyse: > Hex-Wert von Donaukurier Bild wurde ein einziges mal im 2. RAM

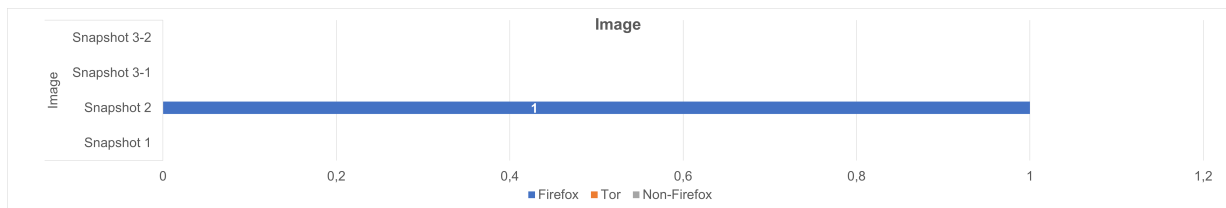


Abbildung 5.32: Image

Dump in einem Firefox Prozess gefunden

Zusammenfassung = Stacked Bar Chart: - PB Artefakte ausschließlich in RAM Dump 2 und 3-1

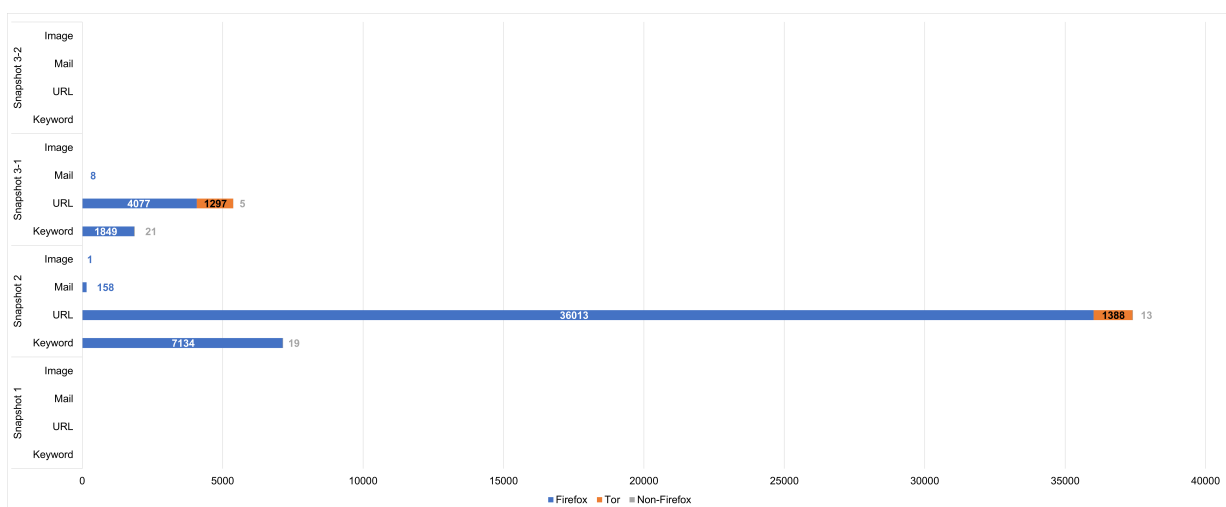


Abbildung 5.33: Summary

gefunden - Nach Identitäts-Reset deutlich weniger Artefakte in vorhanden - Am meisten URL-Artefakte gefunden, wobei mallofamerica.com dominant - HTML Artefakte wurden in keinem RAM Dump gefunden

TODO: Kreisdiagramme/Balkendiagramme mit Gesamtzahl an (Non-)Firefox Yarascan-Treffer erst im Vergleich mit Tor

Uncommon Locations

Literatur:

5.3 Chrome

Uncommon Locations

o Autopsy Keyword-Suche: > Chrome and Edge produced five artefacts as reported by both tools. (FTK, Autopsy) [6] -> Artefakte werden nicht genannt! > only two temporary files (Figure 7) were recovered with Minitool Power Data Recovery but it was a dead end; Location: appdata/.../Chrome/.../Preferences/RF1533fa.TMP [5] > pagefile.sys file showed no traces at all [22]

5.4 Brave

6 Vergleich der Browser

- Zusammenfassung: Vergleich Tor v. Firefox und Brave v. Chrome

Firefox vs Tor: > Gestacktes Balkendiagramm zu veränderten SQLite DBs => Erst bei Vergleich mit Tor!

TODO: Kreisdiagramme/Balkendiagramme mit Gesamtzahl an (Non-)Firefox Yarascan-Treffer erst im Vergleich mit Tor

- Firefox v. Chrome (SStandardbrowser") - Tor v. Brave (SSichere Browser") - Zum Schluss: Eine große Tabelle"mit den wichtigsten Kategorien?

7 Diskussion

> Artefakte im DNS Cache: [24] ■ DNS-Caching ist eine Bedrohung für private Browsing ■ Diese Schwachstelle entsteht, weil das Betriebssystem DNS-Anfragen des Browsers im Cache speichert, unabhängig davon, ob der Browser im privaten Modus ist oder nicht ■ Mehrere Jahre nach der Meldung dieser Schwachstelle besteht sie immer noch in allen Browsern fort ■ Es wurden einige Erweiterungen von Drittanbietern entwickelt, um dieses Problem zu beheben, aber keine davon wurde von den Browserherstellern übernommen.

> Viele RAM-Artefakte - Firefox [16] ■ Darcie et al. (2014) fanden Beweise für das Web-Browsing in Form von JPEG- und HTML-Dateien in Live-Forensik, aber eine statische Forensik war erfolglos. ■ Eine vorherige Live-Forensik-Analyse des Firefox-Browsers zeigte, dass Artefakte aus einer privaten Browsing-Sitzung aus dem Speicher wiederhergestellt werden konnten. (Findlay and Leimich, 2014).

> IE hinterlässt viele Spuren im Gegensatz zu Ergebnissen: [14] o hidden folders are usually stored at C:/Users/User/AppData o evidence searches are conducted extensively in the C: partition o bookmarks remain and can be viewed o downloads remain in the downloads folder until the user manually deletes them o CacheView trace entire URL and browsing histories including the temporary files CacheView enables to find the image's URL and from specific website

> Urteil über die Privatheit von Tor nach [16] The design aim of preventing Tor from writing to disk (Perry et al., 2018) is not achieved in this version. ■ Configuration files, downloaded files, and browserrelated data are recoverable from the file system. ■ Significant data-leakage from the browsing session occurred: HTTP header information, titles of web pages and an instance of a URL were found in registry files, system files, and unallocated space. ■ The data-leakage contained the German word for 'search' in reference to a Google search. This hints at the locale of the Tor server used to exit the network (exit relay). The Tor Project's design aim of enabling secure deletion of the browser (Sandvik, 2013) is not achieved in this version. ■ References to: the installation directory, Firefox SQLite files, bridging IPs/ports, default bookmarks, Tor-related DLLs and Tor product information were all recovered after the browser was deleted. ■ In a scenario where the operating system paged memory, an instance

Weiterführende Arbeiten: > Cross-mode interference [10]: o the Chrome://memory page displays all the opened tabs in the browser regardless if they are in the usual or private mode -> Nicht mehr aktuell -> Stattdessen: Chrome Task-manager (Ctrl + Esc), Funktioniert auch bei Firefox > Unser Scope: Process Monitor nach Prozessnamen gefiltert - Weiterführend: Nach Pathnamen filtern: "Common Locations"

> Für wen wird Browser entwickelt > Warum und für wen wird Private Browsing analysiert? > Ist das Auffinden privater Browsing Artefakte Schuld von Browser Entwicklern? (Oder Schuld des Betriebssystems, wie in (TODO!) erwähnt)

> bei Process Monitor nur nach Browser-Prozessen gefiltert

8 Fazit

Einleitend werden Struktur, Motivation und die abgeleiteten Forschungsfragen diskutiert.

Appendices

All File Operations Firefox

[illegible]

Abbildung .1: All File Operations Firefox: Logfile 1 vs. Logfile 2

All File Operations To

[illegible]

Abbildung .2: All File Operations Firefox: Logfile 1 vs. Logfile 2 vs. Logfile 3

Literatur

- [1] Gaurav Aggarwal u. a. "An Analysis of Private Browsing Modes in Modern Browsers." In: *USENIX security symposium*. 2010, S. 79–94.
- [2] Gabriele Bonetti u. a. "Black-box forensic and antifoensic characteristics of solid-state drives". In: *Journal of Computer Virology and Hacking Techniques* 10 (2014), S. 255–271.
- [3] Howard Chivers. "Private browsing: A window of forensic opportunity". In: *Digital Investigation* 11.1 (2014), S. 20–29.
- [4] Divya Dayalamurthy. "Forensic memory dump analysis and recovery of the artefacts of using tor bundle browser–the need". In: (2013).
- [5] Hasan Fayyad-Kazan u. a. "Forensic analysis of private browsing mechanisms: Tracing internet activities". In: (2021).
- [6] Ryan M Gabet, Kathryn C Seigfried-Spellar und Marcus K Rogers. "A comparative forensic analysis of privacy enhanced web browsers and private browsing modes of common web browsers". In: *International Journal of Electronic Security and Digital Forensics* 10.4 (2018), S. 356–371.
- [7] Ms Pooja Gupta. "Capturing Ephemeral Evidence Using Live Forensics". In: *IOSR J. Electron. Commun. Eng* (2013), S. 109–113.
- [8] Meenu Hariharan, Akash Thakar und Parvesh Sharma. "Forensic Analysis of Private Mode Browsing Artifacts in Portable Web Browsers Using Memory Forensics". In: *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*. IEEE. 2022, S. 1–5.
- [9] Nihad A Hassan. *Digital forensics basics: A practical guide using Windows OS*. Apress, 2019.
- [10] Ashley Hedberg. *The privacy of private browsing*. Techn. Ber. Technical Report, Tufts University, MA, USA, 2013.
- [11] Graeme Horsman u. a. "A forensic examination of web browser privacy-modes". In: *Forensic Science International: Reports* 1 (2019), S. 100036.
- [12] Aina Izzati und Nurul Hidayah Ab Rahman. "A Comparative Analysis of Residual Data Between Private Browsing and Normal Browsing Using Live Memory Acquisition". In: *Applied Information Technology And Computer Science* 3.2 (2022), S. 68–83.
- [13] Ahmed Redha Mahlous und Houssam Mahlous. "Private Browsing Forensic Analysis: A Case Study of Privacy Preservation in the Brave Browser". In: *International Journal of Intelligent Engineering Systems* 13.06 (2020), S. 294–306.
- [14] Raihana Md Saidi u. a. "Analysis of Private Browsing Activities". In: *Regional Conference on Science, Technology and Social Sciences (RCSTSS 2016) Theoretical and Applied Sciences*. Springer. 2018, S. 217–228.

-
- [15] Reza Montasari und Pekka Peltola. "Computer forensic analysis of private browsing modes". In: *Global Security, Safety and Sustainability: Tomorrow's Challenges of Cyber Security: 10th International Conference, ICGS3 2015, London, UK, September 15-17, 2015. Proceedings 10*. Springer. 2015, S. 96–109.
- [16] Matt Muir, Petra Leimich und William J Buchanan. "A forensic audit of the tor browser bundle". In: *Digital Investigation* 29 (2019), S. 118–128.
- [17] Apurva Nalawade, Smita Bharne und Vanita Mane. "Forensic analysis and evidence collection for web browser activity". In: *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. IEEE. 2016, S. 518–522.
- [18] Donny Jacob Ohana und Narasimha Shashidhar. "Do private and portable web browsers leave incriminating evidence? a forensic analysis of residual artifacts from private and portable web browsing sessions". In: *2013 IEEE Security and Privacy Workshops*. IEEE. 2013, S. 135–142.
- [19] Daniel Perdices u. a. "Web browsing privacy in the deep learning era: Beyond VPNs and encryption". In: *Computer Networks* 220 (2023), S. 109471.
- [20] Digvijaysinh Rathod. "Darknet forensics". In: *future* 11 (2017), S. 12.
- [21] Tri Rochmadi, Imam Riadi und Yudi Prayudi. "Live forensics for anti-forensics analysis on private portable web browser". In: *Int. J. Comput. Appl* 164.8 (2017), S. 31–37.
- [22] Huwida Said u. a. "Forensic analysis of private browsing artifacts". In: *2011 International Conference on Innovations in Information Technology*. IEEE. 2011, S. 197–202.
- [23] Priya P Sajan u. a. "Tor Browser Forensics". In: *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.11 (2021), S. 5599–5608.
- [24] Kiavash Satvat u. a. "On the privacy of private browsing—a forensic approach". In: *Data Privacy Management and Autonomous Spontaneous Security: 8th International Workshop, DPM 2013, and 6th International Workshop, SETOP 2013, Egham, UK, September 12-13, 2013, Revised Selected Papers*. Springer. 2014, S. 380–389.
- [25] Yunus Yusoff, Roslan Ismail und Zainuddin Hassan. "Common phases of computer forensics investigation models". In: *International Journal of Computer Science & Information Technology* 3.3 (2011), S. 17–31.