

Fall 11-2020

A Framework for Identifying Host-based Artifacts in Dark Web Investigations

Arica Kulm
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>



Part of the [Databases and Information Systems Commons](#), [Information Security Commons](#), and the [Systems Architecture Commons](#)

Recommended Citation

Kulm, Arica, "A Framework for Identifying Host-based Artifacts in Dark Web Investigations" (2020).
Masters Theses & Doctoral Dissertations. 357.
<https://scholar.dsu.edu/theses/357>

This Dissertation is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses & Doctoral Dissertations by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.



A FRAMEWORK FOR IDENTIFYING HOST-BASED ARTIFACTS IN DARK WEB INVESTIGATIONS

A dissertation submitted to Dakota State University in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Cyber Defense

November 2020

By

Arica Kulm

Dissertation Committee:

Dr. Ashley Podhradsky

Dr. Kevin Streff

Dr. Omar El-Gayar

Cynthia Hetherington

Trevor Jones



DISSERTATION APPROVAL FORM

This dissertation is approved as a credible and independent investigation by a candidate for the Doctor of Philosophy in Cyber Defense degree and is acceptable for meeting the dissertation requirements for this degree. Acceptance of this dissertation does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department or university.

Student Name: Arica Kulm

Dissertation Title: A Framework for Identifying Host-based Artifacts in Dark Web Investigations

Dissertation Chair: Ashley Podhradsky Date: 11/12/20

Committee member: [Signature] Date: 11/12/2020

Committee member: DocuSigned by: Dr. Kevin Struff
DFE9CC7B27D9456... Date: 11/12/2020

Committee member: DocuSigned by: Cynthia Hetherington
8687A076BC08447... Date: 11/12/2020

Committee member: DocuSigned by: Trevor Jones
AEF7A7B744EB46D... Date: 11/12/2020

ACKNOWLEDGMENT

First, I would like to thank Dr. Ashley Podhradsky, my dissertation chair. Ashley you were the first person I talked to when I first considered coming to DSU and you have been encouraging and motivating every step of the way. I've followed your guidance since the beginning, it has never let me down and has been such a wonderful experience. I can't wait to see what comes next!

I am thankful for my dissertation committee members Dr. Kevin Streff, Dr. Omar El-Gayar, Cynthia Hetherington, and Trevor Jones. Dr. Streff, you have been supportive and helpful since my very first class at DSU. Dr. El-Gayar, I appreciate your knowledge of design science methodology and you sharing your insight to make this dissertation stronger. Cynthia, your industry expertise in OSINT is exceptional and you are such an awesome person. Trevor, your law enforcement experience was invaluable, and your sense of humor always makes me smile.

I would also like to thank the members of the SD DCI ICAC task force for your assistance in evaluating my artifact. Brent Gromer, Toby Russell, Hollie Strand, Kendra Russell, and Jackson Brown, all of you have incredibly stressful and difficult jobs that you handle with supreme professionalism.

Finally, I would like to thank my family for their love and support. Chad, thank you for encouraging me, believing in me, and giving me space to work when I needed it. Alexis, I followed your lead into cyber and now you are following mine. I love that we are taking this journey together. Jacey, you are such an energetic young woman and always there for those that need a friend. You have a compassionate heart that shines through. Ty, you are a quiet young man with an incredibly clever and witty sense of humor. You are a creative soul with many talents to share. Jasmine, you are such a bright, thoughtful young woman. Your path will become clear and you will do great things.

ABSTRACT

The dark web is the hidden part of the internet that is not indexed by search engines and is only accessible with a specific browser like The Onion Router (Tor). Tor was originally developed as a means of secure communications and is still used worldwide for individuals seeking privacy or those wanting to circumvent restrictive regimes. The dark web has become synonymous with nefarious and illicit content which manifests itself in underground marketplaces containing illegal goods such as drugs, stolen credit cards, stolen user credentials, child pornography, and more (Kohen, 2017). Dark web marketplaces contribute both to illegal drug usage and child pornography. Given the fundamental goal of privacy and anonymity, there are limited techniques for finding forensic artifacts and evidence files when investigating misuse and criminal activity in the dark web.

Previous studies of digital forensics frameworks reveal a common theme of collection, examination, analysis, and reporting. The existence and frequency of proposed frameworks demonstrate the acceptance and utility of these frameworks in the field of digital forensics. Previous studies of dark web forensics have focused on network forensics rather than host-based forensics. macOS is the second most popular operating system after Windows (Net Marketshare, n.d.); however, previous research has focused on the Windows operating system with little attention given to macOS forensics.

This research uses design science methodology to develop a framework for identifying host-based artifacts during a digital forensic investigation involving suspected dark web use. Both the Windows operating system and macOS are included with the expected result being a reusable, comprehensive framework that is easy to follow and assists investigators in finding artifacts that are designed to be hidden or otherwise hard to find. The contribution of this framework will assist investigators in identifying evidence in cases where the user is suspected of accessing the dark web for criminal intent when little or no other evidence of a crime is present.

The artifact produced for this research, The Dark Web Artifact Framework, was evaluated using three different methods to ensure that it met the stated goals of being easy to follow, considering both Windows and macOS operating systems, considering multiple ways of accessing the dark web, and being adaptable to future platforms. The methods of evaluation

included experimental evaluation conducted using a simulation of the framework, comparison of a previously worked dark web case using the created framework, and the expert opinion of members of the South Dakota Internet Crimes Against Children taskforce (ICAC) and the Division of Criminal Investigation (DCI).

A digital component can be found in nearly every crime committed today. The Dark Web Artifact Framework is a reusable, paperless, comprehensive framework that provides investigators with a map to follow to locate the necessary artifacts to determine if the system being investigated has been used to access the dark web for the purpose of committing a crime. In the creation of this framework, a process itself was created that will contribute to future works. The yes/no, if/then structure of the framework is adaptable to fit with workflows in any area that would benefit from a recurring process.

DECLARATION

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Arica Kulm

TABLE OF CONTENTS

DISSERTATION APPROVAL FORM	II
ACKNOWLEDGMENT.....	III
ABSTRACT	IV
DECLARATION.....	VI
LIST OF TABLES	IX
LIST OF FIGURES	X
CHAPTER 1. INTRODUCTION	1
BACKGROUND OF THE PROBLEM	1
HISTORY OF DIGITAL FORENSICS.....	2
HISTORY OF TOR.....	4
THE DARK WEB AND DARK WEB MARKETS	7
STATEMENT OF THE PROBLEM	11
RESEARCH QUESTION	14
OBJECTIVES OF THE PROJECT.....	14
ORGANIZATION OF THIS PAPER.....	15
CHAPTER 2: LITERATURE REVIEW	16
INTRODUCTION	16
DIGITAL FORENSICS FRAMEWORKS	16
DARK WEB FORENSICS	22
CONCLUSION.....	25
CHAPTER 3. RESEARCH METHODOLOGY	27
INTRODUCTION	27
RESEARCH METHOD	27
PROBLEM INVESTIGATION	28
RESEARCH QUESTIONS	30
ARTIFACT DESIGN	31
FRAMEWORK CREATION.....	35
EXPECTED CONTRIBUTION.....	38
TREATMENT EVALUATION.....	38
TIMELINE	40

LIMITATIONS.....	40
SUMMARY	41
CHAPTER 4 – ARTIFACT DESIGN	42
ANONYMIZING METHODS OR TOOLS	43
LAUNCHING TOR FROM REMOVABLE MEDIA	51
ADDITIONAL OBFUSCATION STEPS.....	59
WINDOWS REGISTRY ARTIFACTS	62
DARK WEB SITE HISTORY	68
TAILS	70
HOST-BASED ARTIFACTS	79
CONCLUSION.....	79
CHAPTER 5 – TREATMENT EVALUATION	81
SIMULATION OF PROPOSED FRAMEWORK	82
PREVIOUS CASE COMPARISON.....	100
EXPERT OPINION OF SOUTH DAKOTA DCI ICAC TEAM MEMBERS	102
SUMMARY OF TREATMENT VALIDATION.....	104
CHAPTER 6 – CONCLUSION	107
OBJECTIVE ACCOMPLISHMENT	107
CONTRIBUTION	108
LIMITATIONS AND FUTURE RESEARCH	108
REFERENCES.....	110
APPENDIX A: DIGITAL FORENSIC ANALYSIS METHODOLOGY.....	119
APPENDIX B: DIGITAL FORENSIC INVESTIGATOR CREDENTIALS	124
APPENDIX C: DARK WEB ARTIFACT FRAMEWORK EVALUATION QUESTIONS.....	126

LIST OF TABLES

Table 1 - Research Steps	34
--------------------------------	----

LIST OF FIGURES

Figure 1 - Tor client obtains a list of nodes (“Tor Project,” n.d.)	5
Figure 2 - Tor client picks a random path (“Tor Project,” n.d.).....	6
Figure 3 - Another site visit - a second random path is selected (“Tor Project,” n.d.) ..	6
Figure 4 - The World Wide Web	8
Figure 5 - Image of The Farmer’s Market (Sulphuric12, n.d.)	9
Figure 6 - Silk Road Anonymous Market (Nimuehr, 2018)	9
Figure 7 - Approach to Evidence in Cyberspace (M. M. Pollitt, n.d.).....	17
Figure 8 - Abstract Digital Forensics Model (InfoSec, 2016)	18
Figure 9 - Carrier and Spafford Graphic Representation of the Major Categories of Phases in the Framework (Carrier & Spafford, 2004)	18
Figure 10 - NIST Basic Forensic Collection (Kent et al., 2006)	19
Figure 11 - EIC Framework Overview (Osborne et al., 2010)	21
Figure 12 - Engineering Cycle (Martakis, 2015)	28
Figure 13 - Implementation evaluation (Martakis, 2015)	29
Figure 14 - Start of the framework.....	35
Figure 15 - Framework showing options of Windows PC, macOS, and TAILS.....	36
Figure 16 - Further branching from Windows PC showing Tor installed locally vs. Tor run from a USB drive.....	36
Figure 17 - Further branching from macOS showing Tor installed locally vs Tor run from a USB drive	37
Figure 18 - Further branching of TAILS showing it being run on both Apple and PC hardware.....	37
Figure 19 - Treatment validation (Martakis, 2015).....	39
Figure 20 - Timeline	40
Figure 21 – Windows Tor Local Still Installed.....	45
Figure 22 – Launch Location Artifacts	46
Figure 23 – macOS Tor Installed Local Artifacts	48
Figure 24 – macOS Tor Browser User Files Artifacts.....	49
Figure 25 - Windows Tor from USB (SD Card).....	52

Figure 26 - Windows Tor from USB (SD card) Launch Location Artifacts	53
Figure 27 - Windows Tor from USB (SD card) User Information Artifacts	54
Figure 28 - Windows Tor from USB (SD card) USB Information	55
Figure 29 - Windows Tor from USB (SD card) - USB (SD card) Artifacts	56
Figure 30 - macOS Tor from USB (SD card)	57
Figure 31 - macOS Tor from USB (SD card) HD Image	58
Figure 32 - macOS Tor from USB (SD Card) HD TorBrowser-Data/User Info	59
Figure 33 - Windows Local Tor Deleted Launch Location	61
Figure 34 - macOS Local Install Tor Deleted Tor Browser Files-User	62
Figure 35 - Windows Tor from USB (SD Card) HD Registry	63
Figure 36 - Windows Tor from USB (SD Card) NTUSER.DAT	64
Figure 37 - Windows Tor from USB (SD Card) USB Registry SYSTEM	65
Figure 38 - Windows Tor from USB (SD Card) USB Registry Artifacts	66
Figure 39 - Windows Tor Local Registry	67
Figure 40 - Windows Tor Local Registry SOFTWARE	67
Figure 41 - Windows Tor Local Registry SYSTEM	68
Figure 42 - places.sqlite	68
Figure 43 - Windows .onion Locations	69
Figure 44 - macOS .onion Locations	69
Figure 45 - Memory Capture Artifacts	70
Figure 46 - Memory Images	70
Figure 47 - Windows PC Hard Drive Image Hash Verification Prior to Tails	71
Figure 48 - Windows PC Hard Drive Image Hash Verification After Tails	72
Figure 49 - macOS Hard Drive Image Hash Verification Prior to Tails	73
Figure 50 - macOS Hard Drive Image Hash Verification After Tails	74
Figure 51 - TailsData Unrecognized File System	75
Figure 52 - LUKS Encryption Indicated	75
Figure 53 - Tails Initial Branching	76
Figure 54 - Tails Encrypted Storage	76
Figure 55 - Tails Export Unrecognized Partition	77
Figure 56 - Tails Decrypt, Mount, Copy, Reimage	77

Figure 57 - Tails Process Decrypted Image with FTK	78
Figure 58 - Windows Hard Drive Verification Artifacts	83
Figure 59 - Start Tor Browser.LNK file	84
Figure 60 - TOR.EXE Prefetch File.....	84
Figure 61 - Tor Install Artifacts	85
Figure 62 - Tor Browser Install Executable.....	85
Figure 63 - .onion Artifacts.....	85
Figure 64 - places.sqlite Artifact.....	86
Figure 65 - Registry Artifacts	86
Figure 66 - AppBadgeUpdated Artifact.....	87
Figure 67 - UserAssist.....	87
Figure 68 - Shellbags	87
Figure 69 - Tor Browser in Shellbags.....	88
Figure 70 - Compatibility Assistant Artifact.....	88
Figure 71 - NTUSER.DAT Firefox Launcher Artifact.....	88
Figure 72 - SYSTEM Registry UserSettings Artifact.....	88
Figure 73 - ActivitiesCache.db Artifact	89
Figure 74 - ActivitiesCache.db Artifact for Tor Brower Install.....	89
Figure 75 - ActivitiesCache.db Artifact for firefox.exe.....	90
Figure 76 - Tor Browser Install Location	90
Figure 77 - Tor Executable Artifact.....	91
Figure 78 - macOS Validation Artifacts	92
Figure 79 - knowledgeC.db Artifact	93
Figure 80 - org.torproject.torbrowser.plist Artifact	93
Figure 81 - com.apple.ATS.plist Artifact	94
Figure 82 - Saved Application State	94
Figure 83 - CurrentPowerlog.PLSQL Artifact.....	95
Figure 84 - USB (SD Card) Artifacts.....	95
Figure 85 - FTK Image Verification Drive Serial Number	96
Figure 86 – USB Artifact Serial #.....	96
Figure 87 - Tails Persistent Storage Branching Determination	96

Figure 88 - Tails Persistent Storage	97
Figure 89 - Tails Password Branching	97
Figure 90 - Tails Decrypt, Mount, Copy, Reimage	98
Figure 91 - Tails Persistent Storage Branching	99
Figure 92 - Tails Persistent Storage Artifacts	99
Figure 93 - places.sqlite Artifacts	100
Figure 94 - Nm-system-connections Artifact.....	100
Figure 95 - SD Card Serial Number.....	101
Figure 96 - Windows Artifact Categories	105
Figure 97 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)	119
Figure 98 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)	120
Figure 99 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)	121
Figure 100 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)	122
Figures 101 & 102 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)	123

CHAPTER 1. INTRODUCTION

Background of the Problem

The dark web is the hidden part of the internet that is not indexed by search engines and is only accessible with a specific browser like The Onion Router (Tor). Tor was initially developed by the US government for private communications and is still widely used by individuals all over the world for that purpose. In addition to private communications, citizens around the world access content restricted in their country, or as a way around restricted internet sites in North Korea, Iran, or the Great Firewall of China, for example. In fact, the New York Times has had a constant dark web .onion page for years to ensure people all over the world can access their site regardless of internet censorship in their home countries. During the Arab Spring, activists utilized Tor to circumvent government restrictions and communicate on social media (Cattle, n.d.). However, the dark web has become synonymous with nefarious and illicit content and there are a significant number of .onion sites that sell weapons, drugs, stolen credit cards, stolen user credentials, child pornography, alleged murder-for-hire, hackers for hire, exotic animals, stolen antiquities, and more (Kohen, 2017). Given the fundamental goal of privacy and anonymity, there are limited techniques for finding forensic artifacts and evidence files when investigating misuse and criminal activity in the dark web. In addition to the built-in anonymity of the browsers, the landscape of the dark web is constantly shifting with the average lifespan of a dark web market being just eight months (Routley, 2018). Since the dark web is not indexed like surface web sites, search engines such as Google, Bing, Yahoo are not able to reach these sites. Finding traces of dark web activity indicating a connection to illegal material may be the difference between a suspect being charged with a crime or being allowed to go free to continue to commit these crimes. Therefore, providing evidence that a user has accessed dark web market sites to acquire illegal goods or services is valuable to investigators of these cases.

Therefore, the purpose of this design science research is to develop a reusable framework to assist digital forensic analysts in acquiring host-based artifacts when conducting computer forensic investigations involving suspected dark web criminal activity. Novel

approaches to investigate, acquire, and analyze dark web activity is being sought by agencies worldwide. This chapter is organized as follows: section one details the history of digital forensics and lists anti-forensic techniques used to thwart investigators including the use of private browsers like Tor. The creation and evolution of Tor is detailed along with the beginnings and history of the dark web. Understanding what makes Tor anonymous aids in the understanding of new and innovative approaches for combating that anonymity when used for criminal intent. The latest research on traces left behind by the use of Tor will be explored and will include what can be found after using Tor to access dark web sites in several scenarios. The beginnings of the dark web, dark web market sites, and the cycles of these market sites are explained. The problem being addressed by this research is stated with justification supported by first person interviews with a United States District Court Judge, a member of the South Dakota Division of Criminal Investigation (DCI) who is also a member of the Internet Crimes Against Children (ICAC) task force, and a Special Agent with Homeland Security Investigations (HSI). The end of this chapter outlines the objective of this research which is to develop a reusable framework to guide investigators when searching for host-based artifacts when their investigation is focused on crimes involving suspected dark web activity.

History of digital forensics

Although digital forensics is a relatively new field, the use of computers to commit a crime is not new. The first example of password hacking was in 1961 at the Massachusetts Institute of Technology (MIT) by Allan Scherr (Honan, 2012). Students were given passwords to access shared computers but were given four-hour timeslots. Frustrated by the short time, Scherr created a punch card to trick the computer into printing off all student passwords allowing him to login as other students when time ran out. In 1971, Roswell Steffen embezzled over a million and a half dollars from the Union Dime Savings Bank in Manhattan by manipulating computer records (Fosburgh, 1973). Steffen removed funds from accounts and covered his theft by making sure to transfer funds from other accounts into the depleted accounts when automatic interest payments were due. Accounts accumulated interest at different time periods, so Steffen was able to manipulate the accounts to ensure that each account contained the proper amount at the time of interest calculation thereby not alerting the account holders of their depleted funds. Steffen's shuffling of money between accounts was

caught while authorities were investigating an illegal gambling scheme he was part of (Liles et al., 2015).

The advent of the personal computer (PC) in the early 1980s brought computers out of labs and companies and into homes. Early on members of law enforcement recognized that these PCs could contain evidence of crimes (M. Pollitt & HisTory, 2010). Some of these individuals included those from the Internal Revenue Service (IRS), the United States Secret Service, and the Department of Defense who collectively founded the International Association of Computer Investigative Specialists (IACIS) in late 1989. IACIS is the first known organization dedicated to digital forensics and evolved out of the Computer Investigative Specialist training course offered by the Federal Law Enforcement Training Center (FLETC) in New Brunswick, Georgia. The course later came to be known as “Seized Computer and Evidence Recovery” or SCER and was the first course of its kind to offer training for the seizure, examination, and extraction of evidence from computers that had been involved in a crime and was a collaborative effort between federal, state, and local law enforcement (“IACIS - HisTory,” n.d.). Early investigations focused on stand-alone computers running a variety of operating systems and were performed on whatever equipment was available. Early tools used by investigators consisted of problem-specific command line tools or commercial data recovery tools. The first commercial software created specifically for digital forensics is believed to be SafeBack which was created in 1991 by Chuck Guzis. At this point in the evolution of digital forensics, the primary audience was the law enforcement community (M. Pollitt & HisTory, 2010). In the 1990s, officers tasked with performing digital forensic investigations had minimal or limited training and no official framework to implement to ensure repeatability in their investigation. As both hardware and software technology evolved to include not only stand-alone machines but also internet-connected devices, the use of email, and cellular phones capable of storing data, the need for specialized tools and training evolved. EnCase and Forensic ToolKit (FTK) are two early commercially developed tools that are now considered standards for use in digital investigations. Open-source tools such as Autopsy and The SleuthKit have followed and are two popular options used by investigators (“Popular Computer Forensics Top 21 Tools,” 2019).

The evolution of investigations has gone from looking for deleted files on a desktop machine that may hold 40 megabytes (MB) of data to searching numerous devices that have

multiple connections and within those devices data in various locations. On a single machine an investigator may be looking for email information, database information, log files, memory analysis, file analysis including deleted files, graphics and video analysis, internet browsing history, and much more on a machine that may hold 10 terabytes (TB) or more of data (“Amazing Facts and Figures About the Evolution of Hard Disk Drives - Pingdom Royal,” 2019).

The growth in digital forensics tools and expertise has brought the rise of anti-forensics techniques. In his paper titled *Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem*, Ryan Harris from The Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University describes anti-forensics as “any attempts to compromise the availability or usefulness of evidence to the forensics process” (Harris, 2006). Techniques for anti-forensics include data hiding, wiping, obfuscation, storage prevention, and physical destruction. Types of data hiding include encryption, steganography, and storing data in areas of the hard drive that are difficult to find such as slack space or alternate data streams. Wiping techniques include file or disk wiping utilities like Darik’s Boot and Nuke (DBAN) or BCWipe, and degaussing which uses a magnetic field to clean a storage device. Obfuscation techniques include log alteration or deletion, altering metadata, and altering file extensions such as making a .jpg file look like a .docx file. Prevention techniques such as private browsing prevent the saving of browsing data like cookies, temporary internet files, and internet history. Browsers like Edge, Safari, Chrome, Opera, Firefox, and Brave do not default to private or incognito browsing. A browser that does do this is The Onion Router (Tor). When Tor is installed, the default settings are for cookies to automatically clear on exit along with your browsing history (Perry, Mike; Clark, Erinn; Murdoch, Steven; Koppen, 2018). A user installing and using Tor with the recommended default settings is offered both prevention by not saving browsing information and encryption when data is transmitted.

History of Tor

Initial work on what later came to be known as Tor began at the United States Naval Research Lab (NRL) by David Goldschlag, Mike Reed, and Paul Syverson in 1995. In 1997 funding was contributed by the Defense Advanced Research Projects Agency (DARPA), a

United States Department of Defense agency whose stated mission is “to make pivotal investments in breakthrough technologies for national security” (“About DARPA,” n.d.). The original intent of the project was to protect online communications for United States military personnel stationed overseas by preventing the pinpointing of their location using their internet protocol (IP) address. Tor is still used today by law enforcement and intelligence organizations. The Central Intelligence Agency (CIA) announced in the summer of 2019 its own onion site for users to view the CIA website (Hay Newman, 2019). According to CIA director of public affairs Brittany Bramwell, “our global mission demands that individuals can access us securely from anywhere” (Central Intelligence Agency, 2019).

In addition to not saving browsing information and encryption, Tor uses a technique known as “onion routing” to send data from the host to destination on a path containing stops along the way. The origination of data is not known to the final destination, providing anonymity with encryption being added to provide privacy. Normally when a connection is made, the host computer makes a direct connection to the destination, so both know each IP address. Tor uses a series of connections, known as nodes, to connect the host to the destination. The first step in connecting to the Tor network is the client obtaining a list of Tor nodes from a directory server (Figure 1).

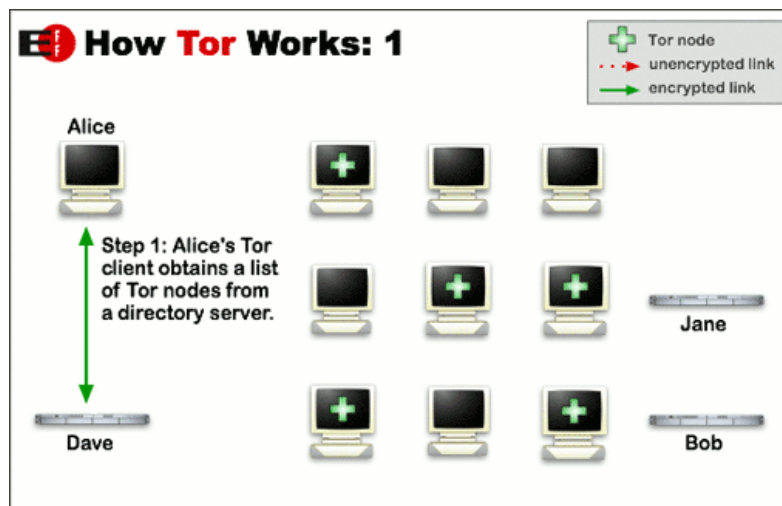


Figure 1 - Tor client obtains a list of nodes (“Tor Project,” n.d.)

During the second step, the client selects a random path to the destination server. For each node the data will pass through, a layer of encryption is added. Each node has a decryption key for only their layer and no other.

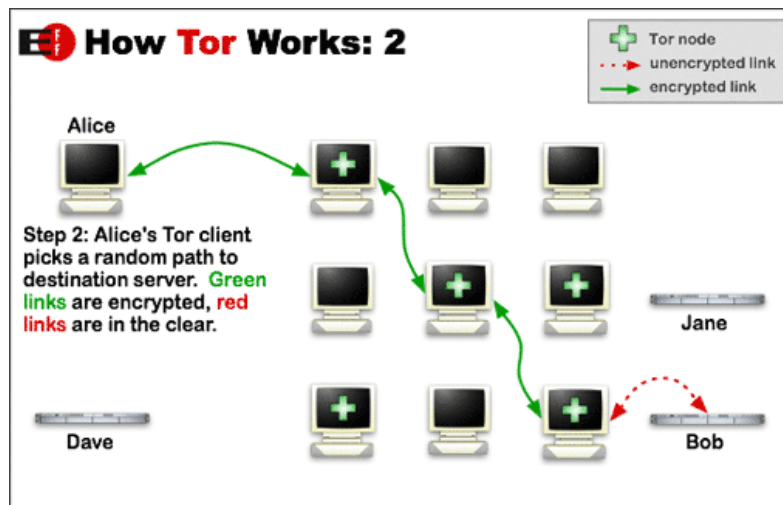


Figure 2 - Tor client picks a random path ("Tor Project," n.d.)

As each node is reached, it decrypts a layer then sends the data to the next node. Only when reaching the final destination is the data in an unencrypted form. If the user visits another site, another random path is taken to the new destination.

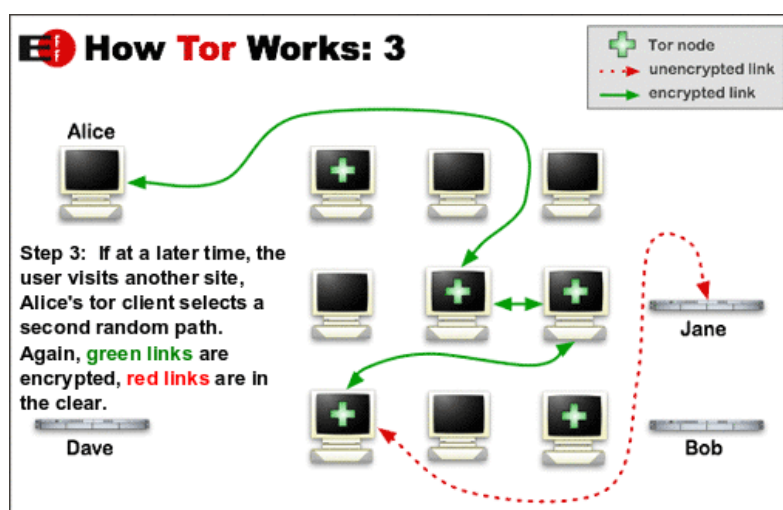


Figure 3 - Another site visit - a second random path is selected ("Tor Project," n.d.)

This layering preserves the privacy of the data preventing eavesdropping. Additionally, each node only knows the node it received the data from and the node it is sending data to, they do not know the full path from the host to the destination which prevents traffic analysis and creates anonymity (Syverson, 2005).

The project that started at NRL received the name The Onion Router or Tor when Roger Dingledine, a Massachusetts Institute of Technology (MIT) graduate joined Paul Syverson. Later both were joined by Nick Mathewson, also an MIT grad. Tor was released as open-source

software and initially had only a few dozen nodes. In 2004, the trio presented a paper titled *Tor: The Second-Generation Onion Router* stating that in May 2004 the Tor network consisted of 32 nodes (24 in the US and 8 in Europe) and several hundred users (Dingledine, Mathewson, & Syverson, 2004). Today, the Tor network consists of thousands of nodes and millions of users making it one of the largest anonymity networks (“Welcome to Tor Metrics,” n.d.). While Tor started as a means for secure communication, it has become a network for those who believe that all users should have access to a private, uncensored web (“Tor HisTory,” n.d.). Other networks such as Freenet and the Invisible Internet Project (I2P) exist that also provide privacy protections. Of these, Tor is the most popular network so is the focus of this research (Negi, 2017). The original purpose of the Tor network was to provide anonymity and privacy and while many valid reasons exist for seeking anonymity, that anonymity also provides a platform for those with criminal intent.

Tor can be installed on Windows, macOS, Linux, and Android systems. Tor can also be used as part of a stand-alone, bootable operating system such as The Amnesiac Incognito Live System (TAILS). TAILS can be booted using either a USB drive or DVD drive and states that it leaves no trace on the computer unless the user changes the default settings. According to the TAILS website, it does not use the host computer’s hard drive. TAILS uses the system RAM, but the website claims that RAM is automatically erased when the computer powers down (“Tails - Privacy for anyone anywhere,” n.d.).

The Dark Web and Dark Web Markets

The world wide web can be broken down into three layers. The part of the web that is available to the general public and accessible with search engines is known as the surface web or the Clearnet. This area of the web is indexed by search engines such as Google, Yahoo, or Bing and have content that can be accessed by anyone. Below the surface web lies the deep web which contains the majority of the world wide web. The deep web holds information that is accessible to a user with the correct credentials. Email accounts, bank accounts, medical records, student records, corporate databases, and cloud data are some examples of deep web information that can be accessed but require a link or credentials to ensure the user has the right to access that information. Within the deep web is a small section known as the dark web. Dark web sites require special software to access such as Tor or I2P and have sites that do not use

the domain name service (DNS) to resolve easy to remember addresses such as www.espn.com or www.dsu.edu but rather have links such as `wcgqzqyfi7a6iu62.onion` (Tor Project .onion Page) or `sntfgwfami5fdbn5.onion` (Sonic the Hedgehog Gamer Site).



Figure 4 - The World Wide Web

The beginnings of the dark web can be traced back to March of 2000 and the start of Freenet, a peer-to-peer platform for censorship-resistant communication and publishing (Moore & Rid, 2016). The privacy and anonymity of these networks led to the creation of marketplaces as a means to exchange illegal and unethical goods. However, an early obstacle of Freenet and Tor networks in the exchange of these illegal materials was payment. One of the first dark web marketplaces to use the Tor network was The Farmer's Market. The site offered various narcotics, including LSD, ecstasy, fentanyl, mescaline, ketamine, DMT, and high-end marijuana (Security, 2012). Transactions were conducted using Western Union, Pecunix, PayPal, I-Golder, and in cash making payment difficult and potentially traceable. The Farmer's Market was shut down in 2012 as part of Operation Adam Bomb and resulted in the arrest of seven U.S. citizens and one Dutch citizen on charges of federal drug trafficking and money laundering (Whitcomb, 2012).



Figure 5 - Image of The Farmer's Market (Sulphuric12, n.d.)

The advent of Bitcoin and other cryptocurrencies facilitated the growth in these underground markets. The first darknet marketplace to utilize cryptocurrency as a form of payment is considered to be the Silk Road, which initially consisted of illegal drugs of varying types. Created in 2011 by Ross Ulbricht, a.k.a. “Dread Pirate Roberts”, the Silk Road was ultimately shut down by the United States Federal Bureau of Investigation (FBI) in October 2013. Before it was shut down, the Silk Road sold drugs, guns, poisons, and contemplated the sale of bodily organs while other items such as stolen personal information, child pornography, and counterfeit currency were strictly off limits (Zajácz, 2017).



Figure 6 - Silk Road Anonymous Market (Nimuehr, 2018)

When one market shuts down another begins. Just 35 days after Silk Road was shut down Silk Road 2.0 began. When Silk Road 2.0 was shut down, Silk Road 3.0 launched 35 days later. At the time of shutdown of Silk Road there were five other active dark web drug marketplaces. Since then, over one hundred sites have gone online with the average length of operation being slightly over eight months (European Monitoring Centre for Drugs and Drug Addiction, 2017). As of the writing of this paper in 2020, there are over twenty current sites with the numbers shifting daily. These markets primarily offer illegal drugs with many more sites offering stolen credit cards, PayPal cards, hacking services, stolen credentials, illegal pornography, stolen antiquities, exotic animals, and worse (Paul, 2018).

At the same time as these drug sites were proliferating, sites containing graphic child pornography were also growing. In July of 2013, Eric Eoin Marques was arrested in Ireland for operating a dark web site containing over 1.4 million images of child pornography (United States Department of Justice, 2019a). Marques was extradited to the United States in March 2019 to face federal charges of advertising and distributing child pornography. The site Playpen was launched in 2014 with a user base of over 150,000 people from around the world before being shut down in February 2015 (Federal Bureau of Investigation, 2017). While the investigation, known as Operation Pacifier, is still ongoing, the FBI listed the arrest numbers as of May 2017 as follows:

- At least 350 U.S.-based individuals arrested
- 25 producers of child pornography prosecuted
- 51 hands-on abusers prosecuted
- 55 American children successfully identified or rescued
- 548 international arrests, with 296 sexually abused children identified or rescued (Federal Bureau of Investigation, 2017)

In 2016, a year after the Playpen site was shut down, a scan of Tor hidden services revealed that almost 30 sites related to child pornography remained active (Cox, 2016). Convictions related to the Playpen site continue with the most recent as of the writing of this paper being September 2019 (Anash, 2019). Law enforcement has been successful in taking down illegal market sites, however more come online in their place. Several operations since the takedown of Silk Road in 2013 have successfully targeted dark web marketplaces.

Operation Onymous in 2014, Operation Hyperion in 2016 (Cox, n.d.), Operation Bayonet in 2017, Operation Disarray in 2018 (Wegberg, Verburgh, Berg, & Staalduinen, 2017), Operation SaboTor in 2019 all resulted in seized websites and several arrests (“Feds Dismantled the Dark-Web Drug Trade—but It’s Already Rebuilding | WIRED,” 2019). The takedown of Wall Street Market as a result of Operation SaboTor provided law enforcement with intelligence to pursue the drug traffickers on these dark web markets. Operation DisrupTor was conducted in 2020 resulting in the arrest of 179 individuals with a seizure of \$6.5 million in cash and cryptocurrency, approximately 500 kilograms of drugs worldwide (The United States Department of Justice, 2020). In the United States alone there were 274 kilograms of drugs seized including fentanyl, oxycodone, hydrocodone, methamphetamine, heroin, cocaine, ecstasy, MDMA, along with 63 firearms.

The dark web is a key search area for open source investigations conducted by firms such as the Hetherington Group, a consulting, publishing, and training firm that is a leader in due diligence, corporate intelligence, and cyber intelligence. The founder of Hetherington Group, Cynthia Hetherington, MLS, MSM, CFE, CII, stated in a first-person interview that 20 to 30 percent of her investigations involve the dark web. According to Hetherington, many companies are still struggling to understand the dark web (Hetherington, 2019).

Statement of the Problem

Dark web marketplaces are having a regional impact in South Dakota, around the country and globally and contribute both to illegal drug usage and child pornography. Nearly every crime committed today contains a digital component. Criminals who are cautious may not leave obvious evidence on their digital devices identifying where or how they acquired illegal contraband. For example, their browser data may contain little information indicating criminal activity, they may not store photos of the illegal goods, they may not leave files on their system indicating criminal activity. The built-in anonymity of the Tor browser can give a false sense of security to criminals who have knowledge of how to access the dark web. Finding traces of dark web activity indicating a connection to illegal material may be the difference between a suspect being charged with a crime or being allowed to go free to continue to commit these crimes. McGregor W. Scott is a United States Attorney for the Eastern District of California and prosecutor of several dark web drug cases. Speaking about the recent case

against Marcos Paulo De Oliveira-Annibale of Sao Paulo, Brazil who faces federal drug distribution and money laundering charges for allegedly acting as a moderator on dark web marketplace Wall Street Market, “We are on the hunt for even the tiniest of breadcrumbs to identify criminals on the dark web. The prosecution of these defendants shows that even the smallest mistake will allow us to figure out a cybercriminal’s true identity.” (United States Department of Justice, 2019b). That tiny breadcrumb may be hidden in the mountains of data contained on a suspect’s machine. Digital forensics was a key component in the evidence collected and used to prosecute Ross Ulbricht, creator of the Silk Road dark web marketplace. Data collected from Ulbricht’s laptop included Bitcoin wallets with 140,000 Bitcoins and a list of Silk Road servers (Hayes, 2015). A reusable, easy to follow framework being designed by this research is needed to assist digital forensic analysts in identifying forensic artifacts on a Windows or macOS system for the purpose of aiding law enforcement when conducting investigations involving dark web criminal activity.

Regionally, South Dakota has seen convictions of users purchasing illegal drugs on the dark web. A snapshot of these convictions:

- 2018 – Four South Dakota men received federal prison charges for purchasing fentanyl and cyclopropyl fentanyl on the dark web (KSFY News, 2018).
- 2019 – A South Dakota man was convicted in federal court of purchasing MDMA, also known as ecstasy, and other controlled substances from sellers in Europe via the dark web for distribution in South Dakota (Epp, 2019).
- 2019 – A South Dakota man was convicted in federal court for purchasing meth online and reselling it to others (Klein, 2019).
- 2019 – A California man is facing conviction for shipping 2.6 million fentanyl pills nationwide which included Chamberlain, Mitchell, and Sioux Falls, SD (Kennecke, n.d.).
- 2020 – Seven South Dakota residents sentenced to prison for a dark web methamphetamine purchase and distribution (Lewis, 2020).

To further establish the impact the dark web is having regionally, first-person interviews were conducted with individuals with experience in the criminal justice system in South Dakota.

Interviews were conducted with Judge Roberto Lange, Toby Russell and Craig Sherer and are outlined below. These individuals were selected due to their work with dark web cases and legal expertise in South Dakota from both a federal and state perspective.

Judge Roberto (Bob) Lange is a United States District Court Judge from South Dakota presiding over the central region. Judge Lange has been a US District Court Judge since November of 2009. Prior to being named to the judgeship, Judge Lange had 20 years of private law practice with Davenport, Evans, Hurwitz and Smith, LLP of Sioux Falls, SD. Judge Lange stated that since he became a US District Court Judge in 2009 that he as probably had 20-30 cases total come before him that involved the dark web or approximately 2-3 per year. He stated that most do not get tried but rather end in a plea agreement. Judge Lange indicated that the cases he is seeing involving the dark web relate to child pornography. Judge Lange stated that the child pornography cases he has seen routinely involve the dark web and feels that child pornography cases are steadily growing and always involve the use of the internet. The increase in accessibility and ubiquitous nature of the internet contributes to this. Judge Lange indicated that the cases that are prosecuted now seem to involve more images than cases 10 years ago and that there are more cases overall as when he started he saw 1-2 cases per year and now sees 2-3 cases per year (Lange, 2019).

Toby Russell is a Special Assistant to the Attorney General (SAAG) within the South Dakota Division of Criminal Investigation (DCI) with 22 years of law enforcement experience who has been working internet crimes since 2002 and has been on the Internet Crimes Against Children (ICAC) task force since 2007. SAAG Russell is aware of numerous drug investigations in South Dakota that involve the purchase of drugs from the dark web using cryptocurrency and being shipped in from overseas and since 2017 there has been an influx with more and more individuals discussing getting drugs like heroin through dark web purchases. SAAG Russell's experience with ICAC cases is that the dark web is used as a communication platform for individuals seeking child pornography and provides what is believed by these individuals to be a secure mechanism for connecting with like-minded individuals. SAAG Russell has seen cases where little evidence remains except for a statement from a suspect that they are using the dark web which then leads to a forensic examiner finding these remnants. It is SAAG Russell's expert opinion that we will continue to see an increase in dark web activity in South Dakota, particularly in cases of illegal drugs. In addition, SAAG

Russell believes South Dakota will begin to see an increase in illegal weapons and stolen property being obtained from the dark web as it is not monitored by law enforcement (Russell, 2020).

Craig Sherer is a Special Agent with HSI, a position he has held since December 2004. Special Agent Sherer has been involved in local dark web investigations including one which resulted in the seizure of approximately 1 kilogram of Methylenedioxyamphetamine (MDMA), commonly known as Ecstasy, along with approximately \$36,000 in Bitcoin contained on a seized phone. Special Agent Sherer believes the individuals using the dark web to purchase illegal drugs to have enough sophistication to enable them to purchase drugs on dark web markets but these users feel secure in the ordering process to not take further precautions to mask their activities. It is Special Agent Sherer's expert opinion that dark web crimes will continue to increase in South Dakota as currently investigators have seen an increase in the amount of opioid cases with the local street value being higher than the going rate on the dark web. The dark web being a world market drives the prices lower so it can be purchased from overseas and then resold locally at a profit (Scherer, 2020).

The expert opinions of these individuals indicate a growing presence of dark web criminal activity. In addition, previous research has focused on network artifacts or solely on Windows-based artifacts. Therefore, a need exists for identifying host-based artifacts of dark web activity in criminal investigations.

Research Question

Based on the background information provided in this research regarding the impact of dark net markets at not only an international or national but rather local level, the following questions will be examined in this research:

- What host-based artifacts can be identified on a system running either Windows PC or macOS when the user has been accessing the dark web using Tor or Tails?

Objectives of the Project

The primary objective of this project is to design a framework that can be used by digital forensic analysts when searching for host-based artifacts on a computer where the use of the

dark web is suspected in the commission of a crime. Using the open-source OSINT Framework as a template to facilitate its use (Nordine, n.d.), this project will create a framework that can be used to aid in the search for and identification of artifacts indicative of dark web use for criminal intent. The use of this framework by forensic analysts will provide a roadmap to assist in locating dark web artifacts on a system by a user who is a suspect in a criminal investigation. These hard to find artifacts aide law enforcement in building a case against a suspect who is the subject of an investigation of a crime involving dark web activity.

Organization of this Paper

This research is organized as follows. Chapter 2 presents a review of related and relevant literature. Chapter 3 addresses the design science research methodology used for this research and poses the questions that will be answered in the course of this research. Chapter 4 describes the design of the Dark Web Artifact Framework. Chapter 5 is a comprehensive treatment evaluation to confirm the validity of the designed framework. Chapter 6 concludes this dissertation and discusses limitations and potential future work.

CHAPTER 2: LITERATURE REVIEW

Introduction

This chapter reviews existing literature that addresses digital forensics frameworks, previous studies regarding identifying dark web artifacts, and discusses the gap that exists in both Windows and macOS forensics. A common theme emerges in the previous studies of digital forensics frameworks of collection, examination, analysis, and reporting. The existence and frequency of proposed frameworks demonstrate the acceptance and utility of these frameworks in the field of digital forensics. Studies of dark web forensics have previously focused on network forensics to identify the source of dark web sites or to deanonymize users. Additional research has been conducted using live memory analysis and host-based artifacts on a system running the Windows operating system. This chapter concludes with the inclusion of cryptocurrency as a possible artifact indicative of dark web criminal activity.

Digital Forensics Frameworks

Several frameworks or models to assist in digital investigations have been presented that outline the overall process for conducting a digital investigation. These frameworks provide a repeatable process that address the key aspects in digital forensics of reliable, repeatable, and verifiable results (Forensic Focus, 2012). Some of these frameworks have many phases such as the Enhanced Integrated Digital Investigation Process proposed by Baryamureeba and Tushabe in 2004 (Baryamureeba & Tushabe, 2004) which has 21 phases, while others have few phases such as the Investigation Framework by Kohn, Eloff, and Olivier in 2006 (Kohn, Michael; Eloff, JHP; Olivier, 2006) which has three phases. The frameworks outlined below highlight some of the more cited frameworks that have been presented. The common theme that emerges from these frameworks is the overall process of collection, examination, analysis, and reporting.

One of the first digital forensic models was presented by Special Agent Mark M. Pollitt of the Federal Bureau of Investigation at the 18th National Information Systems Security Conference in 1995 (M. M. Pollitt, 1995). Pollitt outlines a four-step process of acquisition, identification, evaluation, and admissibility of evidence drawing a parallel to paper evidence.

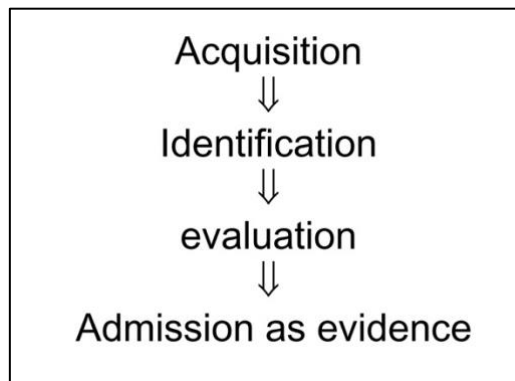


Figure 7 - Approach to Evidence in Cyberspace (M. M. Pollitt, n.d.)

Pollitt points out that the process of presenting paper evidence is clear and obvious because it is visible to the human eye whereas digital evidence requires the use of tools and must follow a process that is documented, reliable, repeatable, and presentable to members of the court for it to be useful in a criminal case. Three of the four phases of the process are shown to contain both legal and technical aspects. When acquiring evidence, legal authorization must be present before acquisition while searching for that evidence may require technical knowledge for manipulating the digital device and determining the location of the evidence. Identifying evidence requires technical knowledge of the device and being able to explain where the evidence physically and logically resides within the device as well as being able to view the evidence in context to ascertain the relevance and meaning. The example Pollitt gives of this is being able to look at data in HEX, ASCII, EBCDIC, or within an application. Finally, evaluation of the data requires both legal and technical judgements. Conclusions are reached when viewing evidence from a technical view of how data was produced, how it was accessed, who accessed it, and when it was accessed. Legally it must be determined if the data is relevant, if it is reliable, and if one can testify to it. Pollitt makes the conclusion that if information is not able to be admitted as evidence, then legally it does not exist. Both law and science must be used together to both identify evidence and explain its significance.

An abstract digital forensics model was proposed in the Fall 2002 edition of the *International Journal of Digital Evidence* by Mark Reith, Clint Carr, and Gregg Gunsch (Reith, Mark; Carr, Clint; Gunsch, 2009). The goal of this model was to provide a standardized method for digital investigations without making it tool-specific. Reith, Carr, and Gunsch's model draws on previous models and includes nine key steps of identification, preparation, approach strategy, preservation, collection, examination, analysis, presentation, and returning evidence.

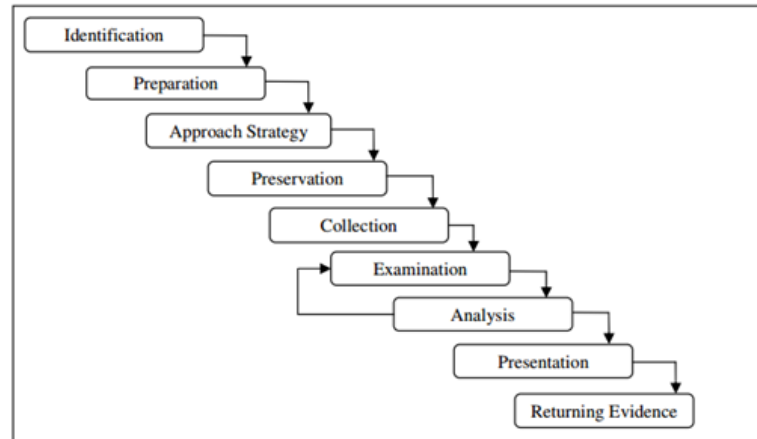


Figure 8 - Abstract Digital Forensics Model (InfoSec, 2016)

The model allows for expansion into sub procedures as needed, giving the example of including sub procedures under the examination category of volatile and non-volatile storage. The authors include both advantages and disadvantages of their proposed model. One point that is included as both an advantage and a disadvantage of their model is the generalized methodology. It is an advantage as it provides the ability to relate a technical process to the non-technical observer, however the authors acknowledge that the proposed model may be too general to be of practical use.

Based on the model proposed by Reith, Carr, and Gunsch a framework for collecting digital evidence was proposed in 2004 by researchers Brian Carrier and Eugene Spafford (Carrier & Spafford, 2004). Carrier and Spafford created an event-based framework based on procedures used when investigating a physical crime scene.

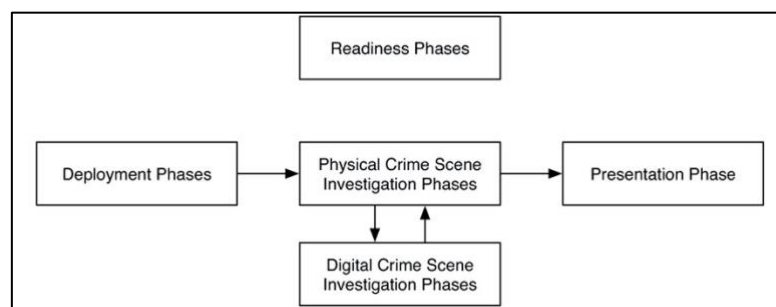


Figure 9 - Carrier and Spafford Graphic Representation of the Major Categories of Phases in the Framework (Carrier & Spafford, 2004)

This framework includes five phases focusing on the overall picture of gathering digital evidence. Included are phases for readiness, deployment, physical crime scene investigation, digital crime scene investigation, and finally presentation. The readiness phase ensures the appropriate training, tools, and equipment configuration are prepared for the investigation. The

deployment phase is when incidents are detected, confirmed, and the proper authorization for investigation is obtained. The physical crime scene investigation phase is the identification of devices at a crime scene that may contain digital evidence. The collection of these devices leads to the digital crime scene investigation phase. The digital crime scene investigation phase is further segmented to include system preservation and documentation, evidence searching and documentation, and event reconstruction and documentation. The final stage is presentation where the results are presented either to a corporate audience or to a court of law. This model proposed by Carrier and Spafford provides a high-level overview of a digital investigation, however lacks specific details of an investigation.

The National Institute of Standards and Technology (NIST) Guide to Integrating Forensic Techniques into Incident Response released in 2006 describes a basic forensic process of collection, examination, analysis, and reporting (Kent, Chevalier, Grance, & Dang, 2006).

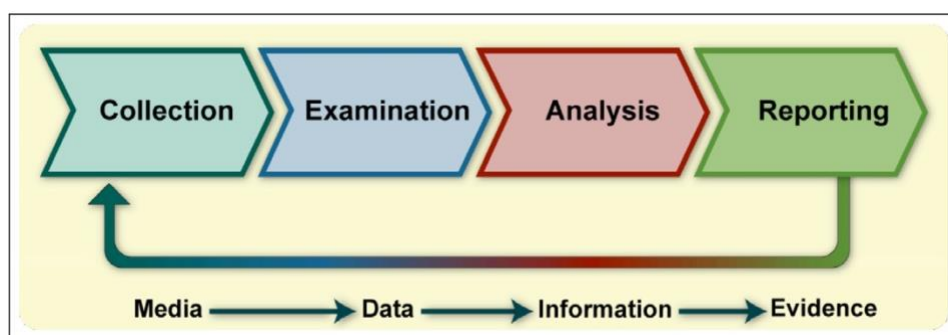


Figure 10 - NIST Basic Forensic Collection (Kent et al., 2006)

In this guide, the forensic process is described as a way to transform media into evidence. Collection of data includes identifying any possible media sources that may contain data both obvious and nonobvious. Obvious sources of data include items located in the physical area including desktops, laptops, thumb drives, CDs, cell phones, digital cameras, and volatile data contained on a system that is running. Nonobvious sources may be located in other locations and include logs from an internet service provider (ISP), information recorded by other organizations, or laptops owned by employees rather than the organization. Nonobvious sources may require court orders or have other legal constraints to consider. After data has been identified it needs to be acquired from the source in a three-step process. A plan is devised to acquire the data, the data is acquired, and the data is verified to ensure integrity. All steps need to be performed in a manner that supports any future legal action. The next phase of the

forensic process is examination which transforms the media into data. Relevant information is extracted from the media that has been collected using available tools and techniques, for example determining the type of file being examined or using text or pattern search techniques. The analysis phase takes the data extracted during examination and transforms it into information. Analysts study the data from all available sources looking for connections between people, places, and events. The final phase of reporting takes the information obtained from the analysis phase and presents it as evidence. A final step in the reporting phase includes identifying any policy or procedural deficiencies that exist that can be remedied before the next event.

A more extensive framework produced by the Department of Justice by Ovie L. Carroll, Stephen K. Brannon, and Thomas Song of the Cybercrime Lab Computer Crime and Intellectual Property Section (CCIPS) is the Digital Forensic Analysis Methodology (Oliver, Brannon, & Song, 2008). The methodology includes a flowchart detailing the preparation/extraction, identification, and analysis phases of their overall forensic process. The overall process includes the steps of obtaining and imaging forensic data, the forensic request, preparation/extraction, identification, analysis, forensic reporting, and case-level analysis. The flowchart details the middle three steps of the process and is shown in Appendix A.

The chart follows steps of a yes/no, if/then process for each stage with a Return on Investment strategy of determining when to stop the process. Five lists are included with the flow chart that are referenced at various stages during the process. Search leads, extracted data, relevant data, new data source leads, and analysis results are included in the framework with examples of how they are used and room for messages or notes. While progressing through the flowchart there are references to using each list. For example, during the preparation/extraction phase after selection of the appropriate forensic tool the requested data is extracted and added to the "Prepared/Extracted Data List". Following the addition to the "Prepared/Extracted Data List" the data search lead is marked as processed on the "Data Search Lead List". The preparation phase starts with determining whether there is sufficient information to start the process. The following steps of the preparation phase is the setup and validation of forensic hardware and software, choosing the appropriate forensic tools, extracting the data, and determining whether all relevant data has been processed. The identification phase starts where the preparation phase ends, with determining whether all relevant data has been processed. If

so, the identification phase continues leading to the analysis phase. If not, then a decision is made of what type of item has not been processed, whether it is relevant to the request that was made and whether it is within the scope of the warrant pertaining to the case. The analysis phase determines if there is sufficient data to be analyzed or if more data is needed. The analysis phase covers the who, what, where, when, and how of the data being analyzed. Other associated artifacts such as registry entries and metadata are considered which may indicate links to other items or events. Findings are documented on a timeline or other forms and the reporting process begins. The entire process can loop back to a previous phase at the identification, analysis, or case-level analysis phases to repeat if needed to provide more information.

The ‘Explore, Investigate and Correlate’ (EIC) framework was presented at the 2010 International Conference on Availability, Reliability and Security by Grant Osborne, Benjamin Tunstill, and Jill Slay of the University of South Australia (Osborne, Turnbull, & Slay, 2010).

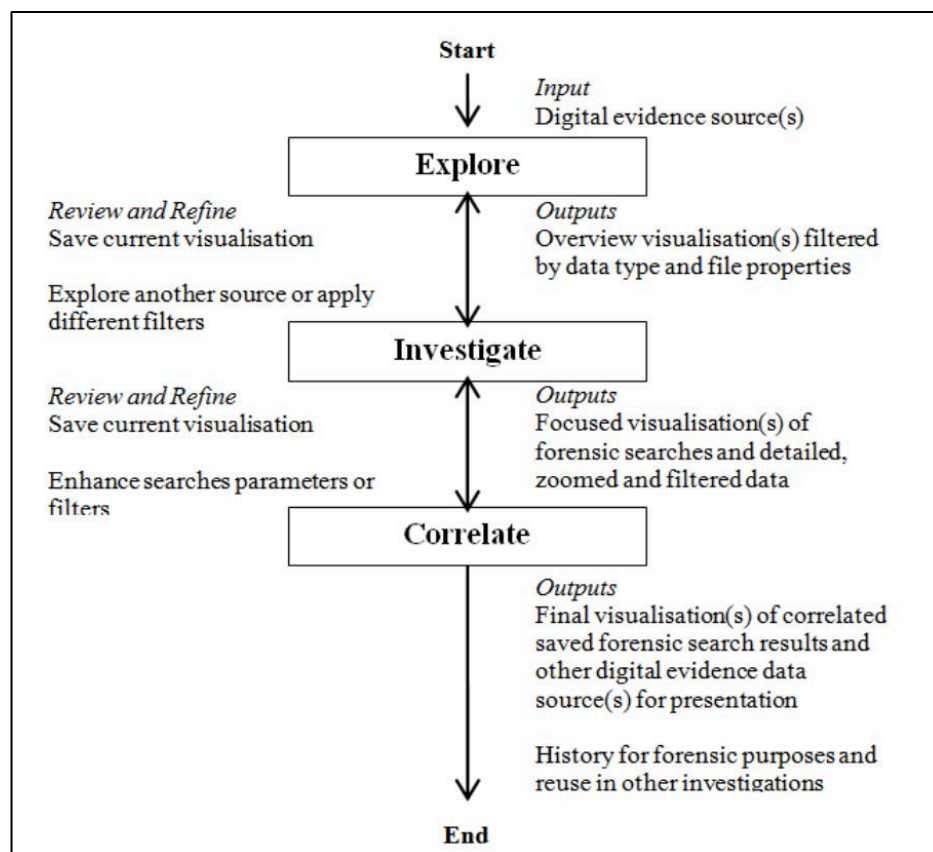


Figure 11 - EIC Framework Overview (Osborne et al., 2010)

The EIC framework is described by the authors as a “high-level conceptual framework to address issues surrounding scalability and comprehension of digital evidence”. The goal of

the EIC framework was to use visualization of information to streamline the processes and tasks of locating digital evidence in the growing number of devices and the complexity of devices in a manner that is scalable to the ever-increasing storage capacity of these various devices that investigators encounter. Visualization of information is a computer-aided, interactive, graphical representation of data (Ware, Mire, Kuniavsky, & Snyder, n.d.). It is a way for investigators to visually see and interpret relationships and concepts within vast amounts of data that may not otherwise be apparent. The EIC framework consists of three main phases of explore, investigate, and correlate. The explore phase is the identification of the types of data that are included and to provide an overview of the data as it relates to the crime being investigated. The investigate phase includes zooming in on data and filtering data to refine the data. Keyword searches are performed, event data is examined which includes file creation, modification, and deletion, and entities such as unique users or applications are identified. Visualization of these events aids in determining connections between events and entities. From this information a timeline of events can be established. The correlate phase combines the overview visualizations from the explore phase with the more focused visualizations of the investigate phase including evidence from all sources for correlation. Overall the EIC framework aims to be scalable to large amounts of data and to enable investigators to incrementally build a visual, presentable view of data that can help reduce the complexity of presenting findings from diverse sources.

The number of existing digital forensics frameworks demonstrates that they are an accepted practice in the field of digital forensics that enable analysts to conduct thorough investigations. Additionally, they provide the repeatable process that addresses the key aspects in digital forensics of reliable, repeatable, and verifiable results. Each of these frameworks outline the overall forensic process in a similar way with each presenting a slightly different approach. The research proposed in this study uses a comprehensive, reusable framework to enable analysts to find artifacts indicative of dark web criminal activity to aide in the prosecution of crimes committed using the dark web.

Dark Web Forensics

Previous studies regarding identifying dark web activity have centered around identifying users on the Tor network by utilizing techniques that de-anonymize these users at

the entry or exit nodes of the network. While they are difficult to exploit, vulnerabilities do exist in the Tor browser that can result in the de-anonymization of users (Jacoby MenTor & Chow, 2016). Two extremely unlikely scenarios are collecting data by controlling all of the nodes on a circuit used or controlling only the entry and exit nodes used by a particular user. There have been instances of malicious scripts being used to infect computers of visitors of the dark web that cause identification of those users. An example of this is the FBI's seizure of the dark web hosting service Freedom Hosting in which the FBI placed malware on the servers that then infected the users visiting the Freedom Hosting sites that held child pornography allowing the FBI to identify those users (Poulsen, 2013).

The Washington Post reported on a paper written in 2006 for the National Security Agency's (NSA) Cryptanalysis and Exploitation Services classified as Top Secret but shown on The Washington Post web page (The Washington Post, n.d.). The goal of the research was to review the Tor source code to search for vulnerabilities. The paper discusses using traffic analysis techniques, including tracing circuits, and was described as an unrealistic, but powerful attack using technology available at the time.

In addition to Tor, dark web sites can be accessed using other methods. The Amnesic Incognito Live System (Tails) is designed to be a live, bootable operating system that can be run from most computers by using either a USB drive or a DVD ("Tails - Privacy for anyone anywhere," n.d.). Little academic research has been done on Tails. Tails comes with Tor installed and also gives the option to use the Invisible Internet Project (I2P), an alternate anonymous network, rather than Tor (Cardenas-Haro & Dawson, 2016). Tails claims to provide privacy and to leave no trace on the system it is booted to, however errors by users can lead to the loss of privacy. Users who download files, bookmark sites, or install browser plugins run the risk of leaving traces on the device they are booting from whether it is a USB or DVD.

Anonymous browsing is also possible using other operating systems that have been developed with the intention of secure, anonymous communication. Whonix is an operating system designed to run within a virtual machine (VM) and was developed to use the Tor network. It is based on the Debian operating system with multiple layers of virtual machines and comes with pre-installed, pre-configured applications ("Whonix," n.d.).

In addition to the Tor network, other dark web networks exist that require a different method of access. A forensic analysis case study has been done on the Invisible Internet Project (I2P) network, a dark web network similar to, but different from Tor (Bazli, Wilson, & Hurst, 2017). I2P uses ‘eepsites’ rather than the .onion addresses that are used by Tor. The study discusses the difficulties in detecting I2P installation and use on a host machine. Commercial forensic software packages such as EnCase and FTK are not able to analyze or detect I2P artifacts or I2P activity. With modifications, detections are possible by using hash detection from the installer files for I2P.

Freenet is network that uses a peer-to-peer (P2P) architecture whose design intention was the “prevention of censorship” and “maintenance of privacy” (Miller, Hong, Wiley, Sandberg, & Clarke, 2002). Users of Freenet dedicate some of their own storage space by running a node on the network. Files stored on Freenet sites are stored on multiple computers so if one user node is not available a file can still be retrieved through a different node. Evidence of a Freenet installation on a system is available, however the files are encrypted so their contents are unknown.

Memory analysis and Registry file analysis was explored in a study published in 2015 by researchers at Marshall University working with members of the West Virginia State Police Digital Forensics Unit. Together they published a paper in which they used a combination of host-based forensics and network forensics to identify artifacts indicating dark web use (Darcie, Boggs, Sammons, & Fenger, 2015). Using RAM dump analysis, Registry file analysis, and packet capture the researchers were able to find useful information in each area they searched. A limitation they acknowledged was the use of RAM analysis on a live machine. This is not always an option for law enforcement as the state of the machine is not always controllable. Additionally, Registry file analysis showed that Tor had been used but no specific browsing activity. The existing research done by Marshall University and the West Virginia State Police provides a foundation for this proposal that would be expanded upon further.

An August, 2017 article for the International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Dr. Digvijaysinh Rathod proposes a framework for dark web forensics (Dr. Digvijaysinh Rathod, 2017). Dr. Rathod’s proposed framework is a broad overview of categories broken down into Tor forensics and Bitcoin forensics. Tor forensics includes random access memory (RAM) forensics, Windows Registry changes, network

forensics, and database information. Bitcoin forensics includes an analysis of Bitcoin wallets. Dr. Rathod's framework provides a good starting point for a framework and suggests tools to be used but lacks specific information for locating the proposed categorical artifacts.

Recent research performed on a Windows 8 virtual machine included analyzing Tor artifacts including memory, hard disk, and system registry with the Tor browser open and closed revealing several artifacts (Jadoon, Iqbal, Amjad, Afzal, & Bangash, 2019). The study largely involved normal user activity with little connection to dark web sites but rather connected to Clearnet sites through the use of the Tor browser. The research by Jadoon, et.al. would provide additional foundation for this proposal. At the time of this research, there wasn't a corresponding study for Windows 10. Research on artifacts on a Windows 10 operating system focused on Tor artifacts found when Tor is installed but did not include visits to dark web sites (Warren, 2017).

Dark web investigations often include detecting traces of cryptocurrency use as it is the payment method for transactions made on dark web marketplaces. Artifacts are left behind when users install the software required to have a bitcoin wallet (Wadas, 2018). Directories, subdirectories, and file names are able to be located on a system image. File directories are created in a user's personal folder under AppData\Roaming\Bitcoin with subfolders of "blocks", "chainstate", and "database". Also included in the Bitcoin folder are the files ".lock", "db.log", "debug.log", "peers.dat", and "wallet.dat". Researchers at the University of Luxembourg, Qatar Computing Research Institute, and Qatar University performed analysis on Bitcoin transactions using a combination of open source intelligence (OSINT) and blockchain transactions to link 125 unique users to dark web hidden services. Their conclusion was that users that use Bitcoin as a payment method on dark web marketplaces can be deanonymized (Al Jawaheri, Al Sabah, Boshmaf, & Erbad, 2020).

Conclusion

There are many existing frameworks for digital forensics which demonstrates their value to the field. These existing frameworks provide a foundation for the framework proposed for this research. Collection, examination, analysis, and reporting are common categories that are included in these frameworks and are important to digital investigations to ensure that findings are considered usable evidence. These frameworks help to provide reliable, repeatable,

and verifiable results. Research has previously been done to deanonymize Tor usage to detect users who are visiting dark web sites, however rather than viewing artifacts left behind on a host these have focused on network monitoring. In addition, users paying for dark web purchases with Bitcoin who also publicly share their wallet addresses on social media are able to be deanonymized which also does not focus on host-based artifacts as this research will. Previous research has also focused on artifacts left behind on Windows systems including live memory analysis and host-based analysis including Windows Registry changes as a result of dark web activity. The research proposed in this study will expand upon live memory forensics, Windows Registry modification, and include any newly identified artifacts to create a reusable, comprehensive framework for identifying artifacts on a host machine that has been used to access the dark web for criminal intent.

Further, this research will abstract away from a particular operating system thereby extending its applicability to other operating systems such as macOS. While Windows holds the majority of the market share of operating system use, macOS is the second leading operating system and appears to be growing (Net Marketshare, n.d.). This proposal will consolidate and combine existing knowledge with any newly discovered artifact locations including both Windows and macOS operating systems to create a comprehensive, reusable, multi-use framework that will provide investigators an avenue for discovering useful artifacts on a system used to access dark web sites for criminal intent.

CHAPTER 3. RESEARCH METHODOLOGY

Introduction

This chapter explains the research method chosen and the steps that will be taken to complete this research. Research methodology is the process followed by a researcher to solve the proposed problem (Rajasekar, Philominathan, & Chinnathambi, 2014). It provides a systematic approach of steps to walk through while studying the research problem (Kothari, 2004). The purpose of this study is to develop a usable framework for identifying host-based artifacts during a digital forensic investigation involving suspected dark web use. The dark web is shrouded in anonymity and is heavily encrypted. Both of these traits, along with the general mystery surrounding the dark web, lend itself to criminal activity. Accessing the dark web requires the use of a special browser such as The Onion Router (Tor). While other browsers exist, at this time Tor is the most popular so has been chosen for this research (“How to Access the Dark Net and Deep Web Safely - Step by Step Guide,” n.d.). The Tor homepage, www.Torproject.org, states the purpose of Tor as: Browse Privately, Explore Freely, and defend against tracking and surveillance (“Tor Project,” n.d.). The default setting for Tor is to clear cookies upon exit and delete any browsing history. The intentions of The Tor Project and default settings make finding host-based artifacts difficult, but not impossible.

Upon completion, the expected result of this research is a reusable, comprehensive framework that is easy to follow and assists investigators in finding artifacts that are designed to be hidden or otherwise hard to find. The contribution of this framework will assist investigators in identifying evidence in cases where the user is suspected of accessing the dark web for criminal intent when little or no other evidence of a crime is present.

Research Method

This research will implement design science to create an artifact to meet a specific goal. Design science focuses on creating something with a specific purpose in mind to add value or improvement (March & Smith, 1995). According to Alan Hevner, distinguished university professor and eminent scholar at the University of South Florida, the goal of design science is utility (Hevner, March, Park, & Ram, 1996). Design science projects employ the design cycle

which is a smaller portion of the overall engineering cycle (Wieringa, 2014). The design cycle includes problem investigation, treatment design, and treatment validation. An artifact is created to treat the identified problem in order to meet the goals of people who have a vested interest in the project, known as stakeholders. It is an iterative cycle where implementation evaluation determines the level of success of the treatment implementation and may lead to the beginning of a new design cycle if improvements are desired. Implementation evaluation and problem investigation both are the start of new cycles. While problem investigation is the start of a new cycle seeking improvement on the problem, implementation evaluation is a continuation of the cycle started during problem investigation with the goal of evaluating how effective the treatment is after it has been applied to the identified problem.

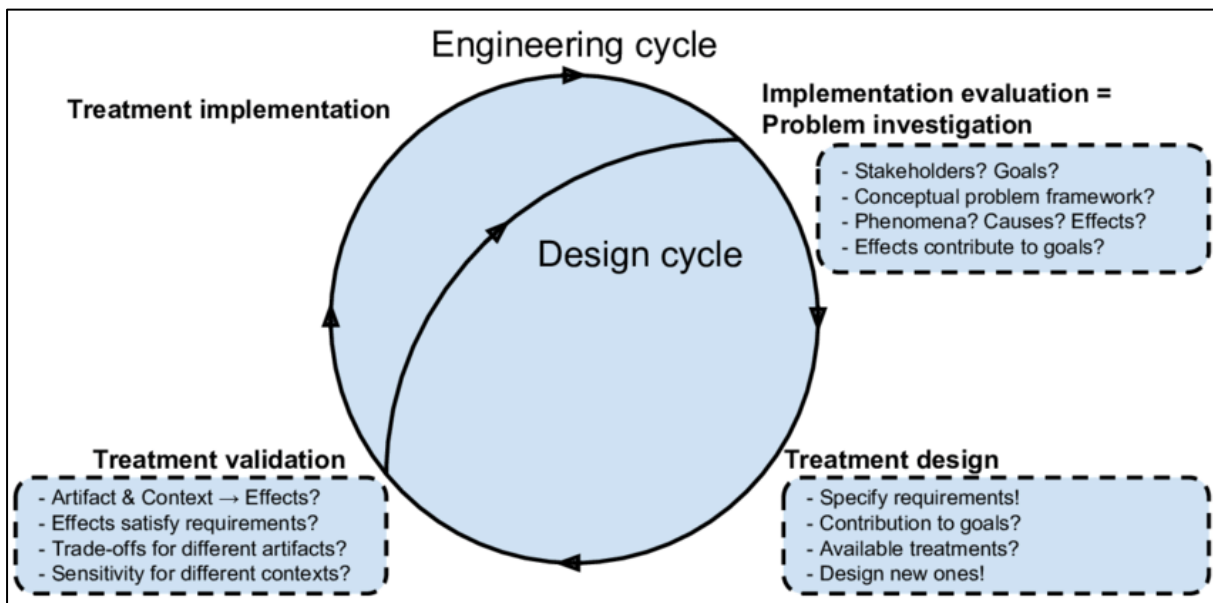


Figure 12 - Engineering Cycle (Martakis, 2015)

Problem Investigation

Problem investigation in design science focuses on improving upon a problem and the reasons that it is important. Real-world design problems are addressed through design and investigation of the problem and are driven by the goals of both the researcher and the stakeholders.

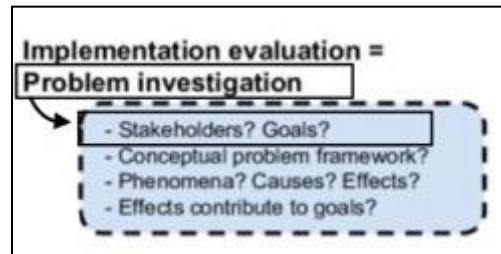


Figure 13 - Implementation evaluation (Martakis, 2015)

These stakeholders are those who are affected positively or negatively by the treatment of the problem. Stakeholders have varying degrees of awareness and interest in the problem being addressed and therefore have different desires. At the lowest level a stakeholder may be unaware that a problem exists, has input no thought or resources to the treatment of the problem, but would be affected by the treatment of the problem. At the highest level a stakeholder is fully aware that a problem exists and has invested thought and resources to addressing the problem. Stakeholder goals are defined as desires that the stakeholder has invested resources toward, either time or money, to achieve the desired outcome (Wieringa, 2014). Stakeholders who have not invested resources may still be affected by the treatment, but do not contribute to the goals of the research.

For this research, the problem is identified as finding host-based dark web artifacts that indicate criminal activity and can be used as evidence when the software used to access the dark web is designed to leave little trace behind. Crimes committed on the dark web are problematic for law enforcement. Criminals using the dark web rely on anonymity and encryption to stay hidden. The dark web, including dark web markets and cryptocurrency payments, are not widely understood by many law enforcement agencies. National law enforcement agencies such as the FBI work to identify users with the goal of shutting down the marketplaces and have achieved some success. Following multi-agency Operation Marco Polo that resulted in the takedown of the Silk Road marketplace in 2013 there have been several more multi-agency operations between the FBI and other U.S. and international agencies to shut down dark net marketplaces. These operations include:

- Operation Onymous in 2014 - that arrest of 17 individuals and take down several online marketplaces (Décary-Héту & Giommoni, 2017)
- Operation Hyperion in 2016 – targeted vendors and users (Cox, n.d.)

- Operation Bayonet in 2017 – took down two leading darknet marketplaces Alphabay and Hansa (Wegberg et al., 2017)
- Operation Disarray in 2018 – targeted vendors and users (FBI, 2018)
- Operation SaboTor in 2019 (U.S. Attorney’s Office, 2018)
- Operation DisrupTor in 2020 (The United States Department of Justice, 2020)

Despite these successes, several dark net markets continue to operate.

Locally however, law enforcement lacks the training and resources to find the source of these marketplaces but rather do have the ability to have an impact on the destination of dark web contraband. An impact can be made by assisting local investigations in finding evidence of dark web use. The primary stakeholders of this research are identified as the investigators trying to solve crimes committed using dark web sites such as buying illegal drugs, identity theft, weapons, terrorism, illegal pornography, and other criminal material. Additional stakeholders would be individuals who are victims of the crimes that are committed. These victims include:

- victims of violent crimes committed using weapons purchased on the dark web
- victims who have their identity stolen
- victims who have their bank accounts or credit cards compromised
- victims of human trafficking or otherwise used to produce child pornography
- victims of overdose caused by drugs purchased on the dark web
- healthcare organizations that are treating patients who overdose on drugs purchased on the dark web
- the families of all of these victims who are helping their loved one recover from any of these crimes

Research Questions

John W. Creswell is an academic who has written numerous journal articles and several books on research and research methods (“Bio — John W Creswell,” n.d.). Creswell states that research questions “narrow the purpose statement for research and become major signposts for readers” (Creswell, 2014). Design science uses knowledge questions that must be answered and are either analytical or empirical. Analytical questions are answered by analyzing collected

data while empirical questions are answered through observation and experience. The main empirical knowledge question that is the focus of this research is *“What host-based artifacts can be identified on a system running either Windows PC or macOS when the user has been accessing the dark web using Tor or Tails?”* To assist in answering this main question, follow-up questions can be asked such as:

- What methods or tools are used to remain anonymous when accessing the dark web?
- What additional steps are users taking to further obfuscate their dark web activity?
- What traces of Tor remain in the Windows Registry after use?
Is it possible to find what sites were visited while using Tor?
- If a system is booted with a TAILS USB drive that supposedly leaves no trace, will any traces of having booted to a USB remain? (Can BIOS memory be read to do this?)

The development of the proposed artifact, A Framework for Identifying Host-Based Artifacts in Dark Web Investigations, by gathering information from previous works conducted in the topic, through the investigations of existing tools, through information gained from interviews with investigators of dark web crimes, through the advice of experts in the field and the demonstration of the use of the artifact will answer these questions.

Artifact Design

Artifact design begins by specifying the requirements of the proposed artifact that meet the goal of the stakeholders. The requirements are design decisions determined and argued by the researcher as a method for contributing to the goals of the stakeholders. The absence of currently available solutions brings the need for new artifacts to address the identified problem. Previous research will be utilized that focused on Windows system memory and Windows registry analysis (Darcie et al., 2015). This will be combined with research conducted that located Tor remnants on a Windows 8 system (Jadoon et al., 2019) using Tor to visit Clearnet sites and research on a Windows 10 system that installed Tor but visited no dark web sites. These together as well macOS system artifacts, for which no previous research was located, will create one comprehensive framework.

The requirements for the proposed framework are as follows:

- Easy to follow – the framework must be easy for investigators to follow so as to not inhibit their investigation by requiring more effort than what is gained in useful evidence.
- Consider Windows and other operating systems – it is possible for investigators to receive multiple pieces of digital evidence when investigating a crime. Windows and macOS are the two most common desktop operating systems, therefore both will be explored for this research (Net Marketshare, n.d.).
- Consider multiple ways to access the dark web using Tor. Tor can be installed on a hard drive, run from a flash drive, or run as part of the bootable TAILS operating system. All three methods of access need to be explored.
- Be adaptable to future platforms.

Meeting these requirements would contribute to the goal of finding usable evidence of dark web criminal activity by considering the different avenues the dark web can be accessed. For the identified problem in this research, there is currently existing research that explores artifacts left behind by the Tor browser bundle including memory dump analysis seeking Tor artifacts (Dayalamurthy, 2013). Existing research also examines forensic artifacts produced by the use of Bitcoin or other cryptocurrencies (Doran, 2014) (Wadas, 2018). MacOS and Linux RAM have been analyzed using the Volatility framework to produce artifacts not found in non-volatile storage (Richard & Case, 2014). Digital forensic frameworks have been presented using traditional physical crime scene procedures as a model (Carrier & Spafford, 2004). Where the research appears to be lacking is combining all of these together to establish a unique framework for examining a static system, non-volatile as well as volatile memory looking for Tor artifacts, Bitcoin or other cryptocurrency signs, and web-based artifacts. An established, easy to follow framework will assist investigators in narrowing their search when faced with terabytes of data providing a roadmap to follow for finding usable forensic artifacts.

The following systems will be used for testing purposes for this study:

- Fujitsu Lifebook laptop PC, model T725 with a clean version of Windows 10 installed
- Apple MacBook Pro, model A1502 laptop with a clean version of macOS installed
- Newly created TAILS bootable USB drive with version 4.3 installed

- Tor installed on a new, unused SanDisk 64 GB USB drive

First, the Windows machine will be imaged using FTK Imager and the macOS machine imaged with Macquisition to produce a baseline image to ensure any data found during the examination stage is created during testing rather than residing on the system. RamCaturer for the Windows machine and Macquisition for the macOS machine will also be run to set a baseline on both volatile and static memory (PAGEFILE.SYS in the case of Windows). The open source tool Regshot will also be run on the Windows machine to get a baseline Windows registry snapshot. Regshot is a tool that can monitor Windows registry changes. Once each baseline has been created, each system will individually be booted using the TAILS operating system, Tor opened, several sites which will be documented accessed both on the Clearnet and dark web before exiting TAILS. Each machine will be restarted upon exit of TAILS and a memory capture performed to determine if any residual trace resides after using TAILS. A second image will then be captured on each machine to determine what, if any, traces of having used Tor and visited dark web sites remains. Regshot will be run on the Windows machine again to get a registry snapshot to compare to the initial.

Second, each machine will be booted into the installed operating system (PC or iOS) and the Tor browser bundle will be started from a USB drive. Several sites will be visited, including Clearnet and dark web sites before exiting Tor. System memory will be captured using RamCaturer for the Windows operating system and Macquisition for the macOS operating system. A system image will be created using FTK Imager for the Windows operating system and Macquisition for the macOS system. Regshot will also be run on the Windows machine again to get another snapshot for comparison to the original. Once completed, each machine will be powered off to erase any volatile memory retained by the system.

Third, each machine will download and install the Tor browser bundle locally, then started. Several sites will be visited, including Clearnet and dark web sites before exiting Tor. System memory will be captured using RamCaturer for the Windows machine and Macquisition for the macOS machine and a system image created using FTK Imager for the Windows machine and Macquisition for the macOS machine. Regshot will again be run on the Windows machine to gain a snapshot for comparison to the original.

Step	Activity	Tool or technique
1	Baseline Image	FTK Imager/Macquisition
	Baseline Memory Capture – Volatile and Non-Volatile	Ram Capturer/Macquisition
	Beginning Windows Registry Snapshot	Regshot
	Boot to TAILS	Tor opened, both Clearnet and dark web sites visited and authenticated, sites bookmarked
	Second Memory Capture	Ram Capturer/Macquisition
	Second System Image	FTK Imager/Macquisition
2	Boot to Operating System (Windows 10 and macOS)	Tor opened from USB drive, both Clearnet and dark web sites visited
	Memory Capture	Ram Capturer/Macquisition
	System Image	FTK Imager/Macquisition
	Windows Registry Snapshot	Regshot
3	Boot to Operating System (Windows 10 and macOS)	Tor opened from desktop, both Clearnet and dark web sites visited
	Memory Capture	Ram Capturer/Macquisition
	System Image	FTK Imager/Macquisition
	Windows Registry Snapshot	Regshot

Table 1 - Research Steps

Extraction and Analysis

The software used to perform forensic extraction and analysis:

- Forensic Toolkit (FTK) Imager
- Macquisition
- RamCapturer
- Forensic Toolkit (FTK)
- Magnet Axium
- Regshot

The captured images at each stage of research will be examined with both FTK and Axiom as each forensic software has advantages and strengths.

Framework Creation

Using the structure of the open-source OSINT Framework created by Justin Nordine, a reusable framework will be created that will allow investigators to click through a pathway to locate the artifacts they are seeking. Using the OSINT Framework is a mechanism for displaying and using the proposed framework rather than creating a paper-based framework to follow. A basic starting point of this framework is shown in Figures 15-19. The branching of the framework will be determined by the operating system being used; macOS, Windows, or Tails. Once the operating system is chosen further branching will indicate whether Tor is installed locally on the hard drive of the system or if it is started from a removable USB drive while booted into the operating system. Upon the selection of the mechanism for starting Tor, further branching will be revealed to indicate areas to be searching such as PAGEFILE.SYS, Windows Registry, or the SQLite database PLACES.SQLITE which stores bookmarks, favorites, and browsing history. Indications of cryptocurrency may be found in any of the above locations and will warrant branching from the main line. Further branching will be added during the course of the research as more indications of dark web use are located.

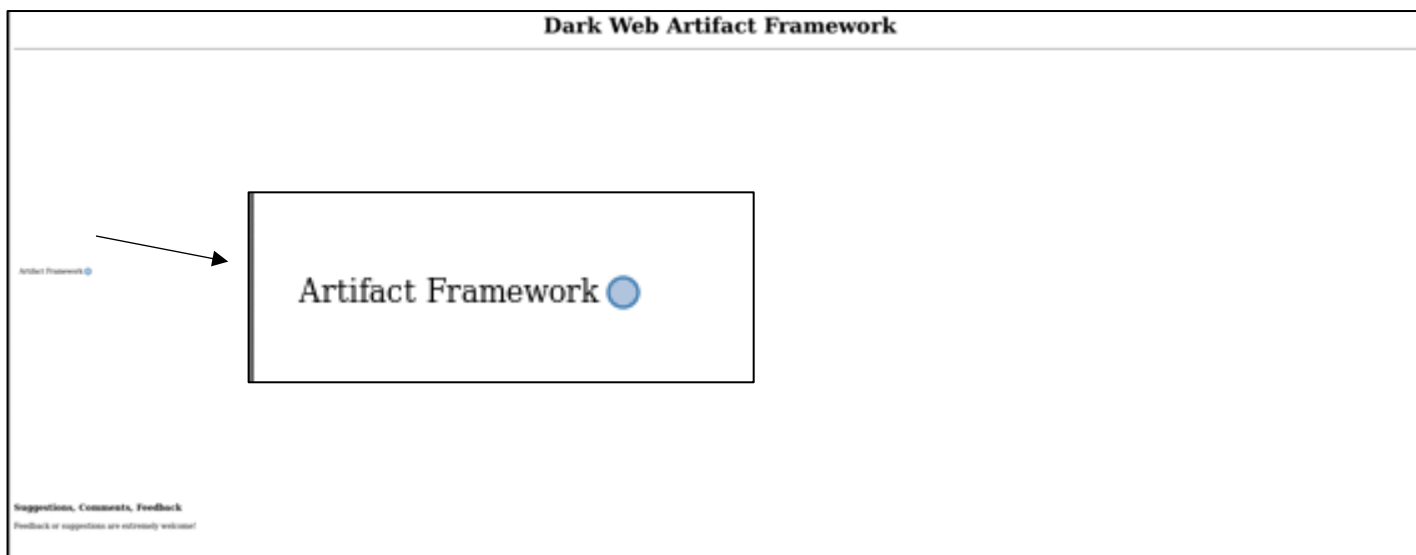


Figure 14 - Start of the framework

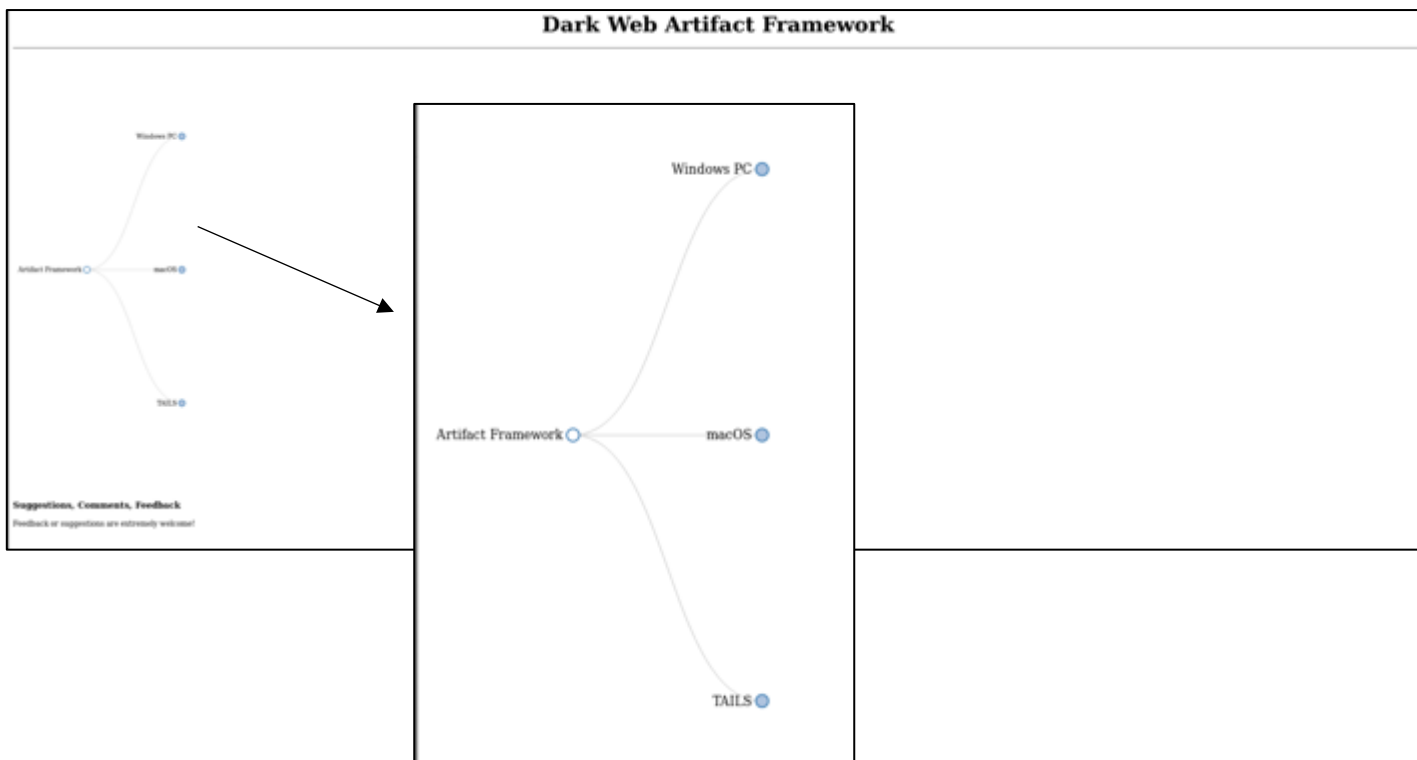


Figure 15 - Framework showing options of Windows PC, macOS, and TAILS

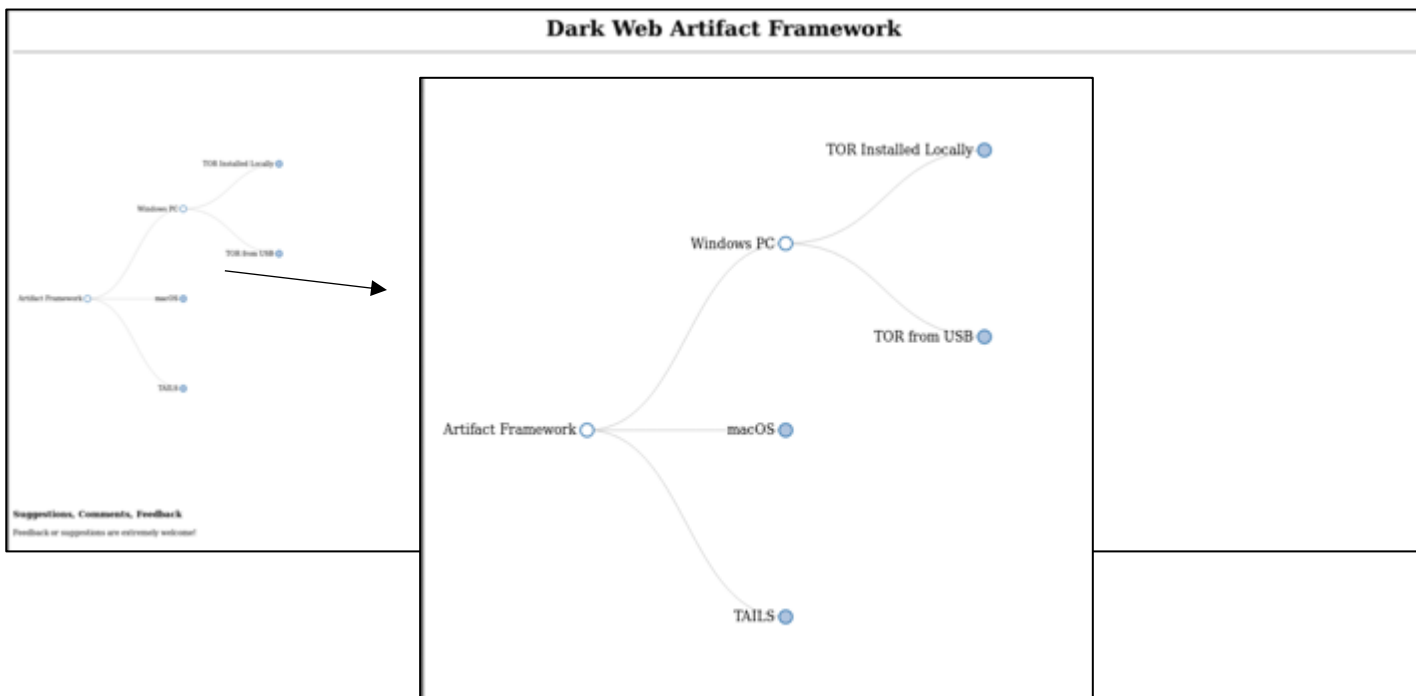


Figure 16 - Further branching from Windows PC showing Tor installed locally vs. Tor run from a USB drive.

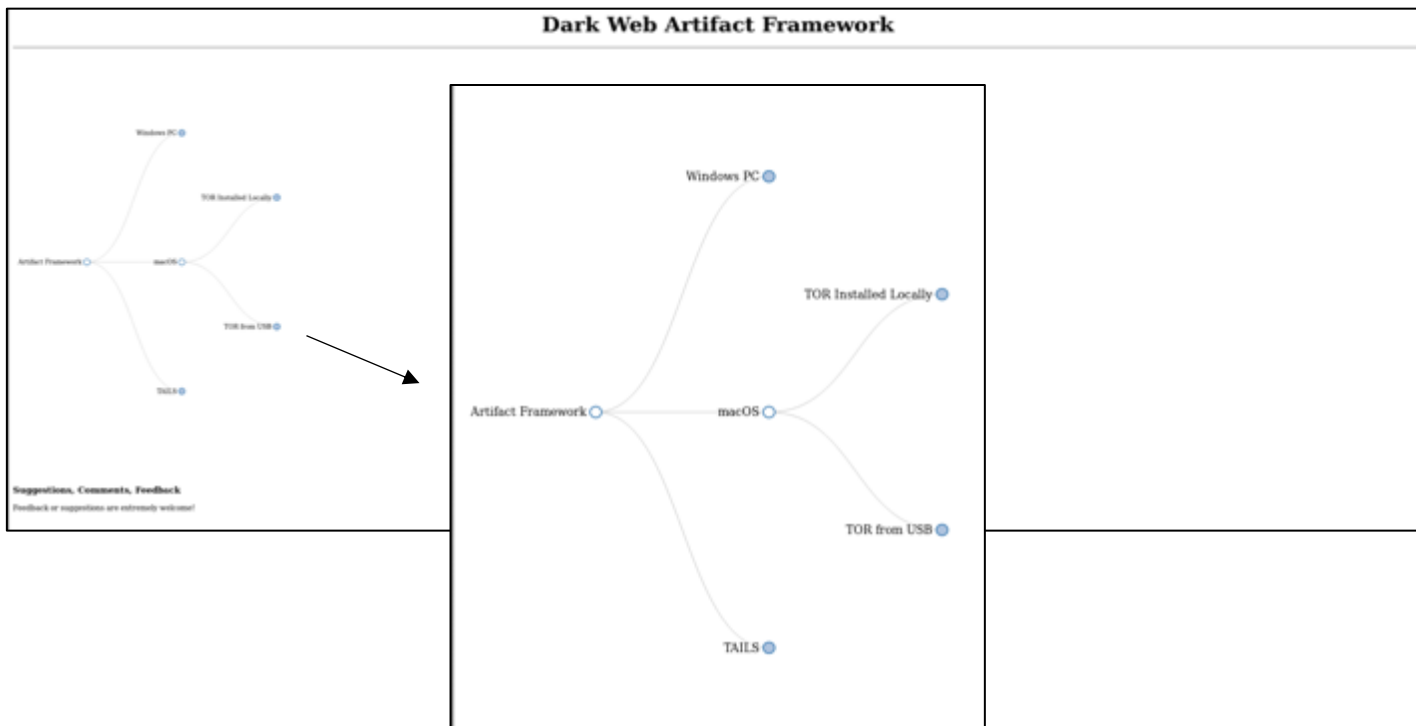


Figure 17 - Further branching from macOS showing Tor installed locally vs Tor run from a USB drive

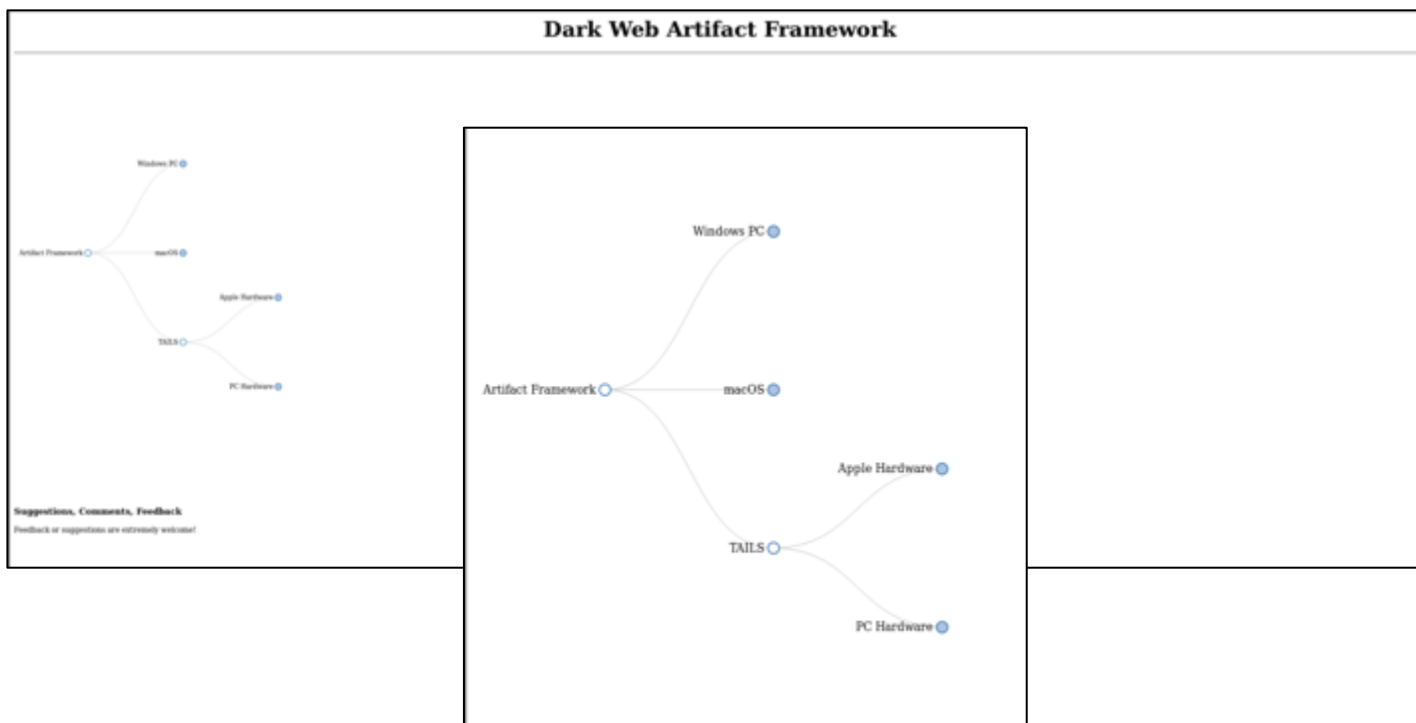


Figure 18 - Further branching of TAILS showing it being run on both Apple and PC hardware.

Expected Contribution

Today, nearly every crime has a digital component. Investigators invest their time in these cases in an attempt to solve serious crimes. The goal of these stakeholders is to find usable evidence of dark web activity to assist their investigations. Digital evidence can be found in different locations depending on the operating system being used and the method of accessing the dark web. Windows stores information differently than a macOS system. Accessing the dark web using Tor installed on a hard drive leaves different traces than using Tor from removable storage. The ultimate goal of this research is to provide investigators with a method that considers both operating systems and multiple methods for accessing the dark web to maximize the amount of evidence located.

The artifact created for this design science research will be a reusable, comprehensive framework for investigators to follow to find usable evidence that may be hidden in the vast amounts of data they may be searching. Rather than simply displayed in paper form, this framework will utilize the existing structure of the OSINT Framework created by Justin Nordine (Nordine, n.d.) which is available as open-source software on GitHub. The underlying design of the OSINT Framework will be modified to create a paperless framework that can be installed for the user. Design science is the most appropriate method for this type of research project as it focuses on creating an artifact, a framework to follow, to solve the problem of locating evidence of dark web use on a computer that may contain terabytes of data. The use of the open source software that was the basis for the OSINT Framework is simply a tool to facilitate the use of the artifact, the created framework.

Treatment Evaluation

Treatment evaluation studies the designed artifact to determine if the effects of the artifact meet the requirements specified by the researcher and ultimately contribute to the goal of the stakeholders. According to Hevner et al., "A design artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to solve." (Hevner et al., 1996).

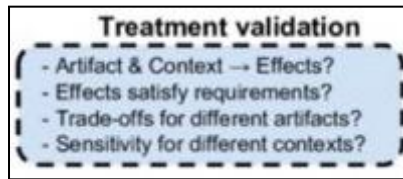


Figure 19 - Treatment validation (Martakis, 2015)

To ensure the validity of the process created during this design science project the following steps will be taken. First, experimental evaluation will be conducted using a simulation of the proposed framework. A clean system will be given to a voluntary user to access dark web content using each of the three methods described above accessing several dark web sites. After the completion of each method, the created framework will be used by an investigator to identify artifacts generated as a result of this dark web use. Second, the image file from a previously worked dark web case will be used both with and without the framework. A comparison will be made using metrics such as total time for the analyst, ease of finding artifacts, and error rate such as false positives and false negatives. Third, the expert opinion of members of the South Dakota Internet Crimes Against Children taskforce (ICAC) and the Division of Criminal Investigation (DCI) will be sought. DCI and ICAC investigators investigate many crimes including illegal drug purchases and child pornography. These agents analyze numerous systems and have expert knowledge of the evidence needed to charge and ultimately aid in convicting criminal suspects.

The framework created will demonstrate success during both validation steps as artifacts of dark web activity can be located on the test systems. These artifacts would include, but are not limited to, items such as .onion sites found in memory analysis, bookmarked sites in SQLite databases, and Windows registry changes indicating the installation and use of the Tor browser bundle. Identifying .onion sites in memory will aid investigators in knowing which dark web sites the suspect is visiting. Sites bookmarked in Tor or Tails stored in the SQLite database PLACES.SQLITE give indications of the favorite markets or forums of the user. Windows registry changes indicate when the suspect first installed Tor and when it was last used or possibly deleted. Indications of cryptocurrency use in memory, wallet keys, or bookmarked favorites contributes to the evidence of dark web use as cryptocurrency is the payment method during dark web transactions. Success of this framework will provide artifacts to investigators

that will contribute to the evidence in cases where there is suspicion of a crime being committed by using the dark web.

Timeline

The following timeline includes steps already completed, steps in process, and steps proposed to complete this research.

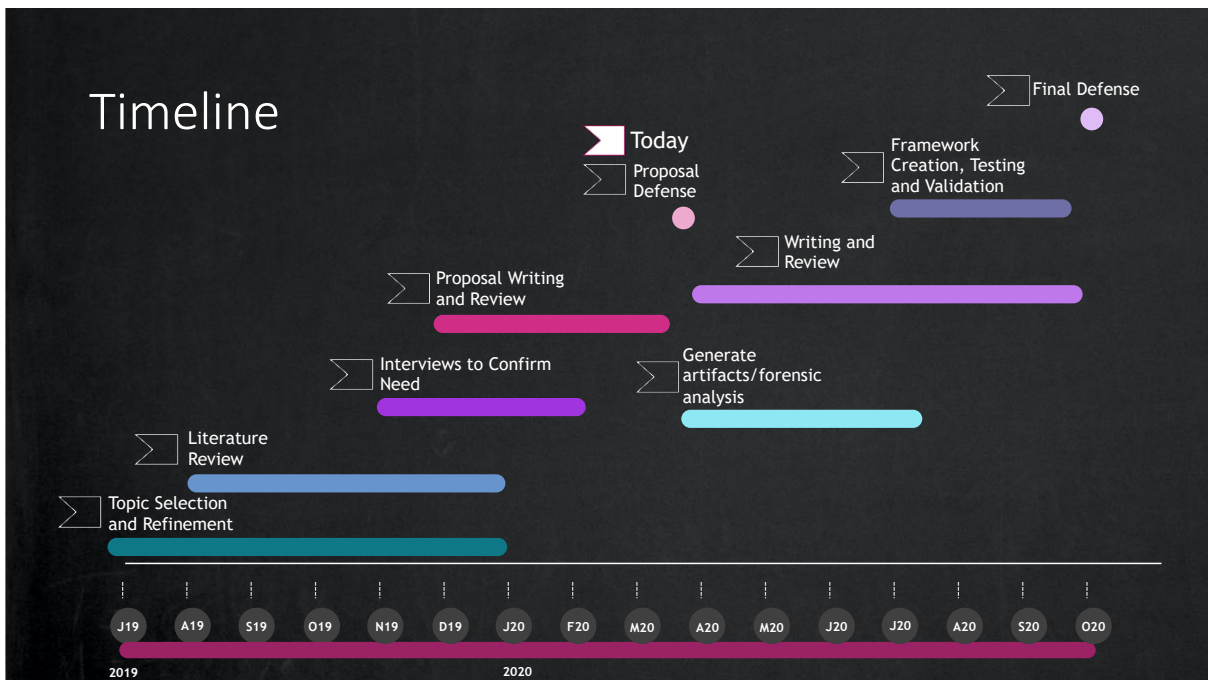


Figure 20 - Timeline

Limitations

A limitation of this study is that this is covering only artifacts that are left on a host while using the Tor browser to access the dark web. Other software exits for accessing dark web sites including I2P and Freenet, however, Tor is the most popular browser used to access the dark web (“5 Dark Web Browsers for Deep Web Browsing in 2018,” n.d.). Network forensics can be used to attempt to identify users of the Tor network by monitoring entry or exit nodes. Attempts can also be made to exploit vulnerabilities in the Tor browser to identify users of the Tor network and therefore expose their use. Both of these techniques are beyond the scope of this research.

Summary

This chapter addresses the methodology to be used to complete this research project. Design science research focuses on the creation of an artifact to address and add value to a real-world problem. The dark web is not well understood by law enforcement. It provides both anonymity and encryption which often leads to criminal activity such as illegal drug purchases and child pornography. The purpose of this research is to produce an artifact that is a framework for digital forensic investigators of dark web crimes to follow to assist in locating evidence on a host-based system. The data gathered for this research will use both a PC laptop with Windows 10 installed and an Apple Laptop with macOS. Testing will be done to locate dark web artifacts if Tor is installed locally or if Tor is run from a USB drive. TAILS operating system will be tested by on Apple hardware and PC hardware. Validation for this research will be sought from South Dakota ICAC task force and DCI investigators. The ultimate goal of the artifact designed for this research would be to assist law enforcement in their investigations of dark web crimes by providing an easy to follow framework for locating evidence of dark web use on a computer system.

CHAPTER 4 – ARTIFACT DESIGN

This chapter discusses the creation of the framework using the methodology described in Chapter 3 to answer the questions posed to facilitate the design. The Dark Web Artifact Framework, found at <https://arica-kulm.github.io/Dark-Web-Artifact-Framework/>, was created using the open source OSINT Framework by Justin Nordine as a template (Nordine, n.d.). When opening the framework, circles in blue indicate further branching while circles in white represent URL links in most cases. Starting with each operating system, the branching begins with a decision of whether Tor was installed locally or from a USB or secure digital (SD) card. If Tor was installed locally, the next branching provides options for whether Tor is still installed or has been deleted from the system. The Tails branch provides steps for an investigator to follow to first determine whether or not there is persistent storage on the imaged Tails drive. If persistent storage exists, steps are provided for attempting to decrypt and reimage to identify artifacts that the user saved. The Dark Web Artifact Framework contains branching for each option that was researched with the end branch containing a link to a URL for further information on that particular artifact. An effort was made to avoid sites that were for commercial purposes or user forums in favor of those that contain articles or other useful information. Sites that link to commercial sites link to those that contain blogs or other helpful information and are not a link to a tool or product. The sites referenced are not intended to be academically rigorous.

The default preferences for security and privacy in Tor were not changed for this research. Users have the option to use a master password which enables saving logins and passwords, site history can be remembered, and the security level can be changed to a different level. None of those options were changed from their defaults, but rather used as installed. The artifacts listed in the Dark Web Artifact Framework may not always be found on a suspect's system and there may be additional artifacts that are not included in this framework. Both Windows PC and macOS operating systems were booted using a USB drive with Tails installed, used with the Tor Browser launched from a USB drive, and used with the Tor Browser installed locally. Separate USB drives were used for booting to Tails and separate USB drives were used for each operating system when the Tor Browser was launched from the USB. Each operating system has unique artifacts that remain after using the Tor Browser either locally installed or

launched from a USB drive or SD card. Artifacts were also gathered both with the Tor Browser still installed on the local system and again after it was deleted. The purpose of this research is not to compare the number of artifacts found for each operating system, but rather to identify what artifacts are found leaving quantification to future research.

Anonymizing Methods or Tools

What methods or tools are used to remain anonymous when accessing the dark web?

Tor or another anonymizing browser is required to access the dark web. As outlined in the introduction, Tor uses a series of relays to avoid detection and tracing (“Tor Project,” n.d.). It is important to emphasize that methods to remain anonymous when accessing the dark web are different depending on the user’s intent. Remaining anonymous to the site being connected to is different than leaving no trace on the user’s host system. The goal of the Dark Web Artifact Framework is to determine host-based artifacts, so the user’s anonymity to the site being connected to is not within the scope of this research. Refining the question being asked is necessary to fit the scope of this framework. The question then becomes, “What methods or tools are used to minimize any trace of host-based artifacts on the system used to access the dark web?”. There are a variety of ways this can be accomplished and is largely determined by the sophistication of the user. The methods used for this research included Tor run from within the stand-alone operating system Tails, Tor installed on a removable drive such as a USB drive and launched after booting into the operating system, and Tor being installed and launched locally. All of these methods were performed both on a macOS and a Windows PC computer. There are other methods that can minimize the host-based artifacts that would remain on a system, such as running Tor in a virtual machine, that were not tested for this research. Using these methods, the artifacts that remain under each circumstance were documented and used to create a clickable, reusable framework. The artifacts are described below and displayed in screenshots of the created Dark Web Artifact Framework. Dots on the framework that are blue indicate there is further branching while dots that are white indicate either a stopping point or a link to a URL explaining the artifact that is found.

Tor Installed Locally

The Tor Browser is available to be installed on different platforms and in 32 different languages. Tor is available for Windows, OS X, Linux, and Android (“Tor Project,” n.d.). The Tor Project website indicates the safest way to download the Tor Browser is directly from their website, however they acknowledge it may not always be accessible if it is blocked by the network of the user. Other alternatives include mirror sites or GetTor (“Tor Project | What is GetTor?,” n.d.). GetTor is a service that automatically responds to user requests for a link to download the Tor Browser from hosted locations such as DropBox, Google Drive, and GitHub. GetTor can also provide the Tor Browser via email or via Jabber. On a Windows system the downloaded file is an .exe file, on macOS it is a .dmg file. Launching the downloaded file initiates the install just as other software is installed. The user has the option to change the default install location, however for this research the defaults were accepted.

Windows Local Install

Installing Tor onto a Windows computer places the Tor Browser onto the Windows desktop which puts the files in the */Users/<user profile>/Desktop/* file path. This local install of the Tor Browser also leaves artifacts in many other locations. Below shows the branching of the framework with Tor installed locally while still installed on the system.

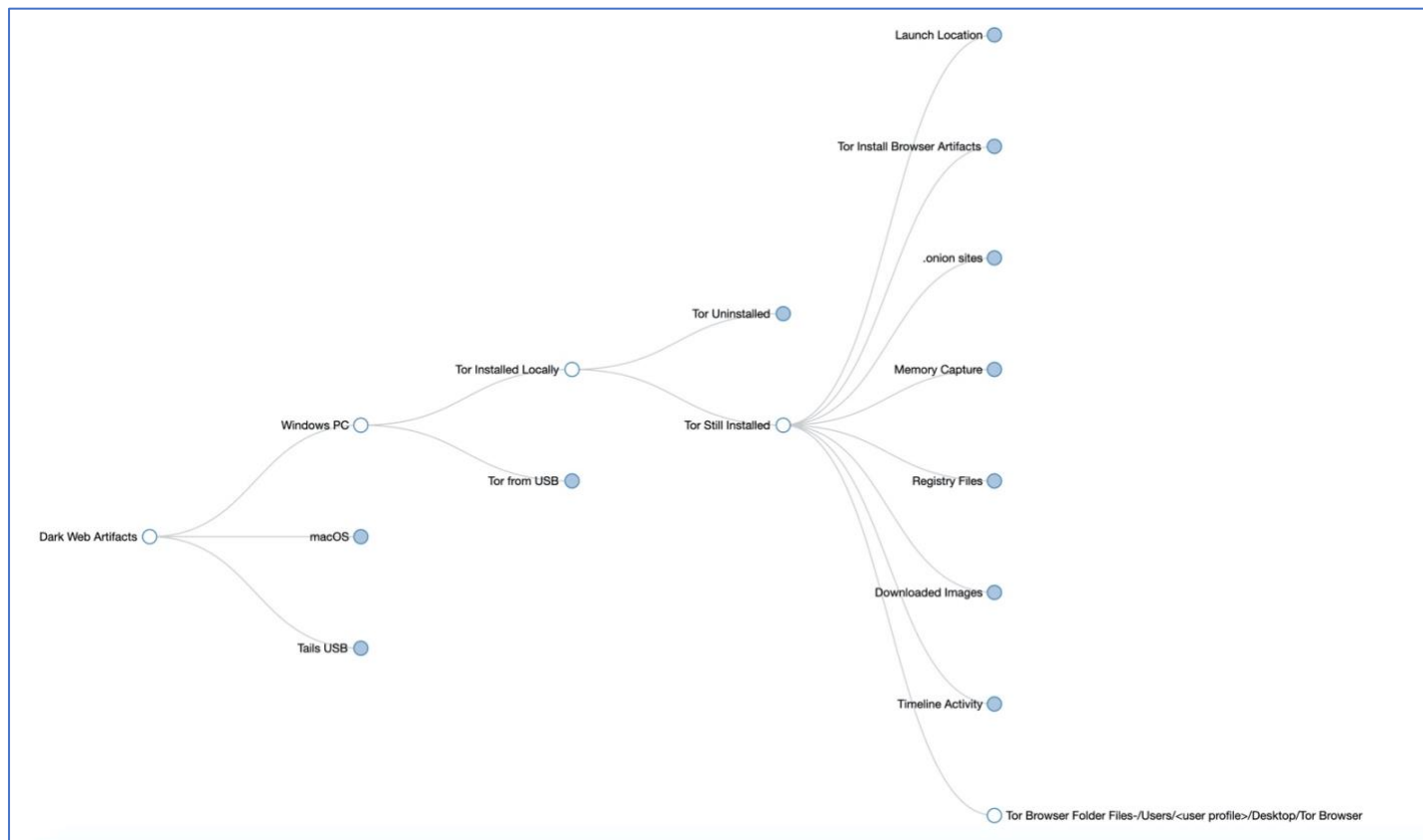


Figure 21 – Windows Tor Local Still Installed

Launch Location

In addition to the location of the Tor Browser install files which contains the executable, LNK files and other files needed to run Tor, there are several artifacts showing the launch location of Tor including:

- SRUM Application Resource/Network Usage - *SRUDB.dat*
- Windows Defender logs – *MPLog*.log*
- Amcache Registry hive files – *Amcache.hve*
- Event logs – *Application.evtx*, *Security.evtx*, *Microsoft-Windows-Security-Mitigations%4KernelMode.evtx*, and other *.evtx* files
- Prefetch Files – *firefox.exe*.pf*, *tor.exe*.pf*, *svchost.exe*.pf*, and other prefetch files
- *UsnJrnl* – a journal file used in Windows New Technology File System (NTFS)
- *\$LogFile* – NTFS logging file

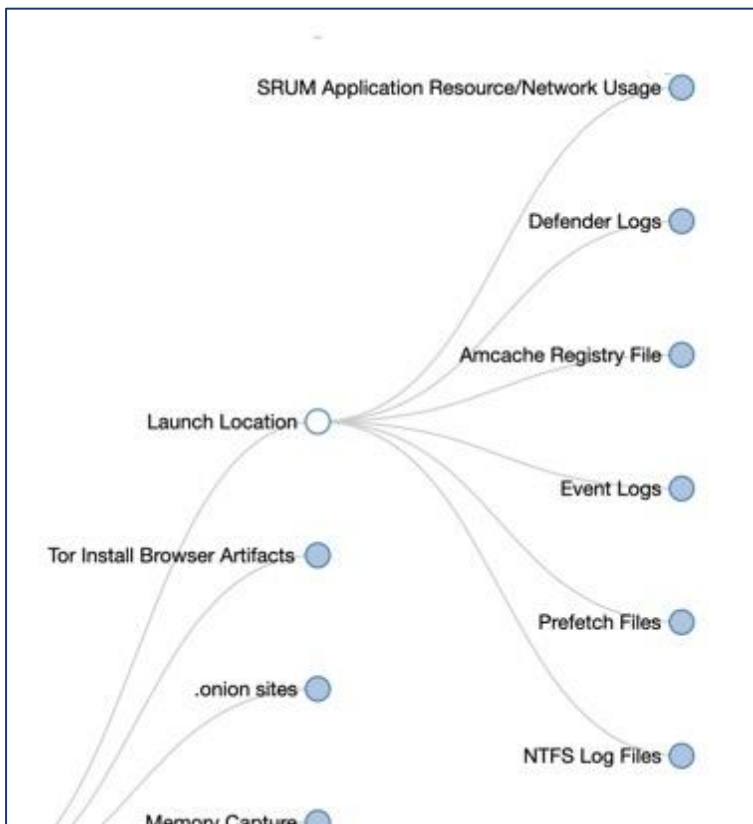


Figure 22 – Launch Location Artifacts

Tor Install Browser Artifacts

The Tor Project URL with the Tor executable file indicating that the Tor Browser was downloaded is found and is dependent on the browser used. For Internet Explorer and older versions of Edge it will be in *WebCacheV01.dat* file, for Chrome and newer versions of Edge it will be found in an SQLite History file within the Edge/Default folder.

.onion sites

Artifacts remain for .onion sites that were bookmarked or add-ons that were installed for .onion sites. *Places.sqlite* contains Tor bookmarks with titles stored in the MOZ_BOOKMARKS table and the corresponding .onion site is stored in both the MOZ_ORIGINS table and MOZ_PLACES table. Artifacts left behind in the *places.sqlite* file are dependent on whether or not the user creates a bookmark for a site. Users who are knowledgeable or cautious may not bookmark sites in Tor, leaving no artifacts behind in *places.sqlite*. Add-ons are found in the *extensions.json* file found in the Tor Browser install location (“Find and install add-ons to add features to Firefox | Firefox Help,” n.d.). Like

bookmarks, if users do not install any add-ons there will be no artifacts from dark web sites in the *extensions.json* file.

Memory Capture

A capture of system memory either while the computer is still running after having accessed the dark web, or after a restart, reveals artifacts of .onion sites, images from dark web sites, and LNK files indicating where the Tor Browser was launched from. Capturing memory after the computer has been used to access the dark web, but prior to a restart, contains the most artifacts. Restarting or shutting down the computer removes the artifacts that are left in volatile memory; however, artifacts may remain in the *pagefile.sys* or *hibefil.sys* files. Artifacts that can be found include .onion site names, images from dark web sites visited such as marketplaces, and Tor Browser icons. URLs to dark web sites are generally not obvious in determining whether the site that was visited is nefarious. To aid in determining what sites have been accessed, using Magnet AXIOM forensic software or another forensics tool, a report can be created by exporting the evidence found in either a comma-separated value (CSV) file or an Excel spreadsheet file for further searching to determine the site contents.

Registry Files

There are multiple locations in the Windows Registry that contain artifacts from Tor Browser. These are addressed in an upcoming question specific to the Windows Registry.

Downloaded Images

Images that are downloaded from sites while using the Tor Browser are saved to the user's Download folder.

Windows Timeline Activity

The *ActivitiesCache.db* file contains the URL of the Tor Project download site along with the location the Tor Browser install executable file that was downloaded which can indicate which user downloaded the file.

macOS Local Install

Installing the Tor Browser on macOS requires downloading the .dmg file. Once launched, Tor Browser is installed when the user drags it to the *Applications* folder or other location (“How to: Use Tor on macOS | Surveillance Self-Defense,” n.d.).

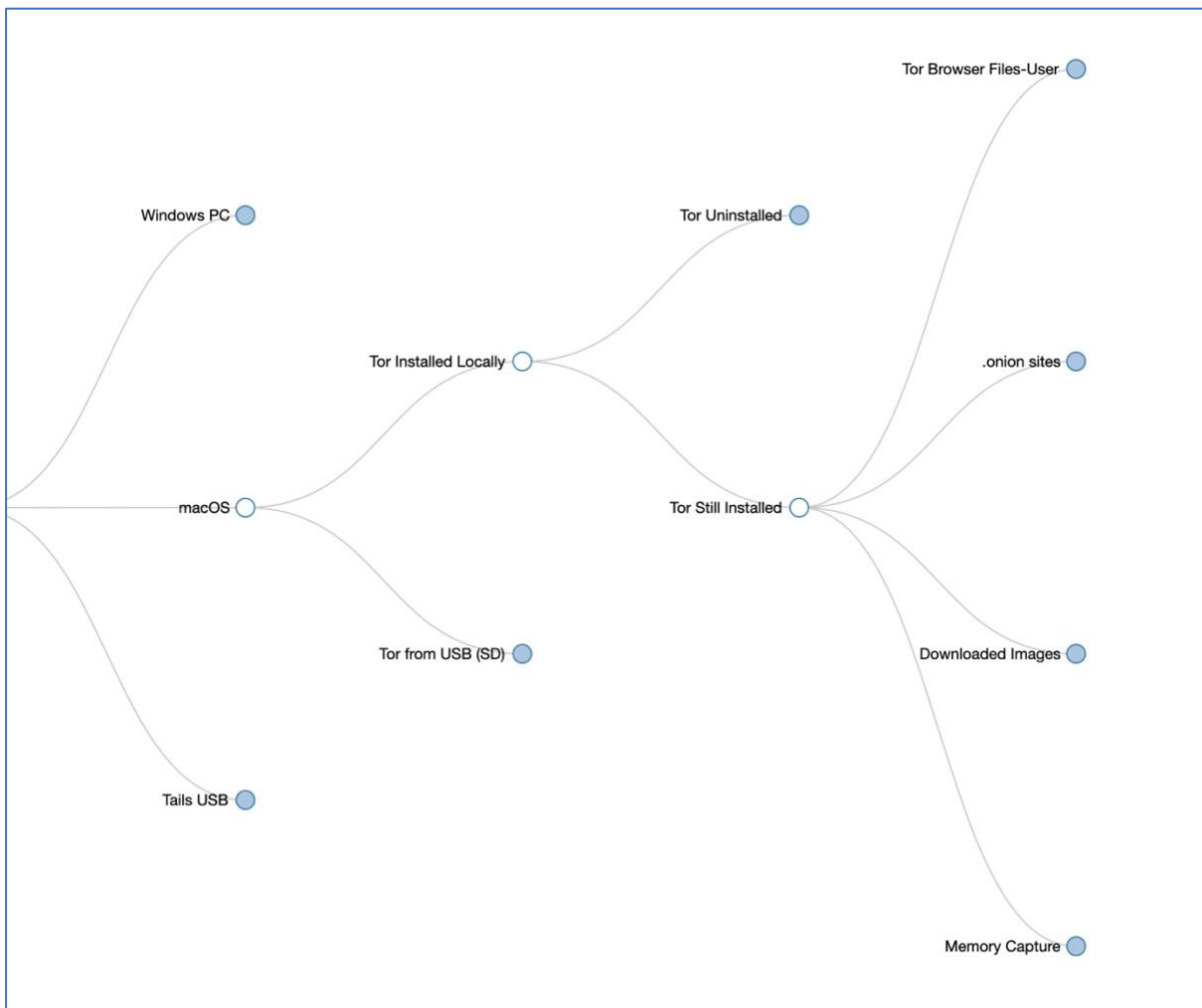


Figure 23 – macOS Tor Installed Local Artifacts

Tor Browser Files – User

In addition to the Tor Browser application files that are installed, files that are installed when a user installs and runs the Tor Browser can be found under that user in numerous artifacts.

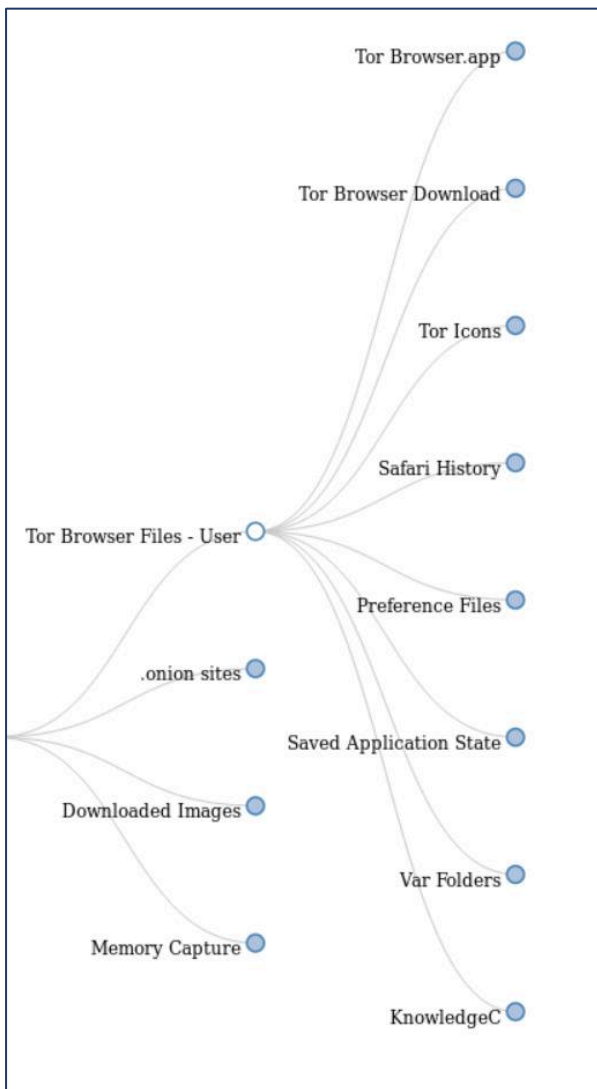


Figure 24 – macOS Tor Browser User Files Artifacts

Tor Browser Download

macOS contains an *Application Support* folder for users that contains support files for running an application. For the Tor Browser, the *TorBrowser-Data* folder is created and contains the supporting files for running the Tor Browser.

Tor Icons

Images from the Tor Project website can be found in the *WebKitCache* folder for the user who installed the Tor Browser.

Safari History

Safari history is located in the user's *Library\Safari* folder and includes files that contain links to the Tor Project website. The *Downloads.plist* file for the user contains the URL

of the *TorBrowser.dmg* file and includes the date and time the file was downloaded. *History.db* contains a link to the Tor Project site including the `\download\` folder. *RecentlyClosedTabs.plist* also contains a link to the Tor Project website `\thank-you\` page.

Preference Files

Tor Browser information is found in *.plist* and other files in the `\Library\Preferences\` folder including dock items in the *com.apple.dock.plist*, quarantined files in *com.apple.LaunchServices.QuarantineEventsV2*, and volume and device identifier information in *com.apple.finder.plist*.

Saved Application State

The Saved Application State is a macOS feature that saves the open screens of an application when the application is terminated (“Controlling Saved Application States - krypted,” n.d.). The user profile `\Library\` folder contains an entry for *org.torproject.torbrowser.savedState*.

Var Folders

The `\private\var\` contains both file system event files and daily log file with Tor Browser artifacts. The daily log file, *daily.out*, contains disk usage and network information (“Mac OS Daily Logs | Salt Forensics,” n.d.). The file system events files provide file and folder modification and creation information and contains several references to the Tor Browser.

KnowledgeC

The *knowledgeC.db* file contains an event log of processes that run within an Apple device. The ZOBJECT table within *knowledgeC.db* contains a type called App/InFocus that shows the date and time an application is being used (“Knowledge is Power! Using the macOS/iOS knowledgeC.db Database to Determine Precise User and Application Usage — mac4n6.com,” n.d.).

.onion sites

Artifacts remain for *.onion* sites that were bookmarked or add-ons that were installed for *.onion* sites on a computer running macOS as they are on a computer running Windows. *places.sqlite* contains Tor bookmarks with titles stored in the MOZ_BOOKMARKS table and the corresponding *.onion* site is stored in both the MOZ_ORIGINS table and MOZ_PLACES table. Artifacts left behind in the *places.sqlite* file are dependent on whether or not the user

bookmarks sites. Users who are knowledgeable or cautious may not bookmark sites in Tor leaving no artifacts behind in *places.sqlite*. Add-ons are found in the *extensions.json* file found in the Tor Browser install location (“Find and install add-ons to add features to Firefox | Firefox Help,” n.d.). Like bookmarks, if users do not install any add-ons there will be no artifacts from dark web sites in the *extensions.json* file.

Downloaded Images

As they are on a computer running Windows, images that are downloaded on a computer running macOS from sites while using the Tor Browser are saved to the user’s Downloads folder.

Memory Capture

A memory capture of a computer running macOS that has used Tor to access the dark web is similar to a computer running Windows. macOS does not contain the *pagefile.sys* file like Windows but does contain a swap file. Artifacts such as *.onion* addresses, Tor icons, and images from dark web sites are all found.

Launching Tor from Removable Media

Installing Tor onto a removable drive such as a USB drive or SD card places the Tor Browser install files onto the USB drive or SD card rather than the local hard drive of the computer that Tor is launched from. Users may incorrectly think that doing this leaves no artifacts, however that is not accurate. Launching Tor from a USB drive or SD card leaves artifacts on the computer and on the USB drive or SD card.

Windows - Tor Run from USB

Many of the same artifacts that are found when Tor is installed locally are found when Tor is launched from a USB drive or SD card such as the launch location, user information, and downloaded images. Additionally, information about the USB drive or SD card that launched Tor can be identified and matched to the USB drive or SD card if it is also located.

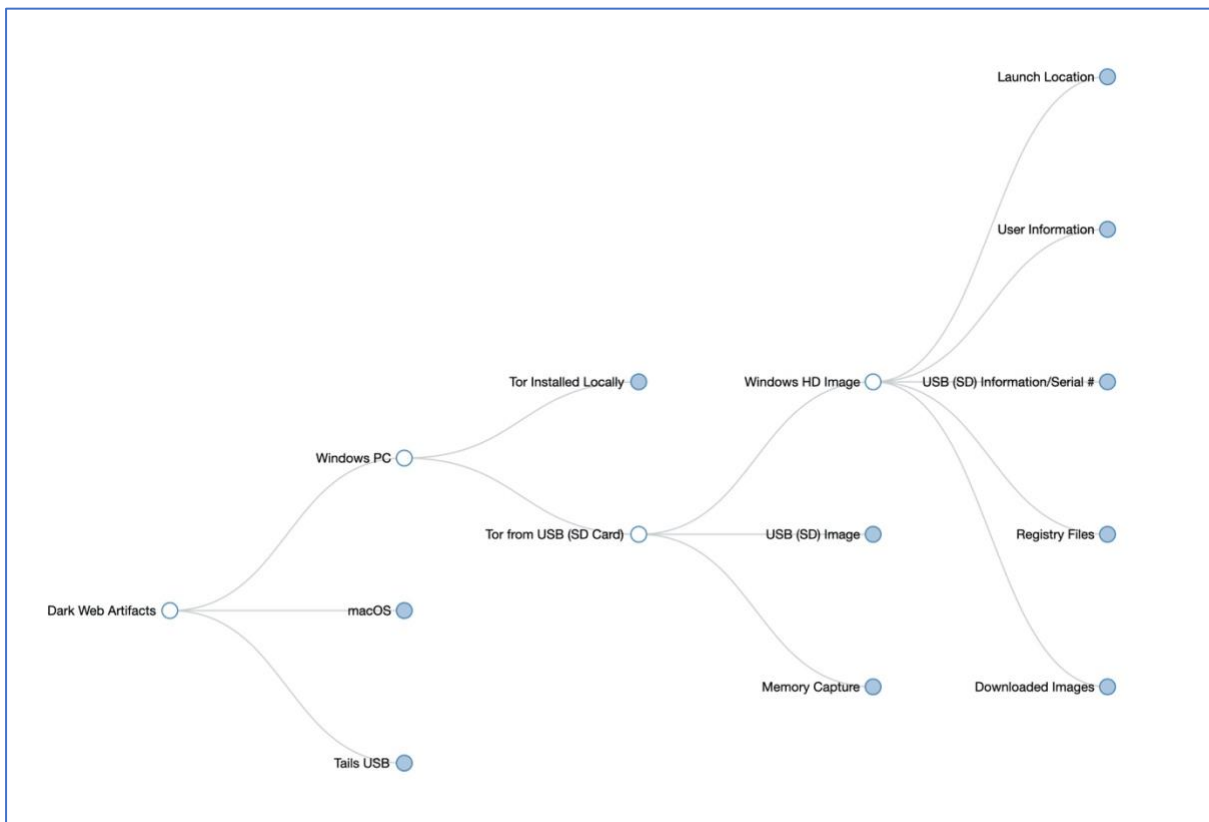


Figure 25 - Windows Tor from USB (SD Card)

Artifacts Found on Windows Hard Drive

Launch Location

The location Tor was launched from is found in multiple artifacts similar to Tor being installed locally on the hard drive. This is significant when running Tor from a USB drive or SD card as there will not be an installed program on the hard drive, it can be clearly shown where Tor was launched from.

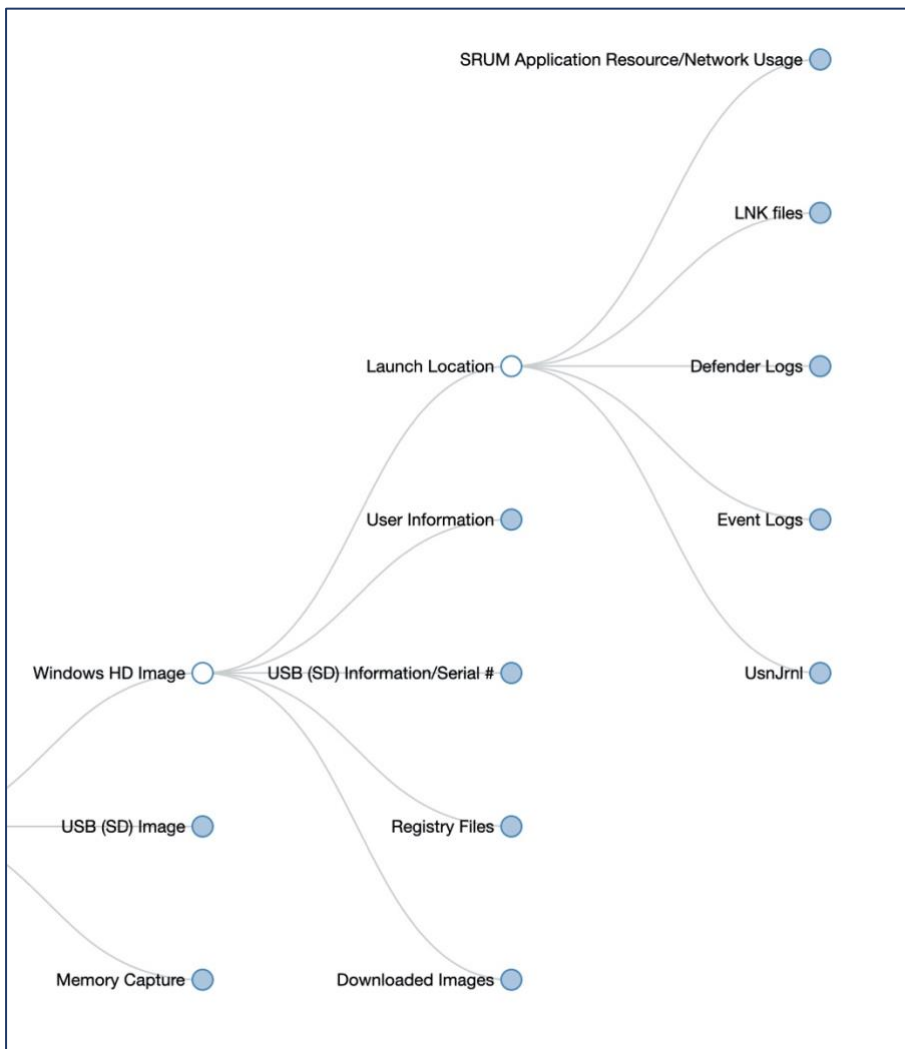


Figure 26 - Windows Tor from USB (SD card) Launch Location Artifacts

The launch location of Tor is found in the following:

- SRUM Application Resource/Network Usage - SRUDB.dat
- Windows Defender logs – MPLog*.log
- Event logs – Security.evtx and Microsoft-Windows-Security-Mitigations%4KernelMode.evtx
- UsnJrnl – a journal file used in Windows New Technology File System (NTFS)

User Information

The Windows timeline activity file *ActivitiesCache.db* contains information on the user that launched Tor. This file also shows the location Tor was launched from so could also be included in the launch location section of the framework, however, to avoid redundancy it is

included in this user information section. The Windows Shellbags file, *usrclass.dat*, also indicates which user launched Tor.

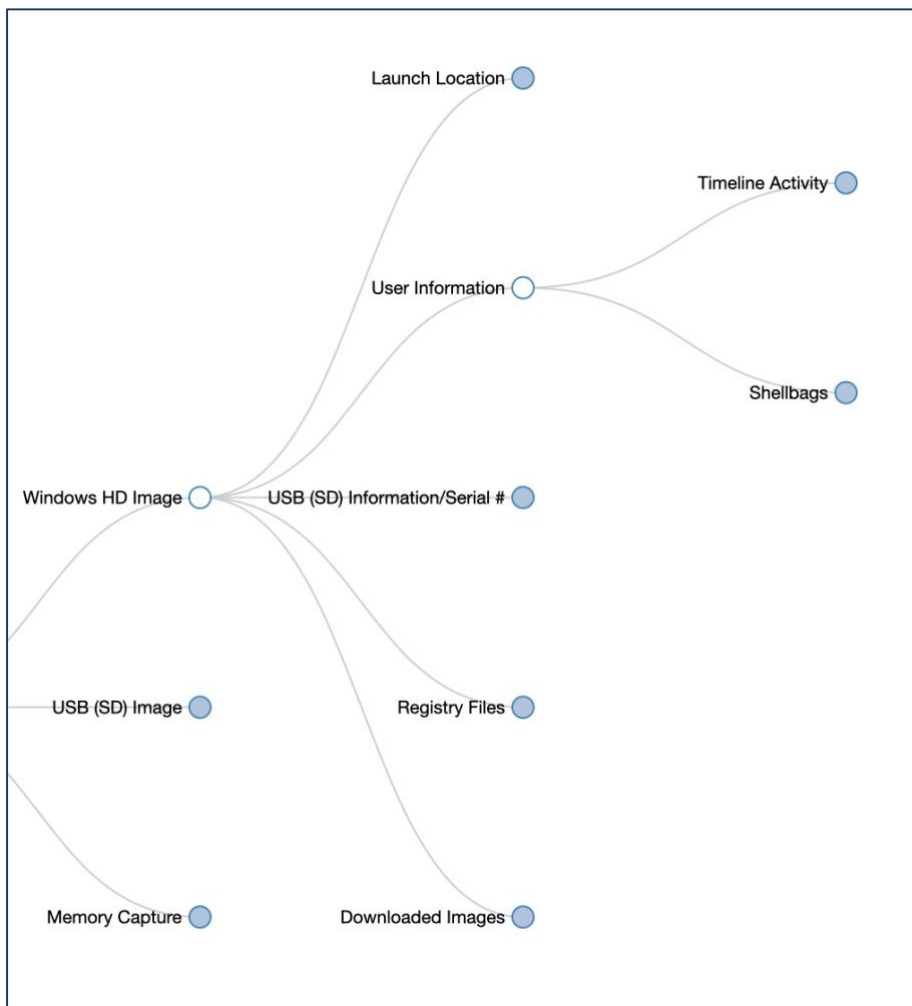


Figure 27 - Windows Tor from USB (SD card) User Information Artifacts

USB (SD Card) Information

Information about the USB or SD card that was used to launch Tor including the make, model, and serial number is found in the Windows Prefetch files for *TOR.EXE*, *FIREFOX.EXE*, and may be found in other prefetch files such as *SVCHOST.EXE*. The Windows compatibility appraiser also contains USB or SD card make and model information.

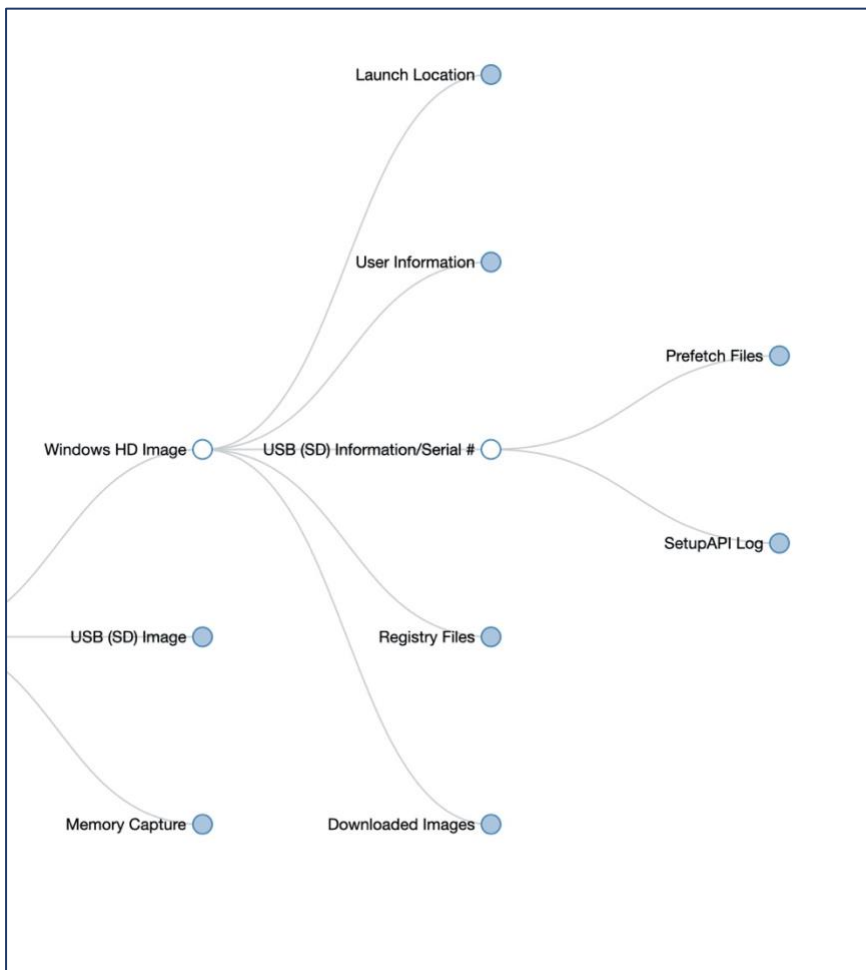


Figure 28 - Windows Tor from USB (SD card) USB Information

Registry Files

There are multiple locations in the Windows Registry that contain artifacts from Tor Browser. These are addressed in an upcoming question specific to the Windows Registry.

Downloaded Images

The default for images that are downloaded from sites while using the Tor Browser when launched from a USB drive or SD card is the user's Download folder rather than to the USB drive or SD card.

Artifacts Found on USB Drive or SD Card

If the USB drive or SD card is located, it can be shown through the artifacts on the computer and the USB drive or SD card that it had previously been used to launch the Tor Browser.

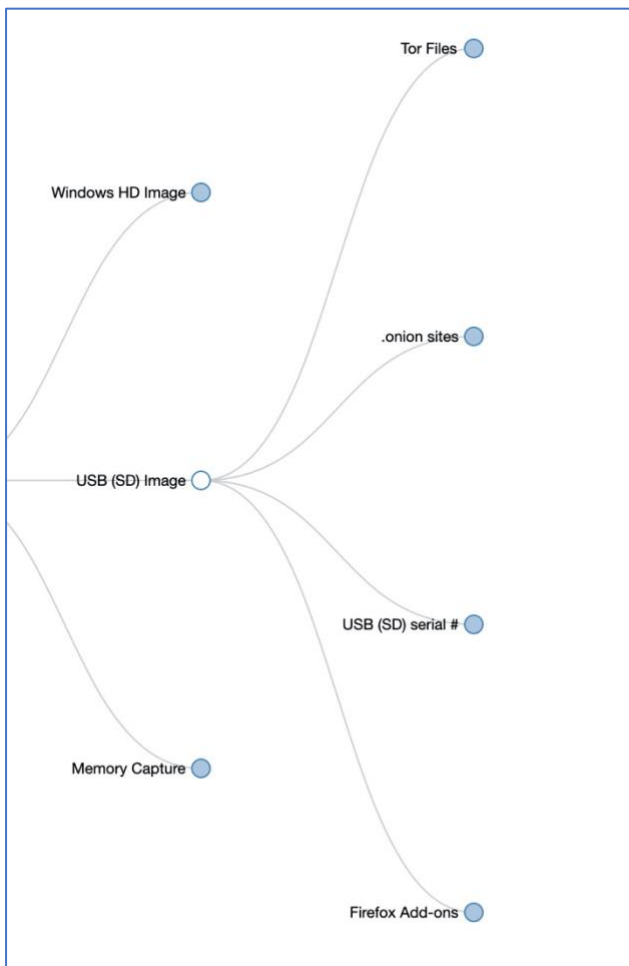


Figure 29 - Windows Tor from USB (SD card) - USB (SD card) Artifacts

The artifacts found on the USB drive or SD card include the following:

- Tor Files – the Tor Browser installation files
- *.onion* sites – *places.sqlite* file contains bookmarked sites
- USB serial # - the ‘*Start Tor Browser.lnk*’ file contains the serial number of the USB drive
- Firefox Add-ons – *extensions.json* file contains any add-ons the user has installed

Memory Capture

A memory capture of a computer that has run Tor from a USB drive or SD card is similar to installing and launching Tor locally. Capturing the memory after the computer has been used to access the dark web but prior to a restart contains the most artifacts. A capture of system memory either while the computer is still running after having accessed the dark web or after a

restart reveals artifacts of .onion sites, images from dark web sites, and LNK files indicating where the Tor Browser was launched from.

macOS Tor Run from USB or SD Card

Similar to launching Tor from a USB or SD card when using Windows, launching Tor from a USB while running macOS leaves artifacts on the hard drive, the USB drive, and in system memory.

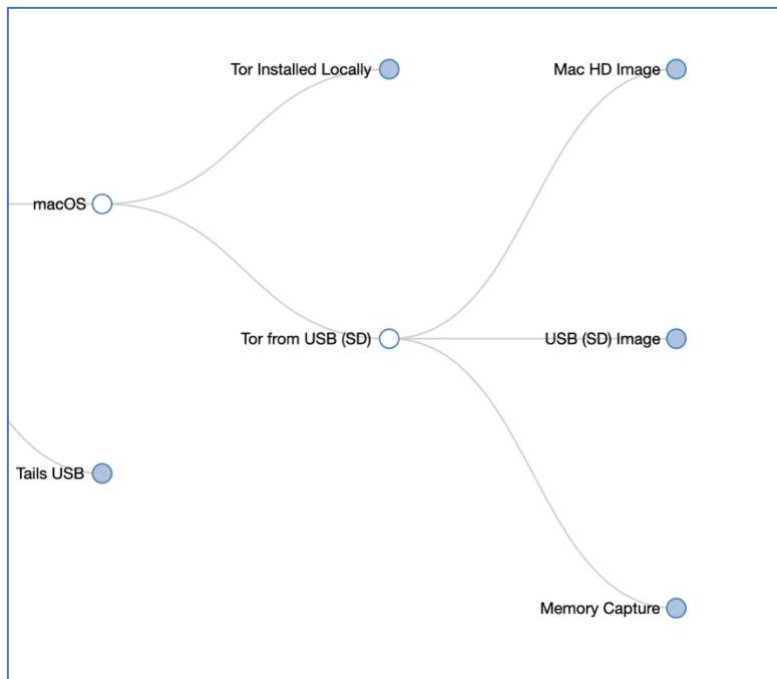


Figure 30 - macOS Tor from USB (SD card)

Artifacts Found on macOS Hard Drive

The artifacts found on the macOS hard drive including artifacts connected to the Tor Browser, artifacts containing information related to the USB drive used to launch of Tor, and downloaded images.

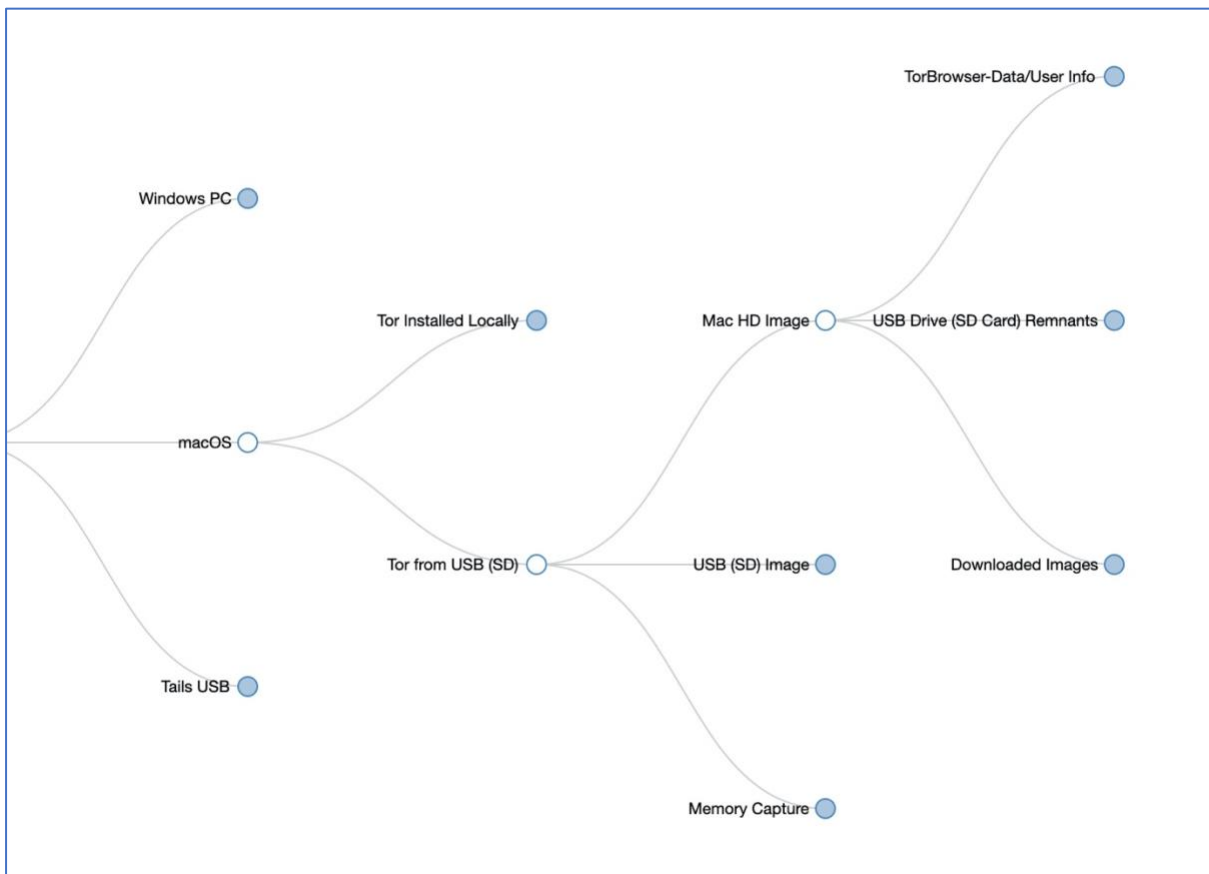


Figure 31 - macOS Tor from USB (SD card) HD Image

Tor Browser Data / User Info

Running Tor from a USB drive or SD card still places files for running Tor Browser in the Application Support folder like it does when installing Tor locally. The Application Support folder resides within the user profile indicating the user that launched Tor Browser. Artifacts also remain for preference files, the saved application state, and var folders when launched from a USB or SD card as they do when installed locally.

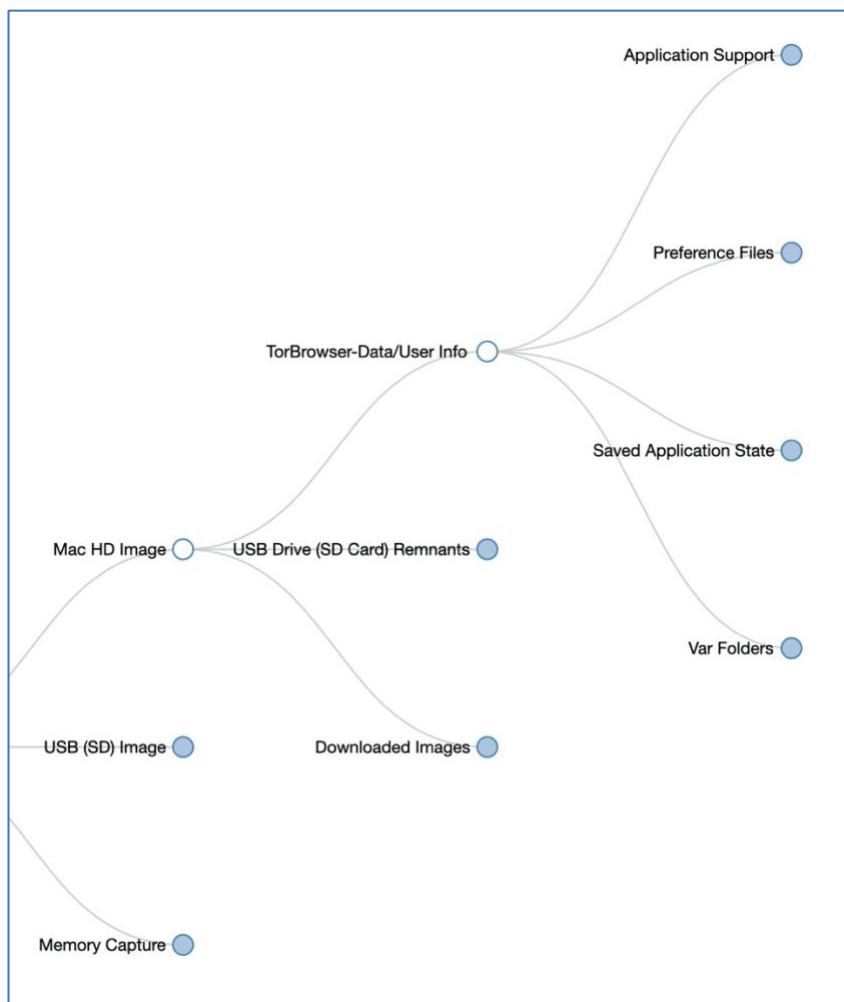


Figure 32 - macOS Tor from USB (SD Card) HD TorBrowser-Data/User Info

USB Drive or SD Card

Information about the USB drive or SD card used to launch Tor Browser can be found in the *CurrentPowerlog.PLSQL* file, specifically the *PLPeripheralAgent_EventForward_DeviceState* table.

Additional Obfuscation Steps

What additional steps are users taking to further obfuscate their dark web activity?

Using Tor itself is an obfuscation technique, however users who are using Tor with criminal intent may seek further methods of ensuring their activities are hidden. As stated in the previous question, Tor can be run from both a USB drive or SD card inserted into the computer, locally installed onto the hard drive of the machine being used, or as part of a bootable operating system. Users may incorrectly assume that by running Tor from a USB

drive or SD card that their system is safe since they are not installing to the local drive. Any of these methods can leave behind artifacts either on the host machine or on the method used such as a USB drive, SD card or the drive used to run Tails.

Users who install Tor locally on their system may delete Tor. The directions for removing Tor from a system can be located in the Tor Browser manual found on the Tor Project website (“UNINSTALLING | Tor Project | Tor Browser Manual,” n.d.). These directions call for the user of a Windows system to simply delete the folder or application and empty the trash. On a macOS system the directions are similar, move the application to Trash, then locate the *~/Application Support/* folder, look for *TorBrowser-Data* folder, move it to Trash, then empty the trash. The website notes that the operating system’s standard “uninstall” utility is not used (“UNINSTALLING | Tor Project | Tor Browser Manual,” n.d.). Users who are rushed or do not fully complete either of these instructions can miss the step of emptying the trash or deleting the *TorBrowser-Data* folder, so in creating this framework both the *Application Support* folder on macOS and the Recycle Bin on a Windows machine were incorporated into the possible locations for artifacts.

Users may take further steps of obfuscation such as clearing cache and cookies to remove any trace of having downloaded and installed Tor or using a cleaning program such as CCleaner (“CCleaner.com - What is CCleaner?,” n.d.) to remove unwanted files and browsing history. None of these options were performed during the creation of the framework so as to provide the most possible locations that artifacts could be found on a system.

Windows Local Install – Tor Deleted

Uninstalling Tor from a Windows system by following the recommendation on the Tor Project website removes the *Tor.exe* executable file, *places.sqlite*, *extensions.json* file, and other files contained within the *TorBrowser* folder but most of the other artifacts remain.

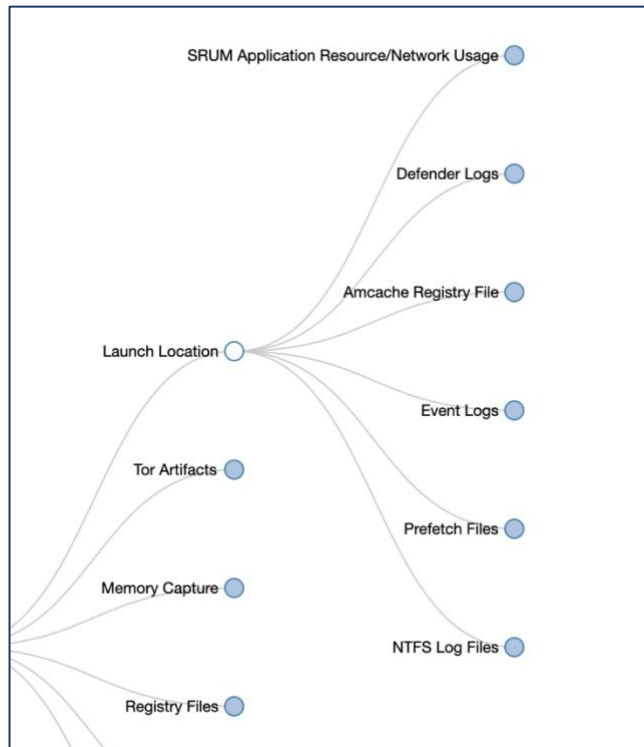


Figure 33 - Windows Local Tor Deleted Launch Location

If the system has not been shut down, a memory capture may still contain *.onion* sites, Tor Browser icons, and images of dark web sites. A user who wants to remove any trace of having downloaded, installed, and launched Tor will need to take extra steps beyond the recommendations.

macOS Local Install – Tor Deleted

The instructions on the Tor website for removing the Tor Browser from a macOS system are similar to removing Tor from a Windows system. Deleting Tor according to the directions leaves many artifacts behind on macOS. Like Windows, the *places.sqlite*, *extensions.json*, and other files that reside within the Tor Browser user files are deleted.

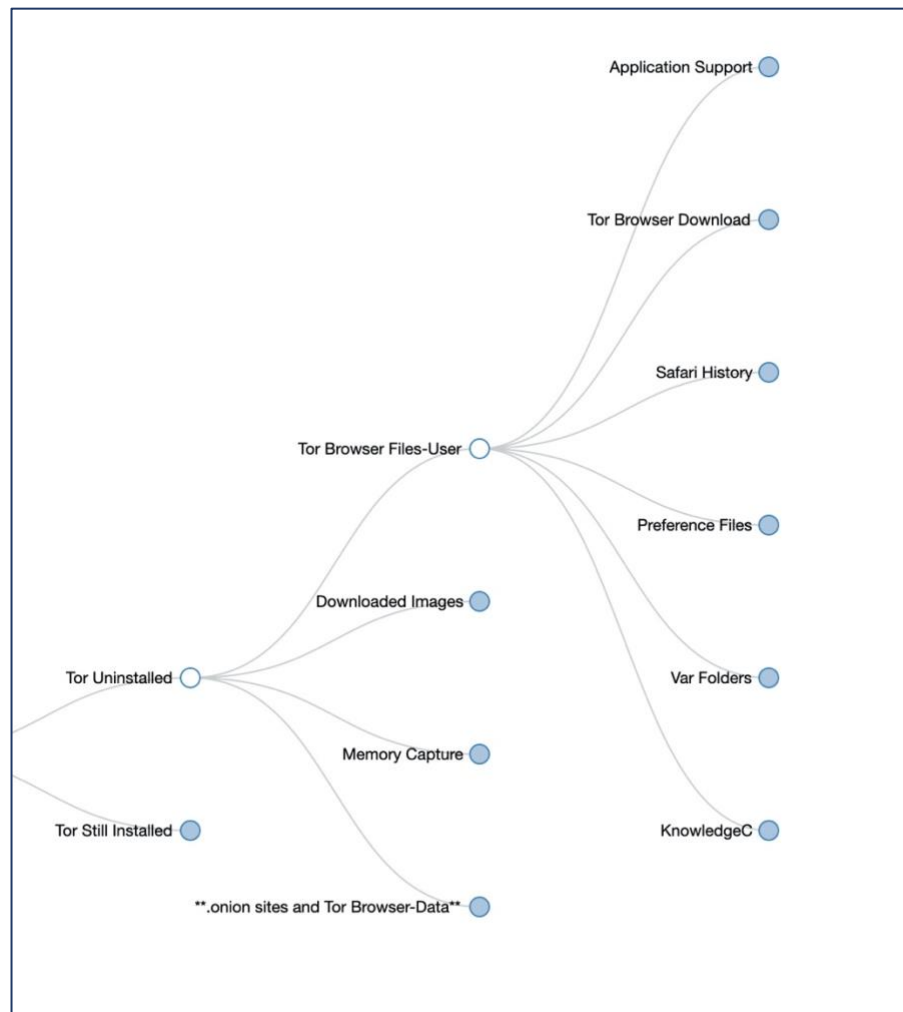


Figure 34 - macOS Local Install Tor Deleted Tor Browser Files-User

What traces of Tor remain in the Windows Registry after use?

Is it possible to find what sites were visited while using Tor?

This becomes two separate questions as it is possible to find sites that were visited using Tor, however they are not necessarily connected to the Windows Registry. The first question is:

Windows Registry Artifacts

What traces of Tor remain in the Windows Registry after use?

Two of the three methods tested for this research, both running Tor from a USB drive or SD card and installing it locally on to a Windows PC, leave traces in the Windows Registry.

Booting to Tails happens before the locally installed operating system boots, so using Tails does not leave traces in the Windows Registry.

Tor USB Install

Installing Tor onto a USB drive, then launching Tor from a Windows PC host machine leaves several artifacts in the Windows Registry. Artifacts for both Tor and the USB drive that was inserted can be identified.

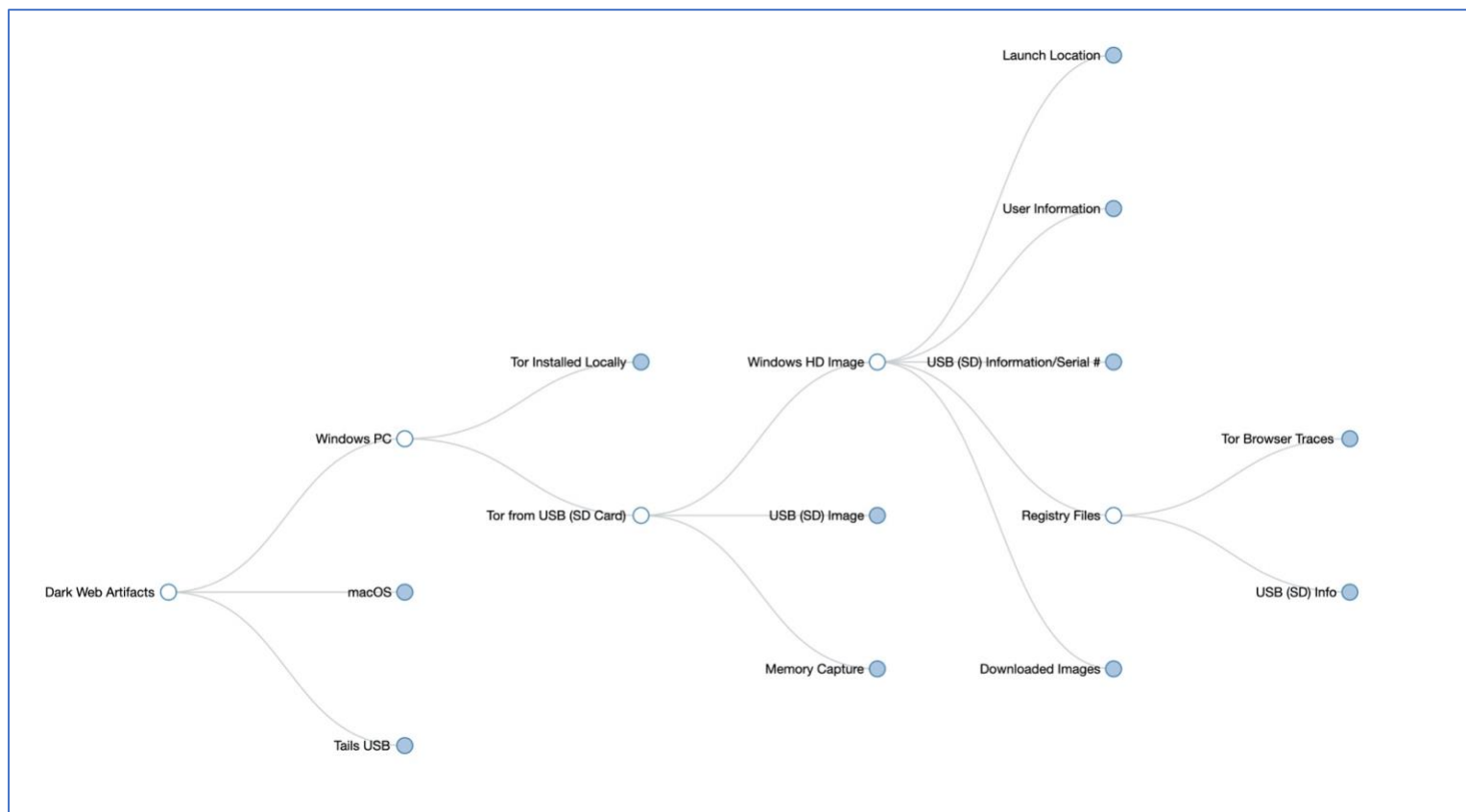


Figure 35 - Windows Tor from USB (SD Card) HD Registry

Several artifacts of the Tor Browser are found in the *NTUSER.DAT* file for the user who launched Tor in the SOFTWARE key including *PropertyStore*, *AppBadgeUpdated*, *Compatibility Assistant* and the *Firefox Launcher*.

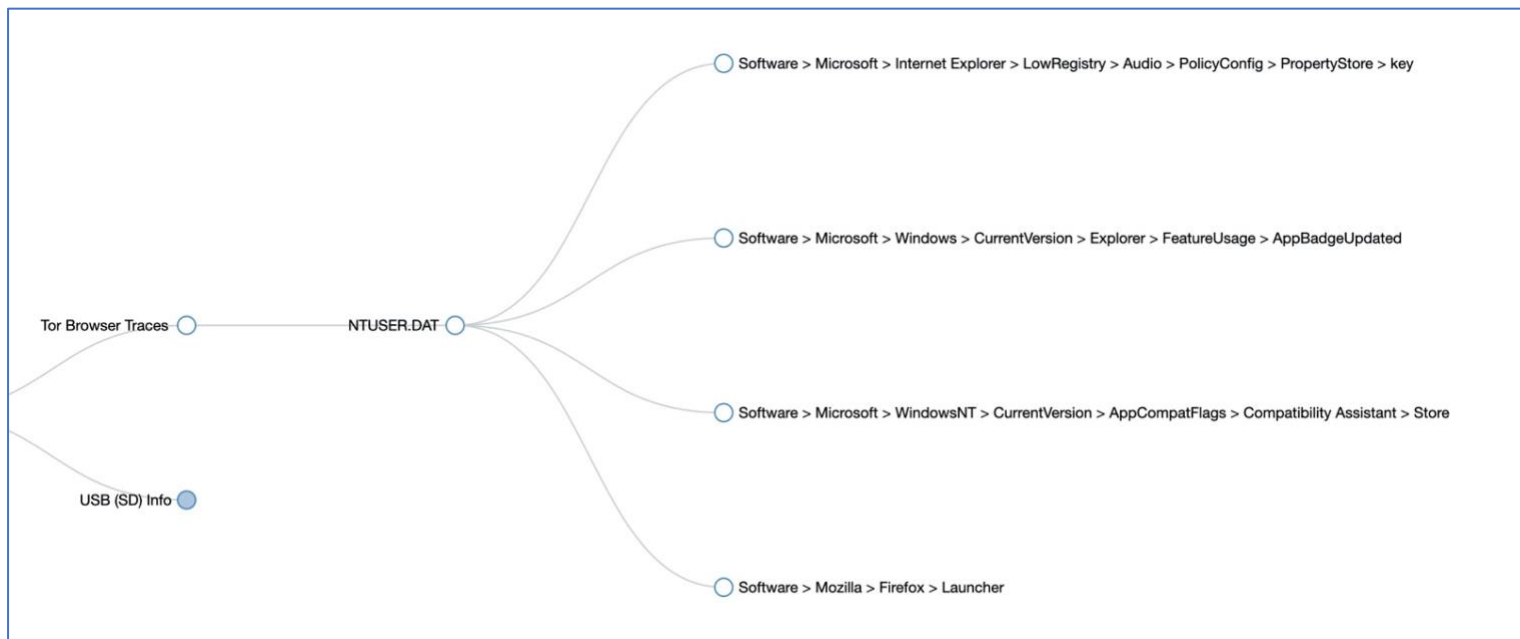


Figure 36 - Windows Tor from USB (SD Card) NTUSER.DAT

USB drive or SD card information is found in several areas of the Windows Registry including the SYSTEM key, SOFTWARE key, and registry log files. The make, model, and serial number of the USB drive can be found in these entries. On their own, the make and model of USB drive does not definitively prove that the USB drive or SD card in question was used to start Tor. However, if the USB drive or SD card is also found with the computer in question it can be definitively linked by locating the serial number on the USB drive and connecting it to the entries in the registry. Shown below are the multiple locations in the SYSTEM key.

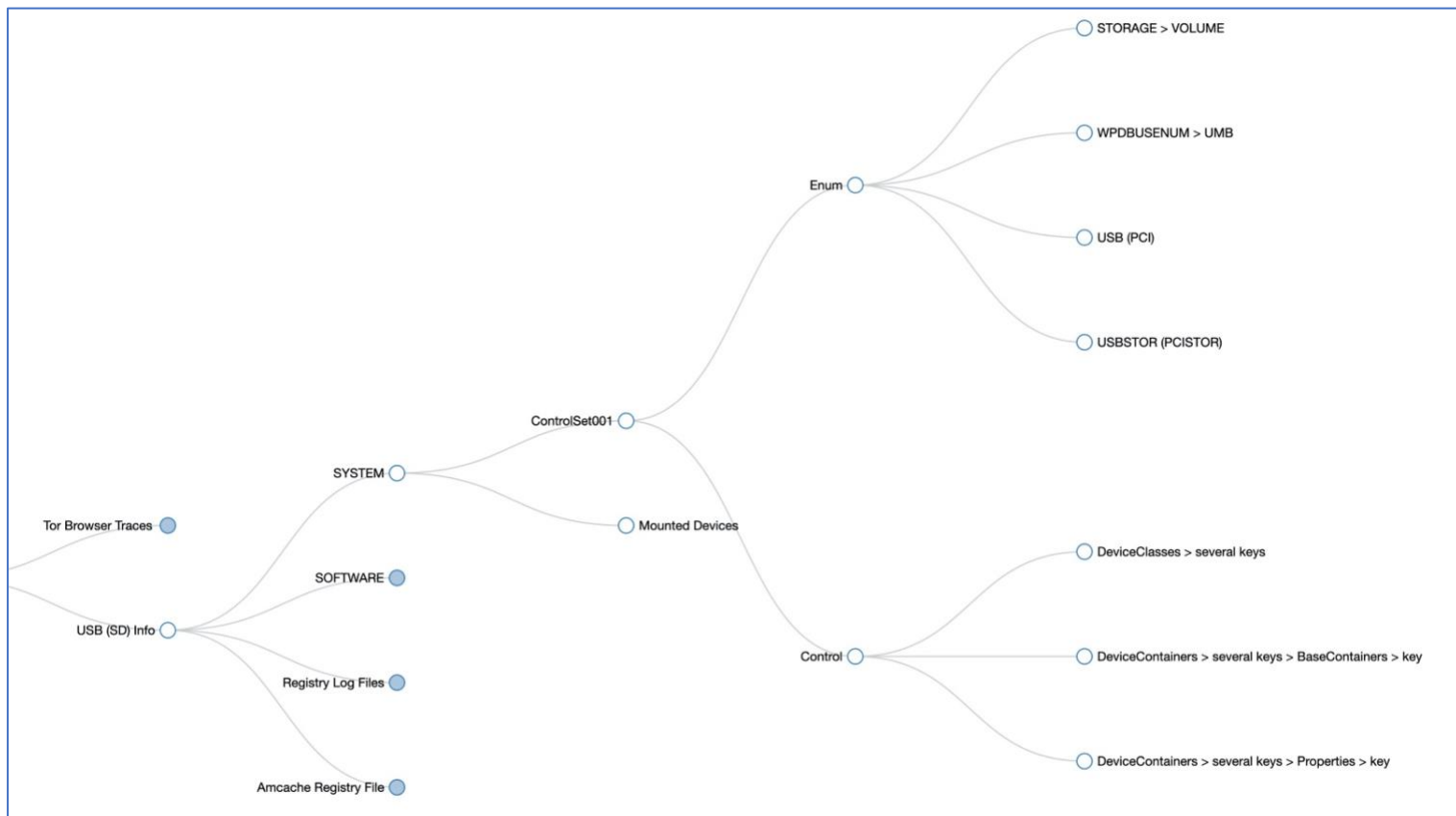


Figure 37 - Windows Tor from USB (SD Card) USB Registry SYSTEM

The SOFTWARE key, registry log files, and Amcache hive files also contain indicators of USB drive and SD card use and identification.

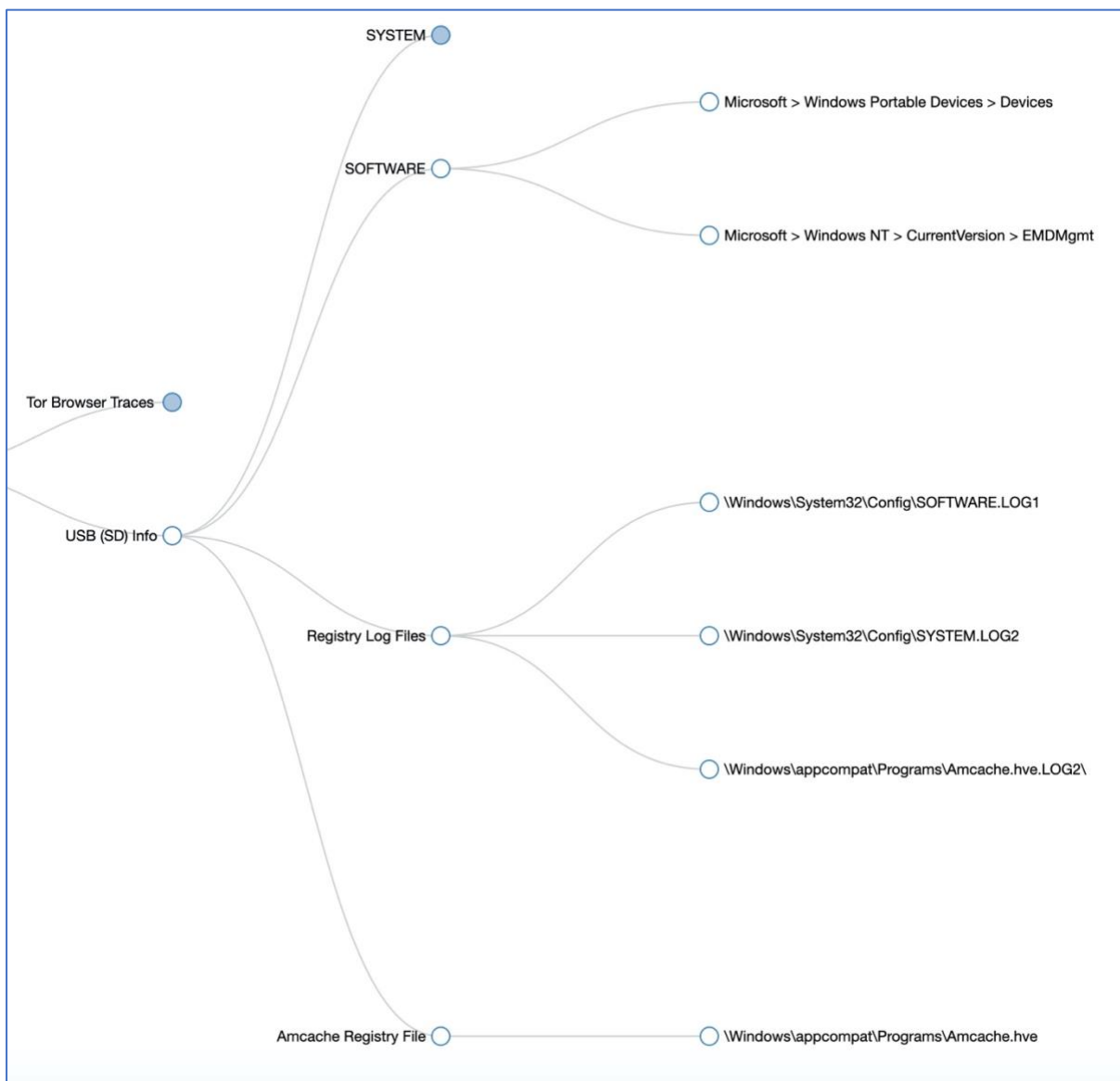


Figure 38 - Windows Tor from USB (SD Card) USB Registry Artifacts

Tor Local Install

Installing Tor locally on a Windows PC system leaves traces of Tor in both the SYSTEM registry file and the NTUSER.DAT file of the user who installed Tor. This is true both with Tor still installed and with Tor uninstalled from the system.

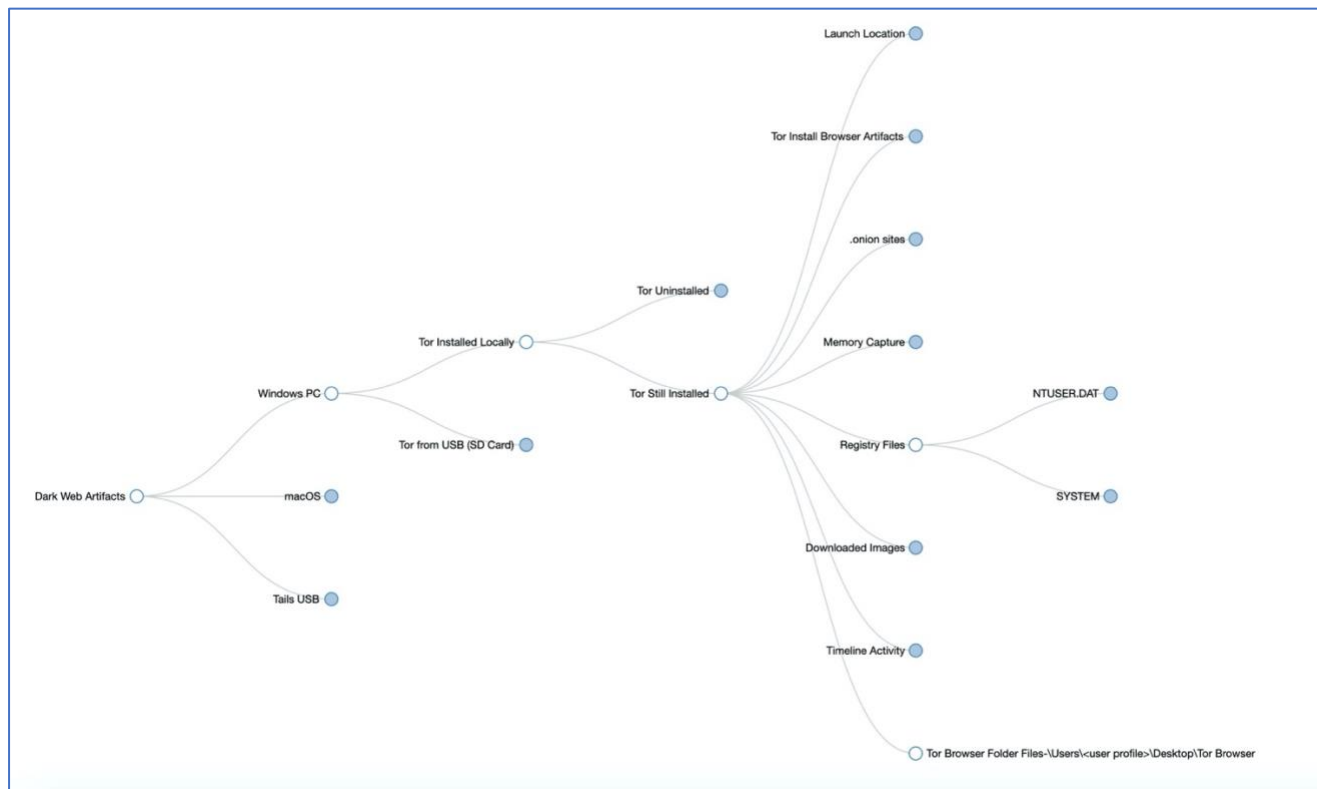


Figure 39 - Windows Tor Local Registry

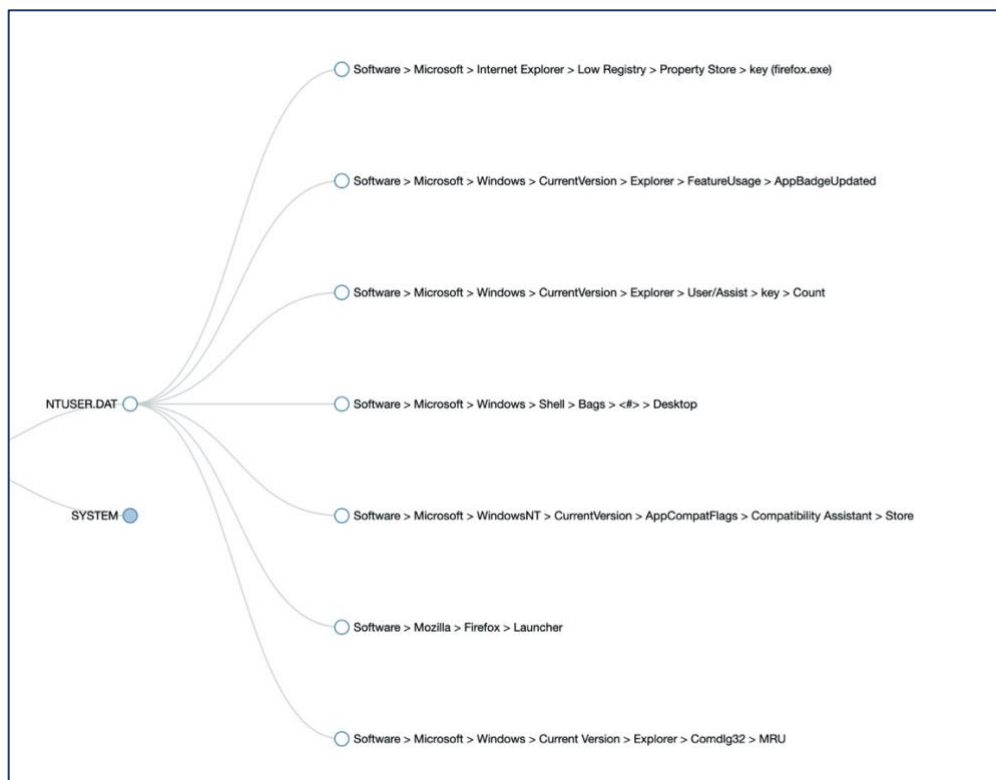


Figure 40 - Windows Tor Local Registry SOFTWARE

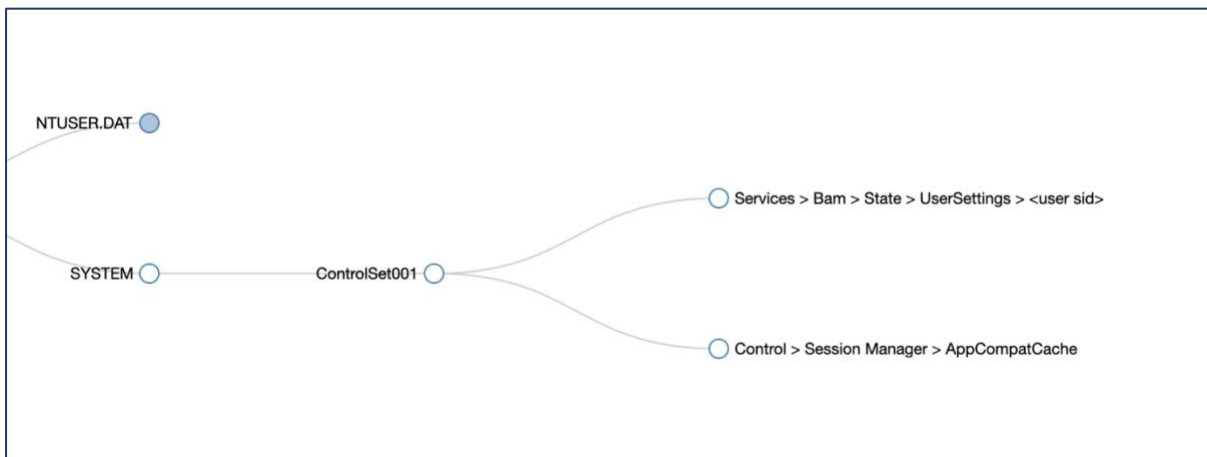


Figure 41 - Windows Tor Local Registry SYSTEM

Both still installed and uninstalled, Tor artifacts remain in multiple locations in the SOFTWARE key of the NTUSER.DAT file of the user who installed Tor.

The second question is:

Dark Web Site History

Is it possible to find what sites were visited while using Tor?

The way Tor is used is relevant when locating host-based artifacts on a system that has been used to access the dark web. As stated in the beginning of this chapter, for this research the default settings of Tor were not changed. Changing the settings can alter the security of what Tor saves. As installed, Tor allows for bookmarking websites and enabling add-ons which leave artifacts behind in the *places.sqlite* and *extensions.json* files.

Firefox SQLite Places Database

Bookmarks

Title	Parent	Created	Modified	URL
		6/7/2020 5:59:06 PM -0500	6/7/2020 7:46:34 PM -0500	
menu		6/7/2020 5:59:06 PM -0500	6/7/2020 5:59:07 PM -0500	
toolbar		6/7/2020 5:59:06 PM -0500	6/7/2020 5:59:07 PM -0500	
tags		6/7/2020 5:59:06 PM -0500	6/7/2020 5:59:06 PM -0500	
unfiled		6/7/2020 5:59:06 PM -0500	6/7/2020 7:46:34 PM -0500	
mobile		6/7/2020 5:59:06 PM -0500	6/7/2020 5:59:07 PM -0500	
	menu	6/7/2020 5:59:07 PM -0500	6/7/2020 5:59:07 PM -0500	
Learn more about Tor	toolbar	6/7/2020 5:59:07 PM -0500	6/7/2020 5:59:07 PM -0500	https://www.torproject.org/
The Tor Blog	toolbar	6/7/2020 5:59:07 PM -0500	6/7/2020 5:59:07 PM -0500	https://blog.torproject.org/
The Onion Web The Invisible Portal - Explore The Deep Web	unfiled	6/7/2020 7:38:42 PM -0500	6/7/2020 7:38:42 PM -0500	http://onionwsoi53xre32jwve7euacadvhprq2jyfttb55hrbo3execodad.onion/
Home - Onion.Live	unfiled	6/7/2020 7:44:55 PM -0500	6/7/2020 7:44:55 PM -0500	https://onion.live/
Title	Parent	Created	Modified	URL
Empire Market	unfiled	6/7/2020 7:46:34 PM -0500	6/7/2020 7:46:34 PM -0500	http://vg43c6zaobirjlm3g25v7l7qw5huxuom2b5ksdh247fz3c2uv4gsnzad.onion/

Figure 42 - places.sqlite

If the sites being bookmarked are dark web sites, they can be identified by investigators. The presence of the *places.sqlite* database file is dependent on where Tor was launched from and whether or not the user has deleted Tor from the system. Tor launched from a USB drive will leave the *places.sqlite* database file on the USB drive rather than on the hard drive of the system. If Tor has been deleted from the system, it may no longer be on the system or may show as a deleted file.

Add-ons that the user installs while using the Tor Browser will be found in the *extensions.json* file, found in the same location as the *places.sqlite* file.

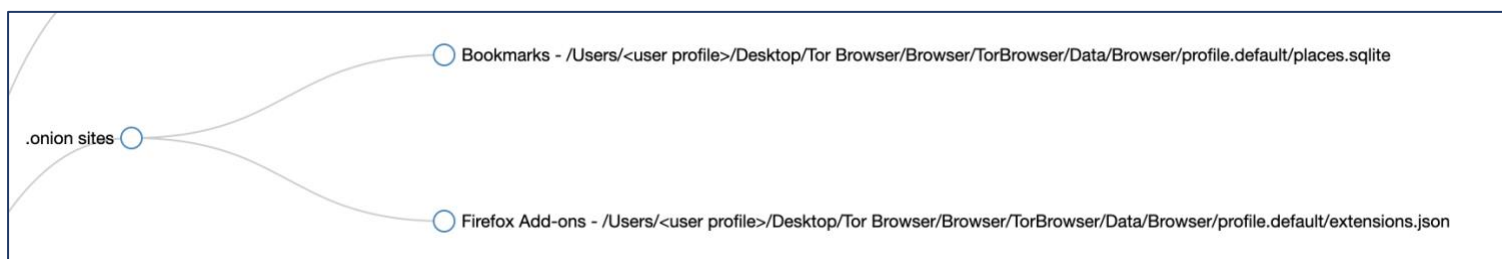


Figure 43 - Windows .onion Locations

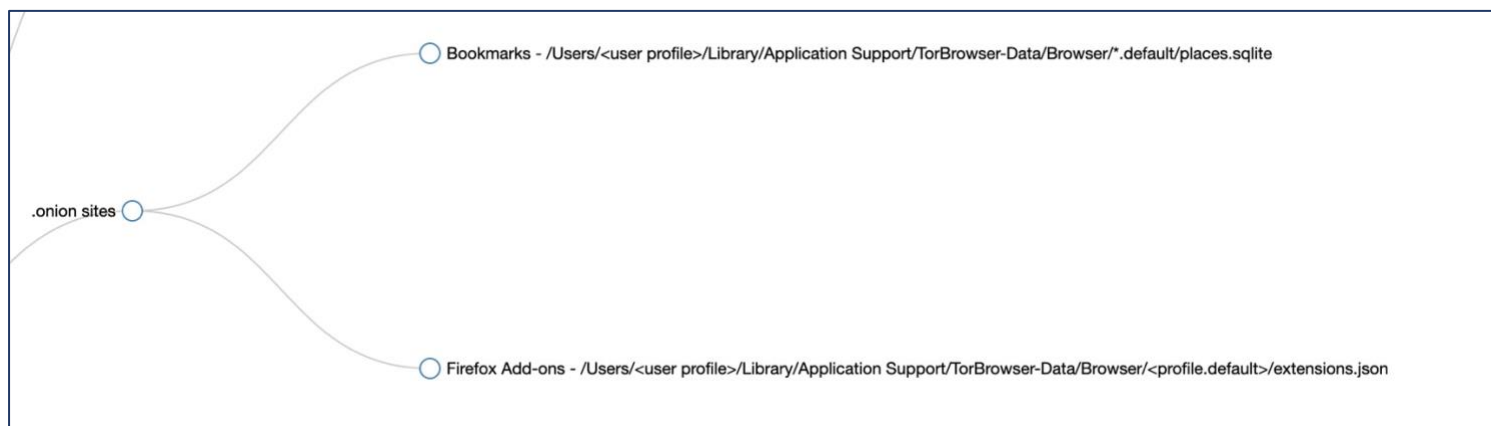


Figure 44 - macOS .onion Locations

Memory Capture

A memory capture of a system that has been accessing the dark web also can reveal traces of sites that were visited and images of both Tor and those sites. Each operating system revealed dark web artifacts in memory. The number of artifacts is dependent on whether or not the system has been shut down for a significant amount of time after the dark web site was accessed, however traces can still remain in the *pagefile.sys* of a Windows system. Artifacts

that can be found in memory include *.onion* sites, images from dark web sites, and Tor icons. The operating system and whether or not Tor was launched from a USB drive or SD card or installed locally are less important than if the system was shut down.

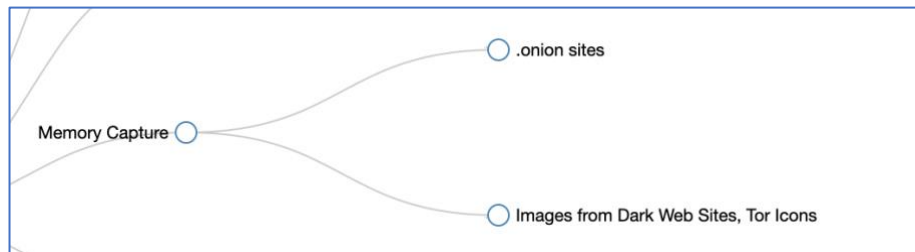


Figure 45 - Memory Capture Artifacts

Images that may be recovered include logos from dark web sites as shown below for Empire Market, Tor Browser icons, and images downloaded from dark web sites.



Figure 46 - Memory Images

Tails

If a system is booted with a Tails USB drive that supposedly leaves no trace, will any traces of having booted to a USB remain?

The Tails website states that using Tails leaves no trace on the computer (Abraham, Silva, Decourcy, & Cardon, n.d.). This was confirmed during this research on both the Windows PC and macOS systems independently. The hard drive of each system was imaged immediately prior and again immediately after booting to the Tails USB drive resulting in matching hash values for both systems.

```
Created By AccessData® FTK® Imager 4.3.0.18

Case Information:
Acquired using: ADI4.3.0.18
Case Number: Windows After Reset
Evidence Number:
Unique description: Windows PC After Resetting Windows
Examiner: Arica Kulm
Notes:
-----

Information for D:\Dissertation Research Data Files\Windows After Reset\WindowsAfterReset:

Physical Evidentiary Item (Source) Information:
[Device Info]
Source Type: Physical
[Drive Geometry]
Cylinders: 30,401
Tracks per Cylinder: 255
Sectors per Track: 63
Bytes per Sector: 512
Sector Count: 488,397,168
[Physical Drive Information]
Drive Model: Samsung SSD 860 EVO 250G USB Device
Drive Serial Number: S3YHNX0M434158P
Drive Interface Type: USB
Removable drive: False
Source data size: 238475 MB
Sector count: 488397168
[Computed Hashes]
MD5 checksum: caab2ddd8f10580ecf8a2a8714a4eb40
SHA1 checksum: bf06133e66de63bef7e7d96c733ac01314b2363a

Image Information:
Acquisition started: Mon May 25 05:57:40 2020
Acquisition finished: Mon May 25 06:18:10 2020
Segment list:
D:\Dissertation Research Data Files\Windows After Reset\WindowsAfterReset.E01

Image Verification Results:
Verification started: Mon May 25 06:18:10 2020
Verification finished: Mon May 25 06:28:14 2020
MD5 checksum: caab2ddd8f10580ecf8a2a8714a4eb40 : verified
SHA1 checksum: bf06133e66de63bef7e7d96c733ac01314b2363a : verified
```

Figure 47 - Windows PC Hard Drive Image Hash Verification Prior to Tails

```
Created By AccessData® FTK® Imager 4.3.0.18

Case Information:
Acquired using: ADI4.3.0.18
Case Number: Tails PC 3
Evidence Number:
Unique description: Image of Windows PC After Tails 3rd Time
Examiner: Arica Kulm
Notes: Following Windows Reset

-----

Information for D:\Dissertation Research Data Files\Tails_PC_PostBoot3\Tails PC Post Boot 3:

Physical Evidentiary Item (Source) Information:
[Device Info]
  Source Type: Physical
[Drive Geometry]
  Cylinders: 30,401
  Tracks per Cylinder: 255
  Sectors per Track: 63
  Bytes per Sector: 512
  Sector Count: 488,397,168
[Physical Drive Information]
  Drive Model: Samsung SSD 860 EVO 250G USB Device
  Drive Serial Number: S3YHNX0M434158P
  Drive Interface Type: USB
  Removable drive: False
  Source data size: 238475 MB
  Sector count: 488397168
[Computed Hashes]
  MD5 checksum: caab2ddd8f10580ecf8a2a8714a4eb40
  SHA1 checksum: bf06133e66de63bef7e7d96c733ac01314b2363a

Image Information:
Acquisition started: Mon May 25 07:37:45 2020
Acquisition finished: Mon May 25 07:58:11 2020
Segment list:
  D:\Dissertation Research Data Files\Tails_PC_PostBoot3\Tails PC Post Boot 3.E01

Image Verification Results:
Verification started: Mon May 25 07:58:11 2020
Verification finished: Mon May 25 08:08:09 2020
MD5 checksum: caab2ddd8f10580ecf8a2a8714a4eb40 : verified
SHA1 checksum: bf06133e66de63bef7e7d96c733ac01314b2363a : verified
```

Figure 48 - Windows PC Hard Drive Image Hash Verification After Tails


```

Created By AccessData® FTK® Imager 4.3.0.18

Case Information:
Acquired using: ADI4.3.0.18
Case Number: Apple Baseline 2
Evidence Number:
Unique description: 2nd Apple Baseline Image
Examiner: Arica Kulm
Notes:

-----

Information for D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2:

Physical Evidentiary Item (Source) Information:
[Device Info]
  Source Type: Physical
[Drive Geometry]
  Cylinders: 60,821
  Tracks per Cylinder: 255
  Sectors per Track: 63
  Bytes per Sector: 512
  Sector Count: 977,105,060
[Physical Drive Information]
  Drive Model: APPLE SSD SM0512G USB Device
  Drive Serial Number: S29ANYAG695933
  Drive Interface Type: USB
  Removable drive: False
  Source data size: 477102 MB
  Sector count: 977105060
[Computed Hashes]
  MD5 checksum: 25b90590a9a42ddcd3ebba67f0eef33
  SHA1 checksum: d7a109e06f1f6e9f039d9fae0c963f39415968ad

Image Information:
Acquisition started: Sat Jun 20 07:24:36 2020
Acquisition finished: Sat Jun 20 08:06:38 2020
Segment list:
  D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2.E01
  D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2.E02
  D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2.E03
  D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2.E04
  D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2.E05
  D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2.E06
  D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2.E07
  D:\Dissertation Research Data Files\Apple Baseline 2\AppleBaseline2.E08

Image Verification Results:
Verification started: Sat Jun 20 08:06:38 2020
Verification finished: Sat Jun 20 08:26:44 2020
MD5 checksum: 25b90590a9a42ddcd3ebba67f0eef33 : verified
SHA1 checksum: d7a109e06f1f6e9f039d9fae0c963f39415968ad : verified

```

Figure 49 - macOS Hard Drive Image Hash Verification Prior to Tails

```

Created By AccessData® FTK® Imager 4.3.0.18

Case Information:
Acquired using: ADI4.3.0.18
Case Number: Apple HD After Tails 2
Evidence Number:
Unique description: Apple Hard Drive after Tails - 2nd attempt
Examiner: Arica Kulm
Notes:
-----

Information for D:\Dissertation Research Data Files\Apple After Tails 2\ApplePostTails2:

Physical Evidentiary Item (Source) Information:
[Device Info]
  Source Type: Physical
[Drive Geometry]
  Cylinders: 60,821
  Tracks per Cylinder: 255
  Sectors per Track: 63
  Bytes per Sector: 512
  Sector Count: 977,105,060
[Physical Drive Information]
  Drive Model: APPLE SSD SM0512G USB Device
  Drive Serial Number: S29ANYAG695933
  Drive Interface Type: USB
  Removable drive: False
  Source data size: 477102 MB
  Sector count: 977105060
[Computed Hashes]
  MD5 checksum: 25b90590a9a42ddcd3ebba67f0eef33
  SHA1 checksum: d7a109e06f1f6e9f039d9fae0c963f39415968ad

Image Information:
Acquisition started: Sun Jun 21 06:11:15 2020
Acquisition finished: Sun Jun 21 06:53:38 2020
Segment list:
  D:\Dissertation Research Data Files\Apple After Tails 2\ApplePostTails2.E01

Image Verification Results:
Verification started: Sun Jun 21 06:53:38 2020
Verification finished: Sun Jun 21 07:13:56 2020
MD5 checksum: 25b90590a9a42ddcd3ebba67f0eef33 : verified
SHA1 checksum: d7a109e06f1f6e9f039d9fae0c963f39415968ad : verified

```

Figure 50 - macOS Hard Drive Image Hash Verification After Tails

Tails contains an option to create persistent storage if it is started from a USB drive. The information stored on persistent storage can include documents, email, chat sessions, and other software information. Tails warns about the security implications of creating persistent storage with the persistent storage being protected by a passphrase. As with other passwords or passphrases, the simpler the password the easier it is to compromise. When imaging a USB

drive that has an installation of Tails with persistent storage enabled, it may not be obvious to investigators that the drive potentially contains a significant amount of evidence. Creating an image of the Tails USB drive using FTK Imager produces an image with both a Tails partition and a TailsData partition (see below).

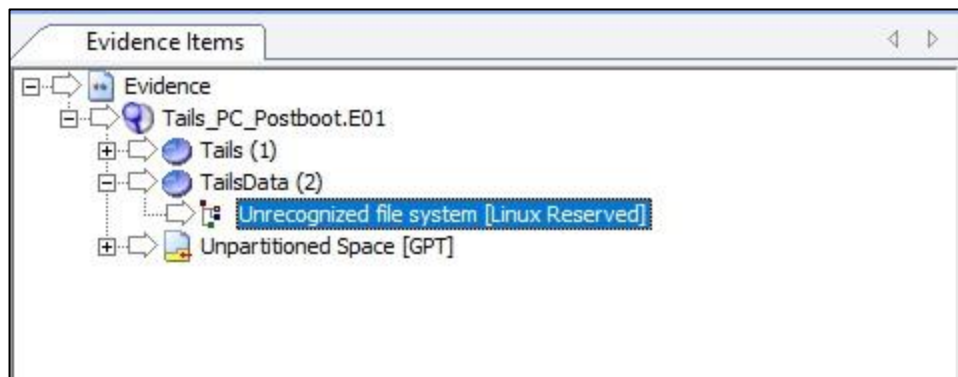


Figure 51 - TailsData Unrecognized File System

The TailsData partition is seen as an Unrecognized file system in FTK Imager requiring an examiner to follow additional steps to read the data that has been stored in persistent storage. Tails persistent storage is encrypted using LUKS encryption as can be seen below in the Hex view of the Unrecognized file system.

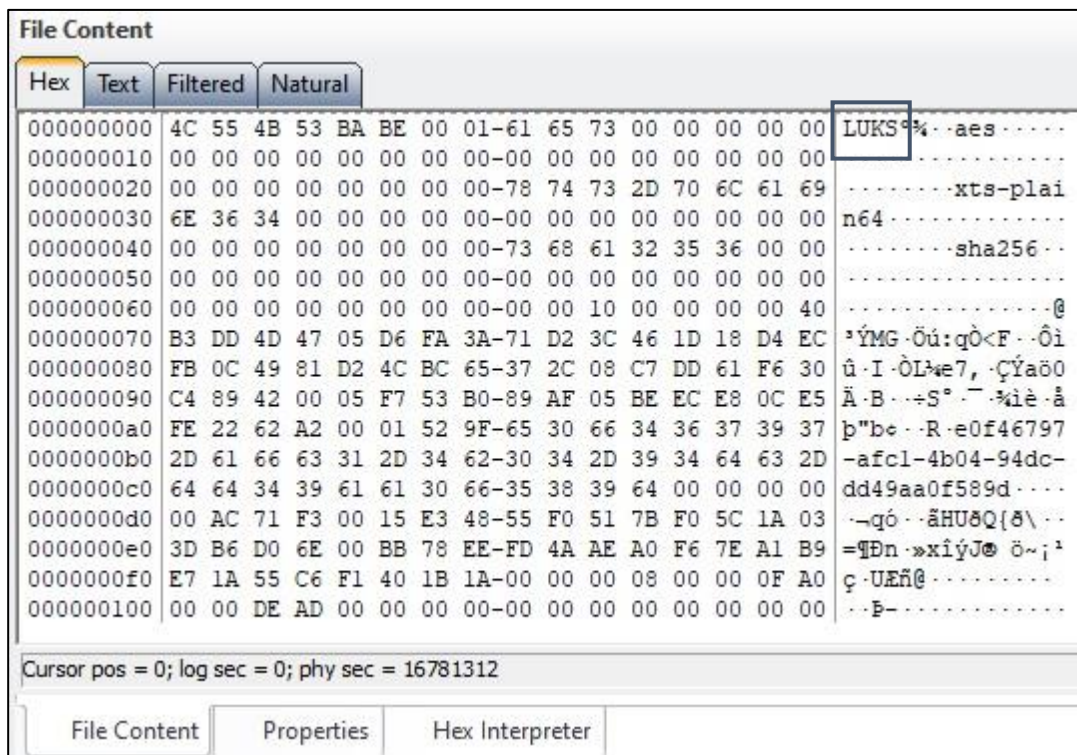


Figure 52 - LUKS Encryption Indicated

Being able to read the data stored within the persistent storage partition is wholly dependent on the examiner either knowing the user's passcode or being able to crack the passcode with a tool such as Hashcat.

A Tails USB drive from a user that sets a complicated passphrase is unlikely to provide any useful evidence to an investigator. With a known or simple passphrase, an investigator has several steps to go through to export, decrypt, and perform further steps to eventually be able to examine the data and information contained within the persistent storage. The created Dark Web Artifact Framework provides steps and links to further information to walk an examiner through these required steps in order to reveal the hidden information.

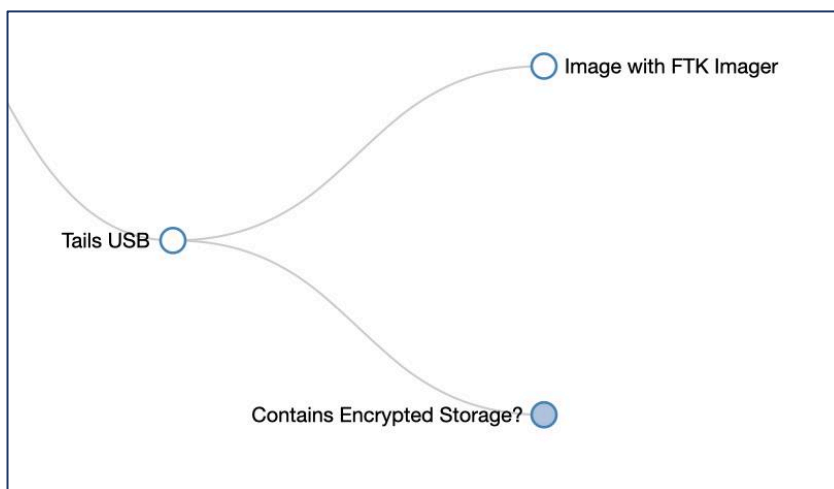


Figure 53 - Tails Initial Branching

Upon acquiring the Tails USB, the examiner first images the drive using FTK Imager or similar imaging tool. The question is then "Does the drive contain encrypted storage?".

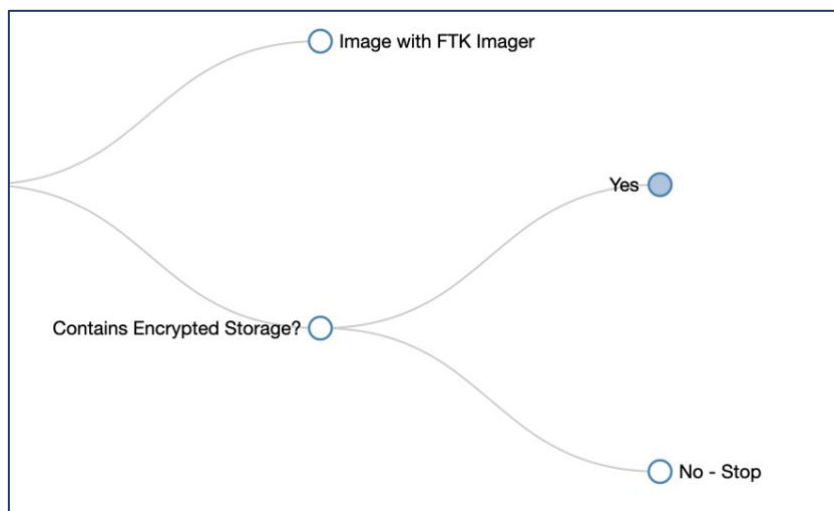


Figure 54 - Tails Encrypted Storage

A user who has not enabled encrypted storage will leave no evidence on the USB drive so the investigation of the drive would stop. The presence of encrypted storage would then move the investigator forward to the next step of exporting the unrecognized partition as a raw .001 image file using FTK Imager.

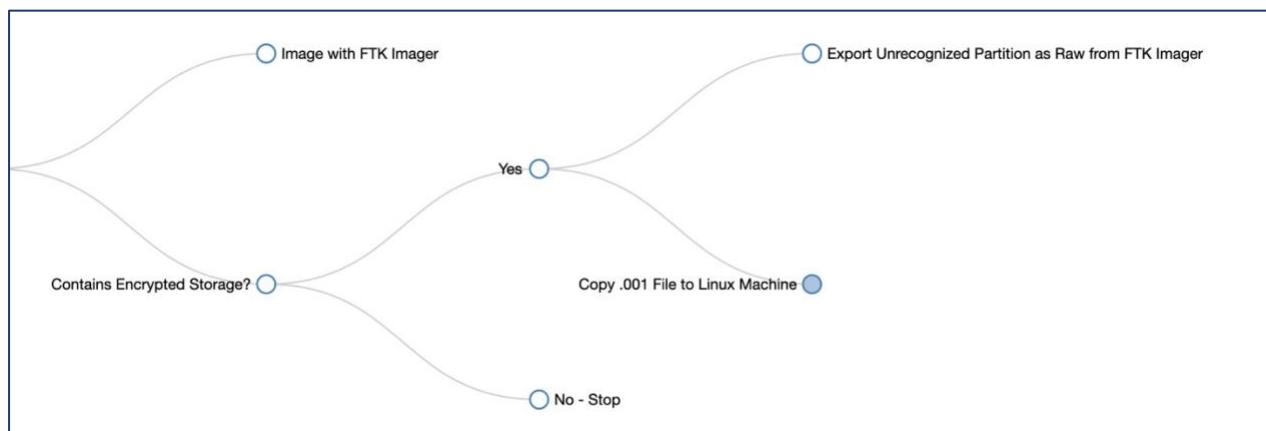


Figure 55 - Tails Export Unrecognized Partition

Once the image is exported, it needs to be decrypted before it can be examined. Tails uses Linux Unified Key System (LUKS) encryption to encrypt the persistent storage partition (“Tails - Privacy for anyone anywhere,” n.d.).

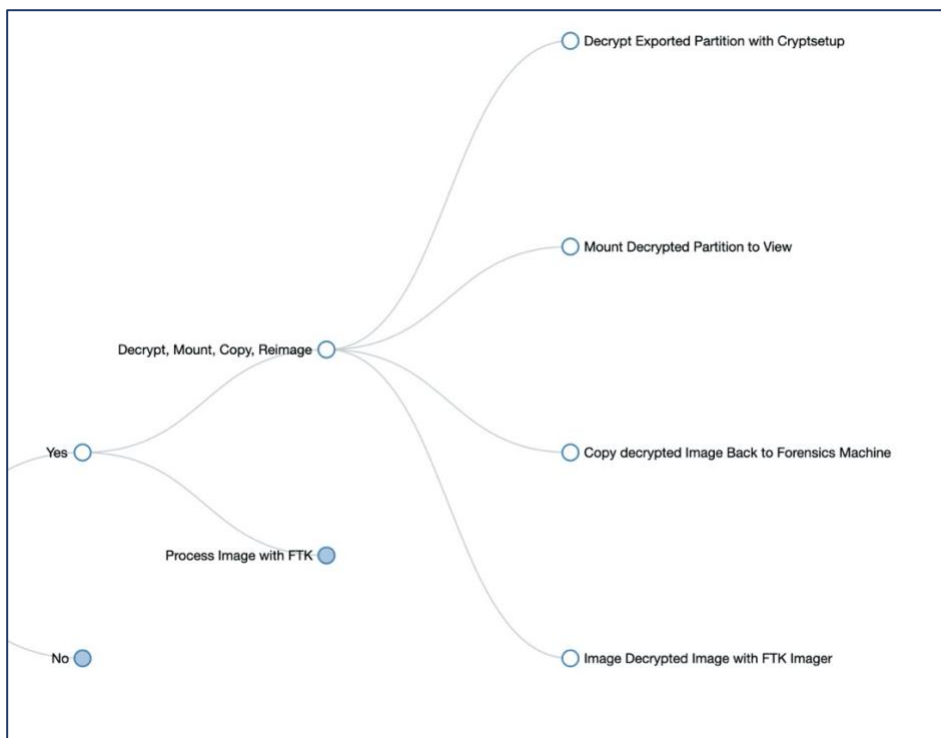


Figure 56 - Tails Decrypt, Mount, Copy, Reimage

A tool that decrypts LUKS partitions is the cryptsetup utility for Linux. The .001 raw image file is then copied to a Linux machine to perform the decryption. If the password is known the decryption steps can move forward. If the password is not known a utility such as Hashcat can be used to attempt to obtain the password (“hashcat - advanced password recovery,” n.d.). Once decrypted, the image can be mounted for viewing in Linux. To perform a full forensic exam the decrypted image is copied back to the original forensics workstation and processed with FTK. Artifacts that are recoverable include bookmarks in the *places.sqlite* file, which will show sites that the user has saved including dark web sites. Nm-system-connections shows Wi-Fi connections that the Tails drive has connected to and includes the Wi-Fi password. Other artifacts include Pidgin log files which contain internet chat sessions, thunderbird email, and stored passwords.

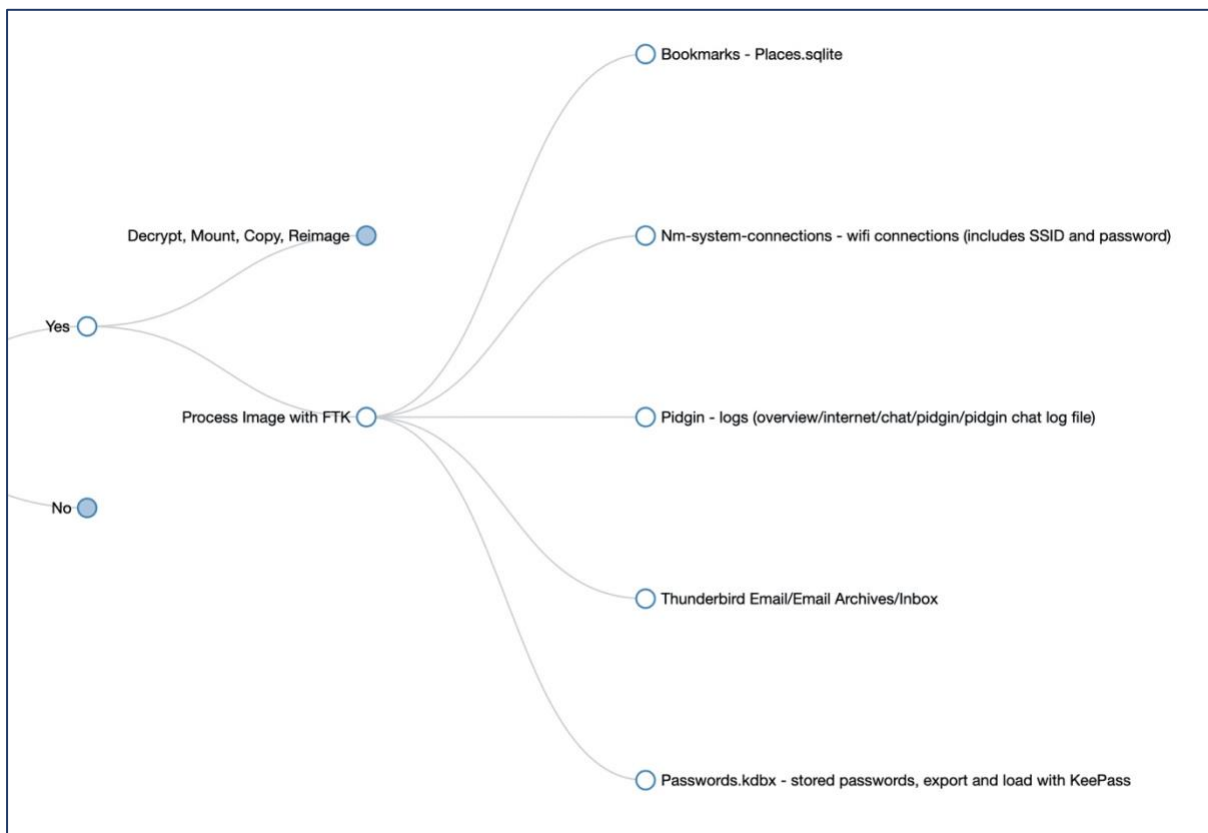


Figure 57 - Tails Process Decrypted Image with FTK

Host-based Artifacts

The primary question of this research to be used in creating the framework is “What host-based artifacts can be identified on a system running either Windows PC or macOS when the user has been accessing the dark web using Tor or Tails?”.

This question has largely been answered by the answers to the previous questions. Both Windows and macOS systems have numerous artifacts left behind when using Tor to access the dark web whether launched from a USB, run locally, or run locally and deleted. As stated earlier, the purpose of this research was not to compare the number or type of artifacts left behind on each system, but rather to provide a framework for identifying these artifacts on each system. The location of Tor, how it was launched, when it was downloaded and installed, the user who downloaded and installed Tor, and many other artifacts remain. Proving the user of Tor was accessing the dark web rather than using Tor for another purpose narrows the artifacts to the *places.sqlite* file, *extensions.json*, and system memory. The memory artifacts can be dependent on whether or not the system has been powered down, whether or not a *pagefile.sys* on Windows or swapfile on macOS is configured.

Accessing the dark web using a Tails drive leaves no host-based artifacts on the computer that launched Tails, however if the Tails drive is found, the persistent storage container may reveal many dark web artifacts.

Conclusion

This chapter discusses the creation of the framework using the questions posed in the methodology described in Chapter 3. Users may install Tor in different ways to try to remain anonymous when accessing the dark web. Tor can be installed locally, installed on a USB drive or SD card, or run from within a bootable operating system such as Tails. Depending on the sophistication of the user, artifacts can be left behind in each case on both Windows and macOS. Further efforts to hide these artifacts include the user deleting Tor after it has been installed locally which also leaves behind many artifacts. Traces in the Windows registry can be found whether installed locally or run from a USB drive. Running Tor from within a bootable operating system like Tails leaves no trace on the user’s hard drive; however, if the drive is

found it may be possible to locate artifacts if the user has enabled persistent storage. Traces of dark web sites can be found in both the *places.sqlite* file and system memory.

The Dark Web Artifact Framework contains branching for each of these options along with links to more information on the particular artifact. Though the framework is detailed and comprehensive, the artifacts listed in the Dark Web Artifact Framework may not always be found on a suspect's system and there may be additional artifacts that are not included in this framework.

CHAPTER 5 – TREATMENT EVALUATION

This chapter discusses the steps taken to evaluate the created Dark Web Artifact Framework. Treatment evaluation in design science research studies the designed artifact to determine if the effects of the artifact meet the requirements specified by the researcher and ultimately contribute to the goal of the stakeholders. Three separate evaluation steps were undertaken to accomplish the evaluation of the created framework.

First, experimental evaluation was conducted using a simulation of the proposed framework. A volunteer user was asked to generate dark web history on both a macOS laptop and a Windows laptop by launching Tor locally on one machine, launching Tor from a USB drive on the alternate machine, and launching Tails on either machine to create persistent storage.

Second, the image file from a previously worked dark web case was used with the framework and compared to the original investigation. This particular case involved a suspect who had launched Tor from a SD card and accessed the dark web. A comparison was made in the type and overall quantity of artifacts that were discovered as well as connecting the SD card to the computer to further show the user's intent.

Third, the expert opinion of a member of the South Dakota Internet Crimes Against Children taskforce (ICAC) and the Division of Criminal Investigation (DCI) was sought. DCI and ICAC investigators analyze numerous systems and have expert knowledge of the evidence needed to charge and ultimately aid in convicting criminal suspects.

The goal of these evaluations is to ensure that the requirements of this researcher were met in the design of the created framework. The requirements as stated in Chapter 3 are as follows:

- Easy to follow – the framework must be easy for investigators to follow so as to not inhibit their investigation by requiring more effort than what is gained in useful evidence.
- Consider both Windows and macOS operating systems – it is possible for investigators to receive multiple pieces of digital evidence when investigating a

crime. Windows and macOS are the two most common desktop operating systems, therefore both need to be explored (Net Marketshare, n.d.).

- Consider multiple ways to access the dark web using Tor. Tor can be installed on a hard drive, run from a flash drive, or run as part of the bootable TAILS operating system. All three methods of access need to be explored.
- Be adaptable to future platforms.

The three evaluation steps outlined above when combined meet the stated requirements. The steps performed in this evaluation and the steps that were met are outlined below.

Simulation of Proposed Framework

The Dark Web Artifact Framework meets all four of the requirements set forth by this researcher. First, both a macOS laptop and Windows PC laptop were reset to factory settings to remove any personal files and a clean operating system installed. Next, for each operating system, two new, unused USB drives were created with Tor Browser installed. Last, a new, unused USB drive was used to create a Tails bootable operating system. The two laptops, two Tor USB drives, and Tails USB drive were given to a volunteer user to generate dark web data. The volunteer user was given the instructions to choose one machine to launch Tor from the provided USB drive, choose the other machine to download and install the Tor Browser, and to boot to Tails creating persistent storage. The investigator had no knowledge of which machine was used for each purpose and was left to determine the details during the investigation using the created Dark Web Artifact Framework.

Following the Dark Web Artifact Framework, the investigator was able to quickly and easily determine, based on the artifacts located, that the volunteer user had launched Tor from the USB drive on the macOS laptop and installed Tor Browser on the Windows laptop. Not all artifacts indicated in the framework were located during the experimental investigation, however new artifacts were discovered that were added to the framework.

Windows PC - Tor Installed Locally

On the Windows PC, indicators of Tor having been launched locally rather than from a USB include the Windows LNK files and prefetch files indicating where Tor was launched from, Tor Browser install executable file located in the downloads folder, *places.sqlite* file

located on the local machine, and Windows registry entries. The section of the framework shown below indicates the artifacts that were located on the Windows hard drive.

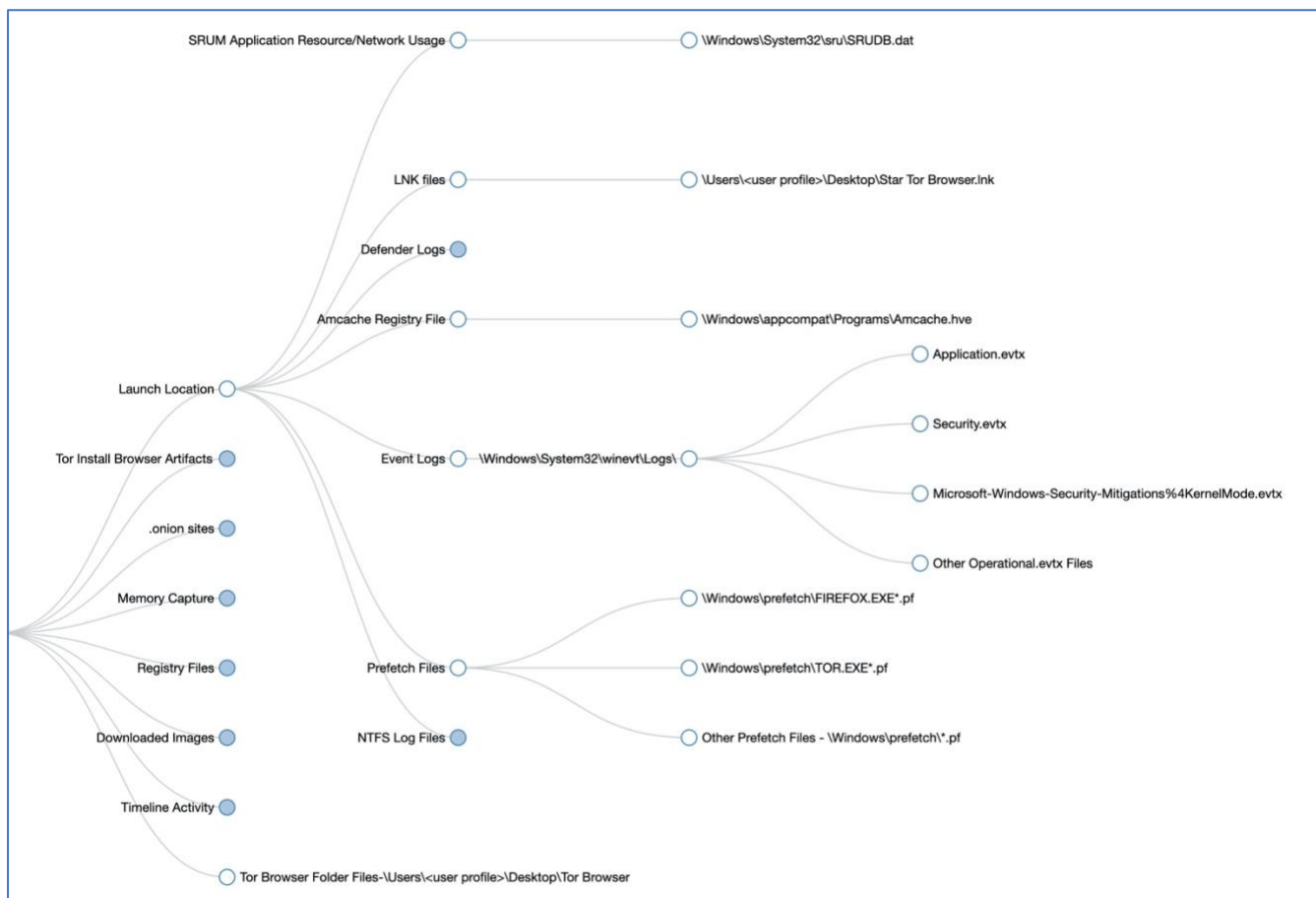


Figure 58 - Windows Hard Drive Verification Artifacts

The artifact details shown below are from for the *Start Tor Browser.LNK* file located using Magnet AXIOM Examine forensic software indicating a path of *C:\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe*. This artifact indicates that the Tor Browser was launched from the Desktop of the Admin user which is the default location of the Tor Browser when installed to a local machine.

DETAILS	
ARTIFACT INFORMATION	
Linked Path	C:\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe
Created Date/Time	9/18/2020 6:10:18 PM
Last Modified Date/Time	9/18/2020 6:10:18 PM
Accessed Date/Time	9/18/2020 6:10:18 PM
Target File Created Date/Time	1/1/2000 12:00:00 AM
Target File Last Modified Date/Time	1/1/2000 12:00:00 AM
Target File Last Accessed Date/Time	9/18/2020 6:10:03 PM
Target Attributes	FILE_ATTRIBUTE_ARCHIVE
Drive Type	DRIVE_FIXED
Volume Serial Number	AA2CB9EC
Show Command	SW_SHOWNORMAL
Net Bios Name	desktop-m5igjes
MAC Address	60:57:18:BA:87:67
Target File Size (Bytes)	1532928
EVIDENCE INFORMATION	
Source	WinValidation.E01 - Partition 4 (Microsoft NTFS, 231.71 GB)\Users\Admin\Desktop\Start Tor Browser.lnk
Recovery Method	Parsing
Deleted source	
Location	n/a
Evidence number	WinValidation.E01

Figure 59 - Start Tor Browser.LNK file

The artifact details for the Windows prefetch files also indicating the Tor executable found in the path `\Users\Admin\Desktop\Tor Browser\Browser\TorBrowser\Tor\Tor.exe`.

DETAILS	
ARTIFACT INFORMATION	
Application Name	TOR.EXE
Application Path	\VOLUME{01d5629b2c807b39-aa2cb9ec}\USERS\ADMIN\DESKTOP\TOR BROWSER\BROWSER\TORBROWSER\TOR\TOR.EXE
Application Run Count	0
File Created Date/Time	9/18/2020 6:10:29 PM
Last Run Date/Time	9/18/2020 6:10:19 PM
File Hash	C3E9E1E6
Volume Name	\VOLUME{01d5629b2c807b39-aa2cb9ec}
Volume Created Date/Time	9/3/2019 9:04:28 PM
EVIDENCE INFORMATION	
Source	WinValidation.E01 - Partition 4 (Microsoft NTFS, 231.71 GB)\Windows\prefetch\TOR.EXE-C3E9E1E6.pf
Recovery Method	Parsing
Deleted source	
Location	n/a
Evidence number	WinValidation.E01

Figure 60 - TOR.EXE Prefetch File

The Tor Browser download file was located in the `\Users\Admin\AppData\Local\` folder indicating that the Tor Browser install file was downloaded by the Admin user.

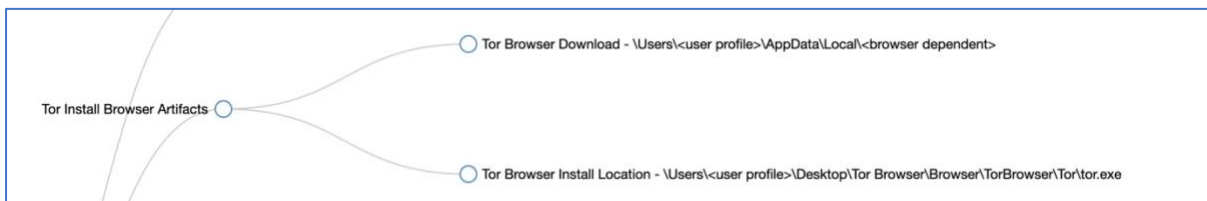


Figure 61 - Tor Install Artifacts

DETAILS

ARTIFACT INFORMATION

Download Source: https://www.torproject.org/dist/torbrowser/9.5.4/torbrowser-install-win64-9.5.4_en-US.exe

File Name: torbrowser-install-win64-9.5.4_en-US.exe

Start Time Date/Time: 9/18/2020 6:09:38 PM

End Time Date/Time: 9/18/2020 6:09:54 PM

Saved To: C:\Users\Admin\Downloads\torbrowser-install-win64-9.5.4_en-US.exe

State: Download Complete

Opened By User: Yes

Bytes Downloaded: 67065744

File Size (Bytes): 67065744

EVIDENCE INFORMATION

Source: WinValidation.E01 - Partition 4 (Microsoft NTFS, 231.71 GB)\Users\Admin\AppData\Local\Microsoft\Edge\User Data\Default\History

Recovery Method: Parsing

Deleted source:

Location: Table: downloads(id: 1)
Table: downloads_url_chains(rowid: 1)

Evidence number: WinValidation.E01

Figure 62 - Tor Browser Install Executable

The *places.sqlite* file was located in the `\Users\Admin\Desktop\Tor Browser\Browser\TorBrowser\Data\Browser\profile.default\` folder as directed by the framework. This also indicates that Tor Browser was installed locally on the system rather than launched from a USB drive or SD card.

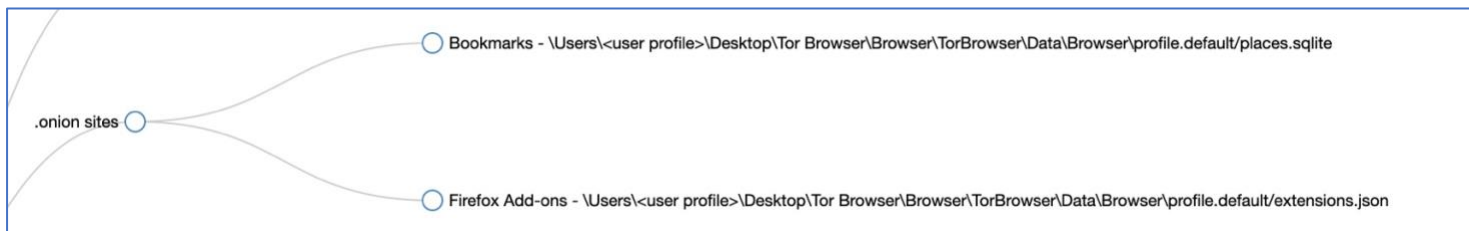


Figure 63 - .onion Artifacts

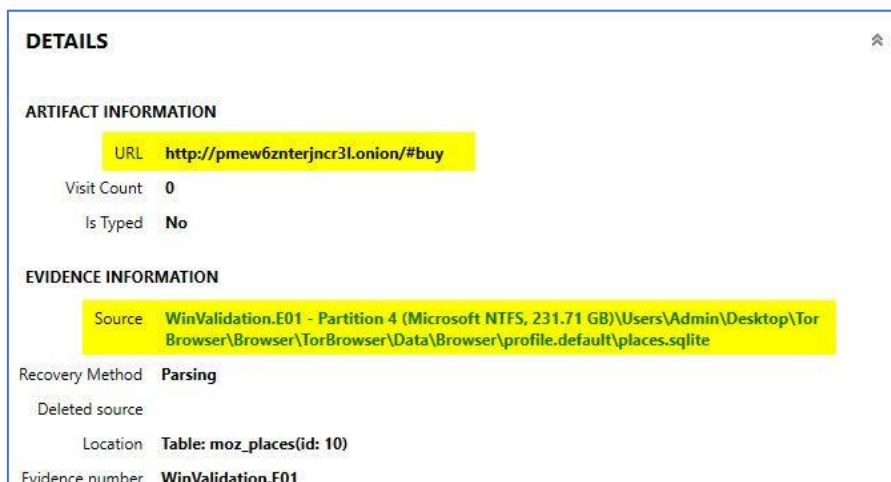


Figure 64 - places.sqlite Artifact

The Windows registry also contains indicators that Tor was installed locally. *AppBadgeUpdated*, *UserAssist*, *Shellbags*, *Compatibility Assistant*, *Mozilla Firefox Launcher*, *Bam State*, and *AppCompatCache* (also known as ShimCache) all contain traces of the Tor Browser.

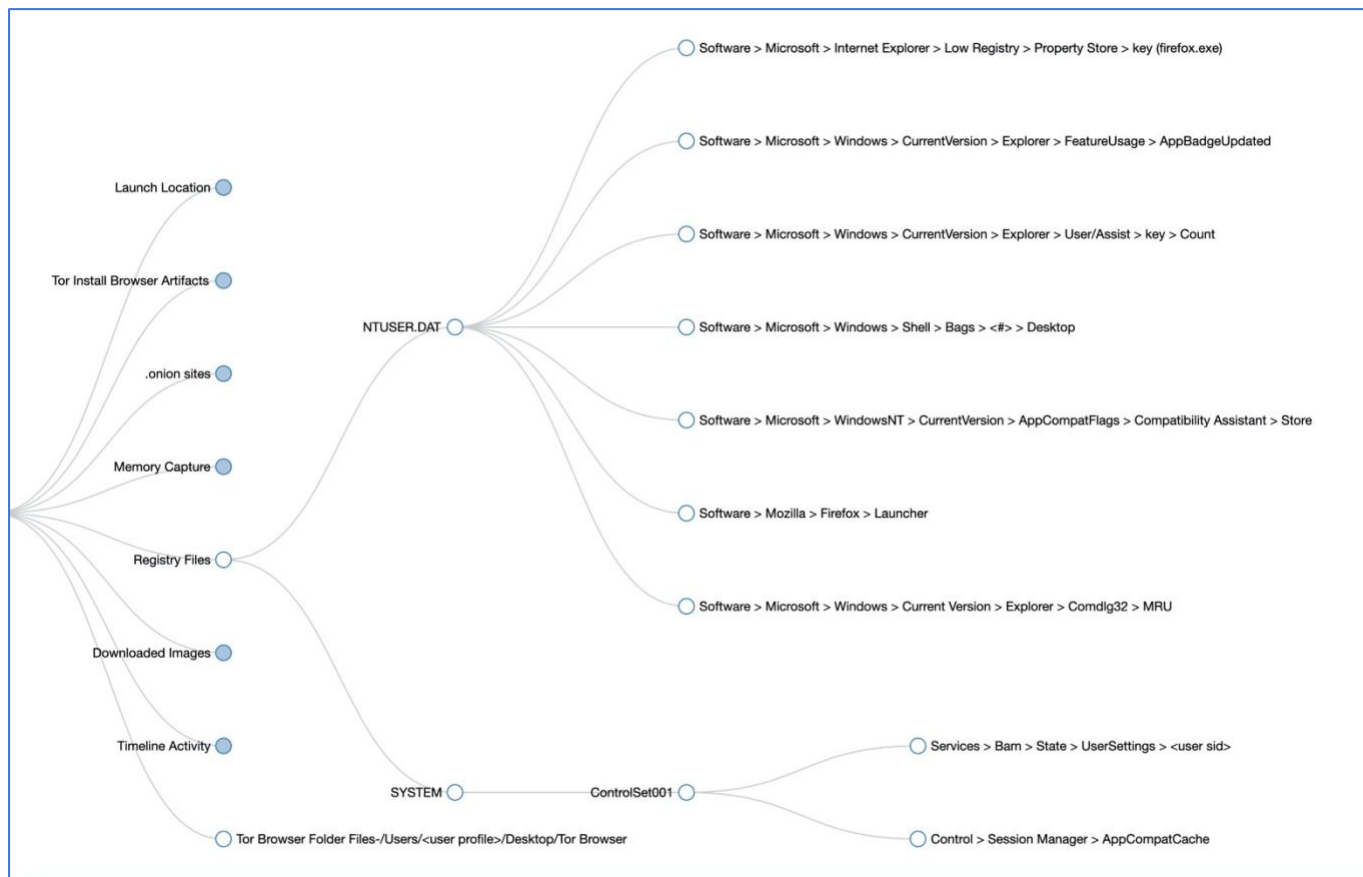


Figure 65 - Registry Artifacts

HEX

Source **IconLayouts**
Current offset **0**

GO TO FIND HIDE DECODING

162	6E	00	6B	00	00	00	0C	00	00	n.k.....
171	00	00	00	00	00	54	00	6F	00T.o.
180	72	00	20	00	42	00	72	00	6F	r. .B.r.o.
189	00	77	00	73	00	65	00	72	00	.w.s.e.r.
198	00	00	16	00	00	00	00	00	00
207	00	53	00	74	00	61	00	72	00	.S.t.a.r.
216	74	00	20	00	54	00	6F	00	72	t. .T.o.r
225	00	20	00	42	00	72	00	6F	00	. .B.r.o.
234	77	00	73	00	65	00	72	00	2E	.w.s.e.r.
243	00	6C	00	6E	00	6B	00	00	00	.l.n.k...
252	01	00	00	00	00	00	00	02	00
261	00	01	00	00	00	00	00	00	00
270	00	00	01	00	00	00	00	00	00
279	00	02	00	01	00	00	00	00	00
288	00	00	00	00	12	00	00	00	07
297	00	00	00	01	00	00	00	04	00
306	00	00	00	00	00	00	00	00	00
315	00	00	00	00	00	00	00	00	00
324	00	00	00	00	80	8F	01	00	00

Figure 69 - Tor Browser in Shellbags

NTUSER.DAT > SOFTWARE > Microsoft > Windows NT > CurrentVersion > AppCompatFlags > Compatibility Assistant > Store

Name	Type	Data
C:\Program Files\ElanTech\ETDCtrlHelper.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00 00 00
C:\Users\Admin\AppData\Local\Microsoft\OneDrive\19.002.0107.0005_1\FileSyncConfig.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00 00 00
C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00 00 00
C:\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00 00 00
C:\Users\Admin\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00 00 00
C:\Users\Admin\AppData\Local\Microsoft\OneDrive\20.143.0716.0003\FileSyncConfig.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00 00 00
E:\RamCatcher\ram64\RamCapture64.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00 00 00

Figure 70 - Compatibility Assistant Artifact

ALL EVIDENCE > WinValidation.E01 > User hives > Admin > NTUSER.DAT > SOFTWARE > Mozilla > Firefox > Launcher

Name	Type	Data
C:\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe\image	REG_DWORD	0x00000000 (0)
C:\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe\launcher	REG_QWORD	0x000000000b18f28c (186184332)
C:\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe\browser	REG_QWORD	0x000000000b1d4cc4 (186469572)
C:\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe\telemetry	REG_DWORD	0x00000000 (0)

Figure 71 - NTUSER.DAT Firefox Launcher Artifact

ControlSet001 > Services > bam > State > UserSettings > S-1-5-21-3922282893-2549882947-1794595462-1001

Name	Type	Data
Version	REG_DWORD	0x00000001 (1)
\Device\HarddiskVolume4\Windows\explorer.exe	REG_BINARY	E0 01 62 32 6D 8E D6 01 00 00 00...
SequenceNumber	REG_DWORD	0x00000008 (8)
\Device\HarddiskVolume4\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	REG_BINARY	B0 B3 32 80 E7 8D D6 01 00 00 00...
Microsoft.Windows.StartMenuExperienceHost_cw5n1h2txyewy	REG_BINARY	36 D9 78 32 6D 8E D6 01 00 00 00...
Microsoft.Windows.Cortana_cw5n1h2txyewy	REG_BINARY	36 D9 78 32 6D 8E D6 01 00 00 00...
Microsoft.LockApp_cw5n1h2txyewy	REG_BINARY	BF 76 76 32 6D 8E D6 01 00 00 00...
Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy	REG_BINARY	AB 0E CC 68 E8 8D D6 01 00 00 00...
\Device\HarddiskVolume4\Users\Admin\Downloads\torbrowser-install-win64-9.5.4_en-US.exe	REG_BINARY	0D 0A 73 F7 E6 8D D6 01 00 00 00...
\Device\HarddiskVolume4\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe	REG_BINARY	67 08 A8 19 EE 8D D6 01 00 00 00...
Microsoft.XboxGamingOverlay_8wekyb3d8bbwe	REG_BINARY	B3 F9 12 BF 68 8E D6 01 00 00 00 00...
\Device\HarddiskVolume6\RamCatcher\ram64\RamCapture64.exe	REG_BINARY	DB C3 49 2D 6D 8E D6 01 00 00 00...

Figure 72 - SYSTEM Registry UserSettings Artifact

Windows timeline activity file, *ActivitiesCache.db*, indicates both the Tor Browser install executable file, *torbrowser-install-win64-9.5.4_en-US.exe* and *firefox.exe* file used to launch the Tor Browser.

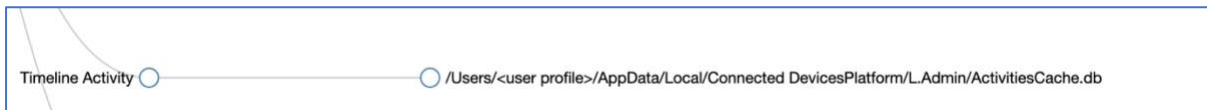


Figure 73 - *ActivitiesCache.db* Artifact

DETAILS	
ARTIFACT INFORMATION	
Application Name	C:\Users\Admin\Downloads\torbrowser-install-win64-9.5.4_en-US.exe
Activity Type	App In Use/Focus
Focus (Seconds)	14
Start Date/Time	9/18/2020 6:10:01 PM
End Date/Time	9/18/2020 6:10:15 PM
Activity ID	be4d3867-72ad-16a1-75ba-e07d625b137a
Platform	x_exe_path
Last Modified Date/Time	9/18/2020 6:10:04 PM
Last Modified On Client Date/Time	9/18/2020 6:10:15 PM
Local Only	False
EVIDENCE INFORMATION	
Source	WinValidation.E01 - Partition 4 (Microsoft NTFS, 231.71 GB)\Users\Admin\AppData\Local\ConnectedDevicesPlatform\L.Admin\ActivitiesCache.db
Recovery Method	Parsing
Deleted source	
Location	Table: Activity(rowid: 11)
Evidence number	WinValidation.E01

Figure 74 - *ActivitiesCache.db* Artifact for Tor Brower Install

DETAILS ⌵

ARTIFACT INFORMATION

Application Name: C:\Users\Admin\Desktop\Tor Browser\Browser\firefox.exe

Display Name: Start Tor Browser

Activity Type: Open App/File/Page

Focus (Seconds): 0

Start Date/Time: 9/18/2020 6:57:16 PM 🕒

Activity ID: b625718f-2e61-0ab5-8c81-f4a20de5f290

Platform: windows_win32

Last Modified Date/Time: 9/18/2020 6:57:16 PM 🕒

Last Modified On Client Date/Time: 9/18/2020 6:57:16 PM 🕒

Local Only: False

EVIDENCE INFORMATION

Source: WinValidation.E01 - Partition 4 (Microsoft NTFS, 231.71 GB)\Users\Admin\AppData\Local\ConnectedDevicesPlatform\L.Admin\ActivitiesCache.db

Recovery Method: Parsing

Deleted source:

Location: Table: Activity(rowid: 15)

Evidence number: WinValidation.E01

Figure 75 - ActivitiesCache.db Artifact for firefox.exe

Finally, the Tor Browser install files are located on the desktop of the Admin user.

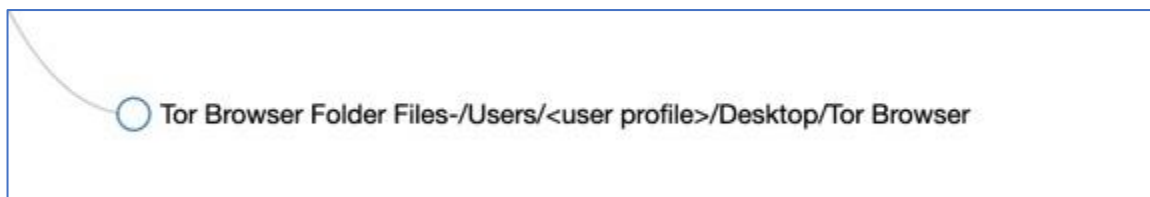


Figure 76 - Tor Browser Install Location

Name	Type	File e...	Size...	Created	Accessed	Modified
PluggableTransports	Folder			9/18/2020 6:10:13 PM	9/18/2020 6:10:13 PM	9/18/2020 6:10:13 PM
libcrypto-1_1-x64.dll	File	.dll	3,547,985	1/1/2000 12:00:00 AM	9/18/2020 6:10:12 PM	1/1/2000 12:00:00 AM
libevent-2-1-7.dll	File	.dll	997,464	1/1/2000 12:00:00 AM	9/18/2020 6:10:12 PM	1/1/2000 12:00:00 AM
libevent_core-2-1-7.dll	File	.dll	661,577	1/1/2000 12:00:00 AM	9/18/2020 6:10:12 PM	1/1/2000 12:00:00 AM
libevent_extra-2-1-7.dll	File	.dll	673,434	1/1/2000 12:00:00 AM	9/18/2020 6:10:12 PM	1/1/2000 12:00:00 AM
libgcc_s_seh-1.dll	File	.dll	1,201,785	1/1/2000 12:00:00 AM	9/18/2020 6:10:12 PM	1/1/2000 12:00:00 AM
libssl-1_1-x64.dll	File	.dll	988,536	1/1/2000 12:00:00 AM	9/18/2020 6:10:13 PM	1/1/2000 12:00:00 AM
libssp-0.dll	File	.dll	320,645	1/1/2000 12:00:00 AM	9/18/2020 6:10:13 PM	1/1/2000 12:00:00 AM
libwinpthread-1.dll	File	.dll	622,549	1/1/2000 12:00:00 AM	9/18/2020 6:10:13 PM	1/1/2000 12:00:00 AM
tor.exe	File	.exe	4,467,200	1/1/2000 12:00:00 AM	9/18/2020 6:10:13 PM	1/1/2000 12:00:00 AM
zlib1.dll	File	.dll	109,568	1/1/2000 12:00:00 AM	9/18/2020 6:10:13 PM	1/1/2000 12:00:00 AM

Figure 77 - Tor Executable Artifact

macOS Tor Launched from USB

The absence of the *places.sqlite* file on the macOS system, along with the presence of Tor Browser files is an indicator that Tor was launched from a USB drive rather than installed locally. Artifacts remain on the system but fewer than if Tor is installed locally. Like in Windows, not all artifacts indicated in the framework were located during the experimental investigation, however new artifacts were discovered that were added to the framework. The framework branching for Tor Browser artifacts that contain *TorBrowser-Data* files or user information is shown in the screenshot below followed by screenshots of the artifact as found in Magnet AXIOM.

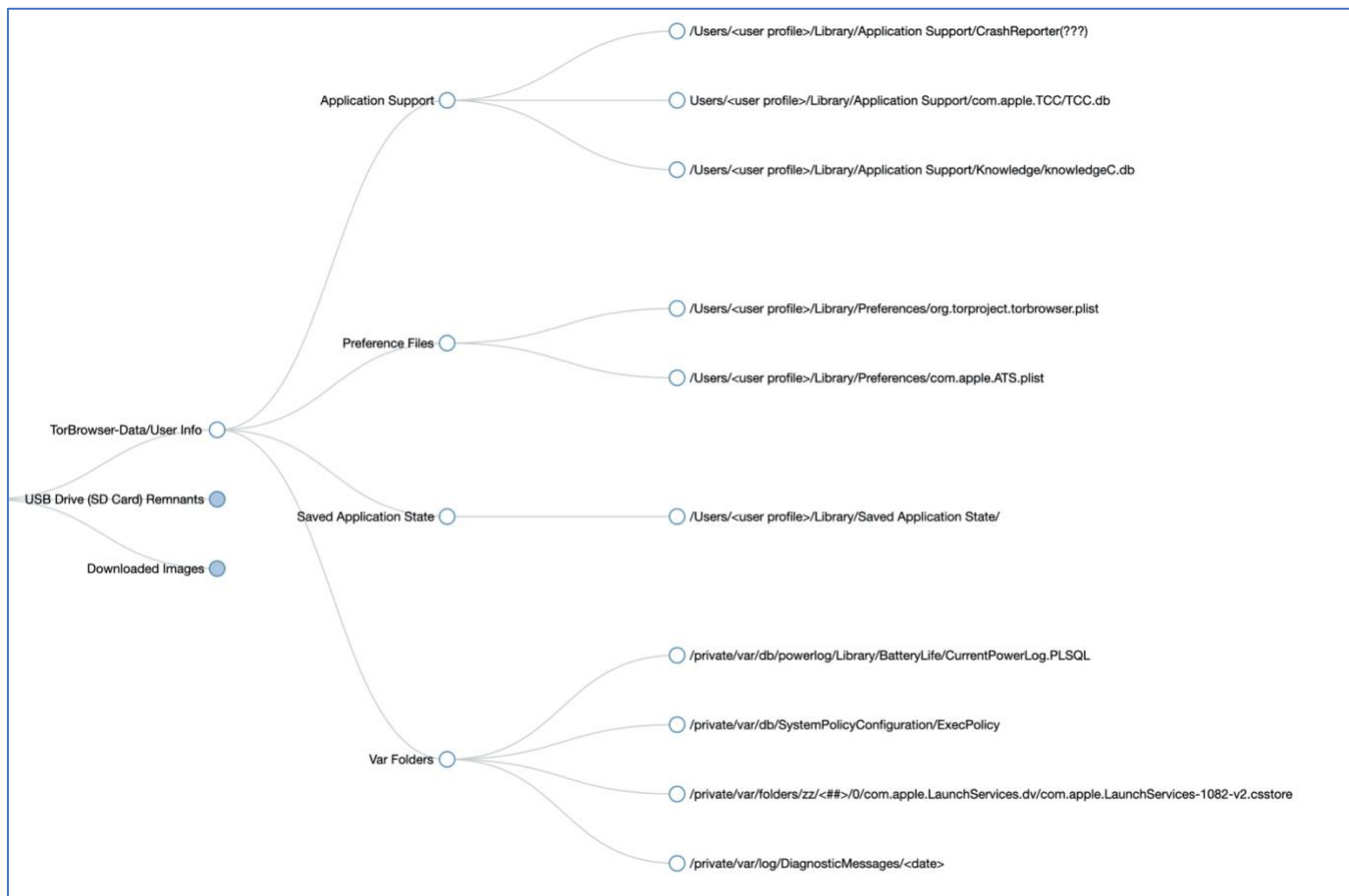


Figure 78 - macOS Validation Artifacts

knowledgeC.db

ZUUID	ZSTREAMNAME	ZVALUESTRING	ZSTRING	ZI
AE8C2C1... CAA2-42... AC0D-2E...	/notification/usage	Receive	(null)	(n
FB2524C... F25D-40... B7FF- D351594...	/notification/usage	Receive	(null)	(n
E125FBD... AB41-48...	/app/usage	com.apple.finder	(null)	(n
3CD290...	/display/isBacklit	(null)	(null)	(n
D6D4FA...	/app/usage	org.torproject.torbrowser	(null)	(n
D545EB9... AA08-79...	/app/usage	com.apple.finder	(null)	(n
EB2B115E- D2BD-46...	/display/isBacklit	(null)	(null)	(n
69E8B29...	/display/isBacklit	(null)	(null)	(n

DETAILS

FILE DETAILS

File name: knowledgeC.db
File extension: .db
Logical size: 229,376 bytes
Created: 8/6/2020 9:46:32 AM
Accessed: 9/19/2020 7:17:48 AM
Modified: 9/19/2020 7:17:11 AM
Added: 8/6/2020 9:46:32 AM
Inode: 31456
File attributes: Normal

EVIDENCE INFORMATION

Source: Partition 2 (APFS Container, 465.72 GB)\32f162c9-f679-4f1a-b79b-5aa16ae0384f\Users\admin\Library\Application Support\Knowledge\knowledgeC.db

Figure 79 - knowledgeC.db Artifact

DETAILS

FILE DETAILS

File name: org.torproject.torbrowser.plist
File extension: .plist
Logical size: 119 bytes
Created: 9/18/2020 5:53:27 PM
Accessed: 9/18/2020 5:53:27 PM
Modified: 9/18/2020 5:53:27 PM
Added: 9/18/2020 5:53:27 PM
Inode: 40595
File attributes: Normal

EVIDENCE INFORMATION

Source: Partition 2 (APFS Container, 465.72 GB)\32f162c9-f679-4f1a-b79b-5aa16ae0384f\Users\admin\Library\Preferences\org.torproject.torbrowser.plist
Evidence number: MachDValidation.E01

Figure 80 - org.torproject.torbrowser.plist Artifact

PREVIEW

EXPAND ALL FIND

- root
 - [0] ATSAutoActivationAppSpecific
 - [0] org.torproject.torbrowser = ATSAutoActivationDisable

DETAILS

FILE DETAILS

File name: com.apple.ATS.plist
 File extension: .plist
 Logical size: 137 bytes
 Created: 9/18/2020 5:53:09 PM
 Accessed: 9/19/2020 7:13:24 AM
 Modified: 9/18/2020 5:53:09 PM
 Added: 9/18/2020 5:53:09 PM
 Inode: 40323
 File attributes: Normal

EVIDENCE INFORMATION

Source: Partition 2 (APFS Container, 465.72 GB)\32f162c9-f679-4f1a-b79b-5aa16ae0384f\Users\admin\Library\Preferences\com.apple.ATS.plist
 Evidence number: MacHDValidation.E01

Figure 81 - com.apple.ATS.plist Artifact



Figure 82 - Saved Application State

CurrentPowerlog.PLSQL

Select table: PLApplicationAgent_EventForwarder

FIND BUILD QUERY EXPORT SHOW / HIDE COLUMNS

#	ID	timestamp	ASN	BundleID	Event	PID	Parent
54	132	1600451669....	155686	com.apple.eap8021x.eaptitrust	2	531	0
55	133	1600451673....	143395	com.apple.TMHelperAgent	2	479	0
56	134	1600451680....	159783	com.apple.SecurityAgent	2	539	0
57	135	1600451687....	163880	org.torproject.torbrowser	3	569	6964!
58	136	1600451687....	163880	org.torproject.torbrowser	2	569	6964!
59	137	1600451697....	167977	org.torproject.torbrowser	1	576	6964!
60	138	1600451698....	172074	org.mozilla.plugincontainer	1	592	1679!

DETAILS

FILE DETAILS

File name: CurrentPowerlog.PLSQL
File extension: .PLSQL
Logical size: 864,256 bytes
Created: 8/6/2020 2:23:09 AM
Accessed: 9/18/2020 7:04:44 PM
Modified: 9/18/2020 7:04:44 PM
Added: 8/6/2020 2:23:09 AM
Inode: 25049
File attributes: Normal

EVIDENCE INFORMATION

Source: Partition 2 (APFS Container, 465.72 GB)\32f162c9-f679-4f1a-b79b-5aa16ae0384f\private\var\db\powerlog\Library\BatteryLife\CurrentPowerlog.PLSQL
Evidence number: MacHDValidation.E01

Figure 83 - CurrentPowerlog.PLSQL Artifact

The serial number of the USB drive is found in the image verification text file created by FTK Imager and can be matched to device information stored in log files in macOS that are located in the `\private\var\db\diagnostics\Persist\` folder.

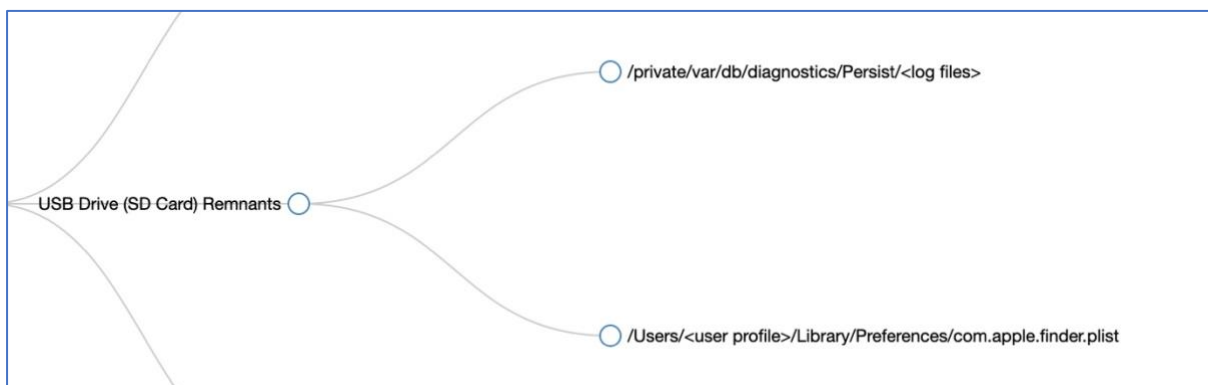


Figure 84 - USB (SD Card) Artifacts

```

Physical Evidentiary Item (Source) Information:
[Device Info]
Source Type: Physical
[Drive Geometry]
Cylinders: 3,740
Tracks per Cylinder: 255
Sectors per Track: 63
Bytes per Sector: 512
Sector Count: 60,088,320
[Physical Drive Information]
Drive Model: SanDisk Cruzer Glide USB Device
Drive Serial Number: 4C530000150307203144
Drive Interface Type: USB
Removable drive: True
Source data size: 29340 MB
Sector count: 60088320
[Computed Hashes]
MD5 checksum: f9d8e8dc5e4198fc25a495761eed8808
SHA1 checksum: 28ee631c9fa2239342a011bf8812ca9fdbe928ba

```

Figure 85 - FTK Image Verification Drive Serial Number

DETAILS

ARTIFACT INFORMATION

Identifier: 4C530000150307203144

Column Name: Serial Number

Original artifact: USB Connection History

EVIDENCE INFORMATION

Source: Partition 2 (APFS Container, 465.72 GB)
\da74ee30-6539-4c62-8ceb-00741fe67265\private\var\db
\diagnostics\Persist\0000000000000001.tracev3

Recovery Method

Deleted source

Location: File Offset 8737776

Evidence number: Apple USB 2.E01

Figure 86 – USB Artifact Serial #

Tails

A newly created Tails drive was given to the volunteer user to boot, create persistent storage, and generate dark web data. Using the framework, the investigator was easily able to follow the steps to first determine the existence of persistent storage.

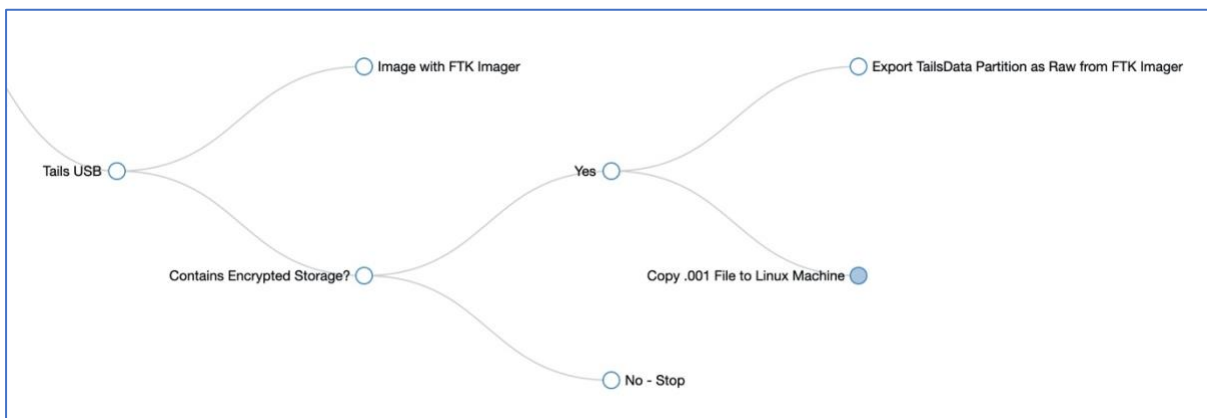


Figure 87 - Tails Persistent Storage Branching Determination

The presence of persistent storage can be determined immediately after imaging the Tails drive by adding the evidence item and checking for the presence of a TailsData partition which shows as an Unrecognized File System.

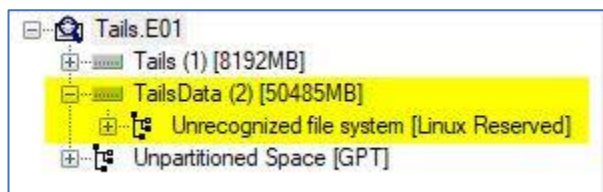


Figure 88 - Tails Persistent Storage

The TailsData partition was exported as a raw image and then copied to a Linux machine for password recovery and decryption.

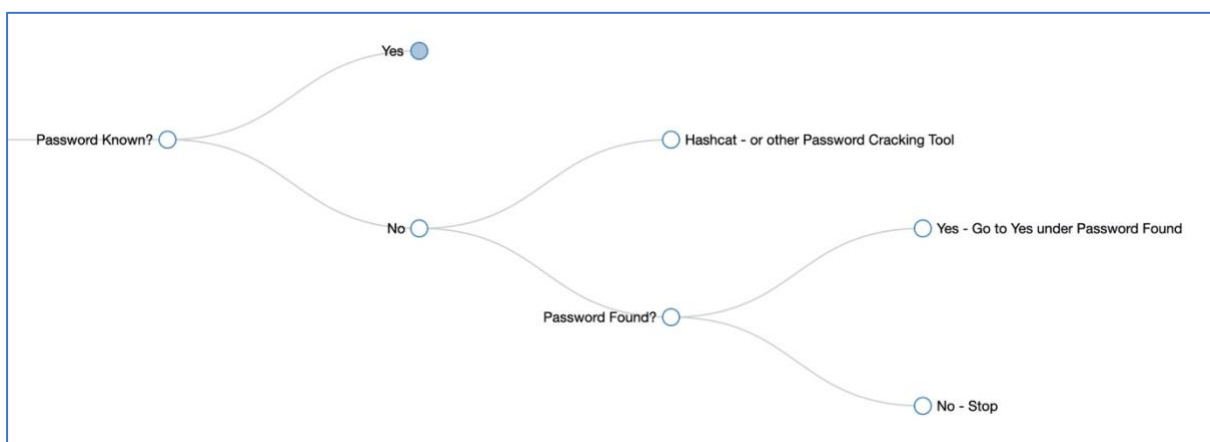


Figure 89 - Tails Password Branching

In this case, the password is unknown so Hashcat (“hashcat - advanced password recovery,” n.d.) was used to find the password (Password1!) meaning the investigation continues. If the password had been unable to be recovered the investigation would stop.

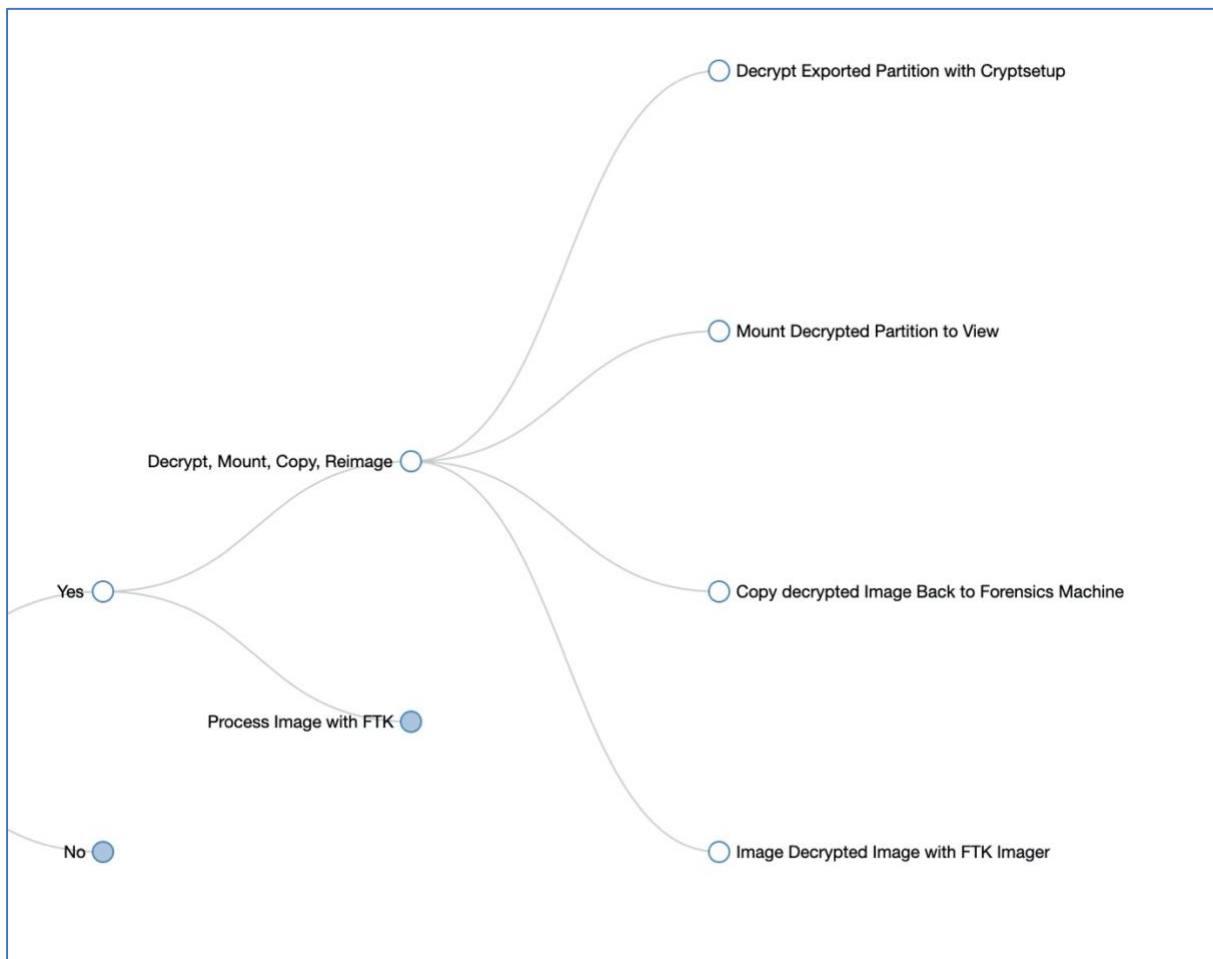


Figure 90 - Tails Decrypt, Mount, Copy, Reimage

The found password was used to decrypt the image using *cryptsetup* (“cryptsetup,” n.d.). Once decrypted, the decrypted image was mounted at which time the contents of the drive can be previewed on the Linux machine. To forensically view the data, it is copied back to the Windows forensics machine and reimaged to obtain the valuable data stored within. In this example, the existence of *.onion* sites marked as bookmarks were located in the *places.sqlite* file. The *nm-system-connections* file contains the WiFi SSID and password of the system connected to.

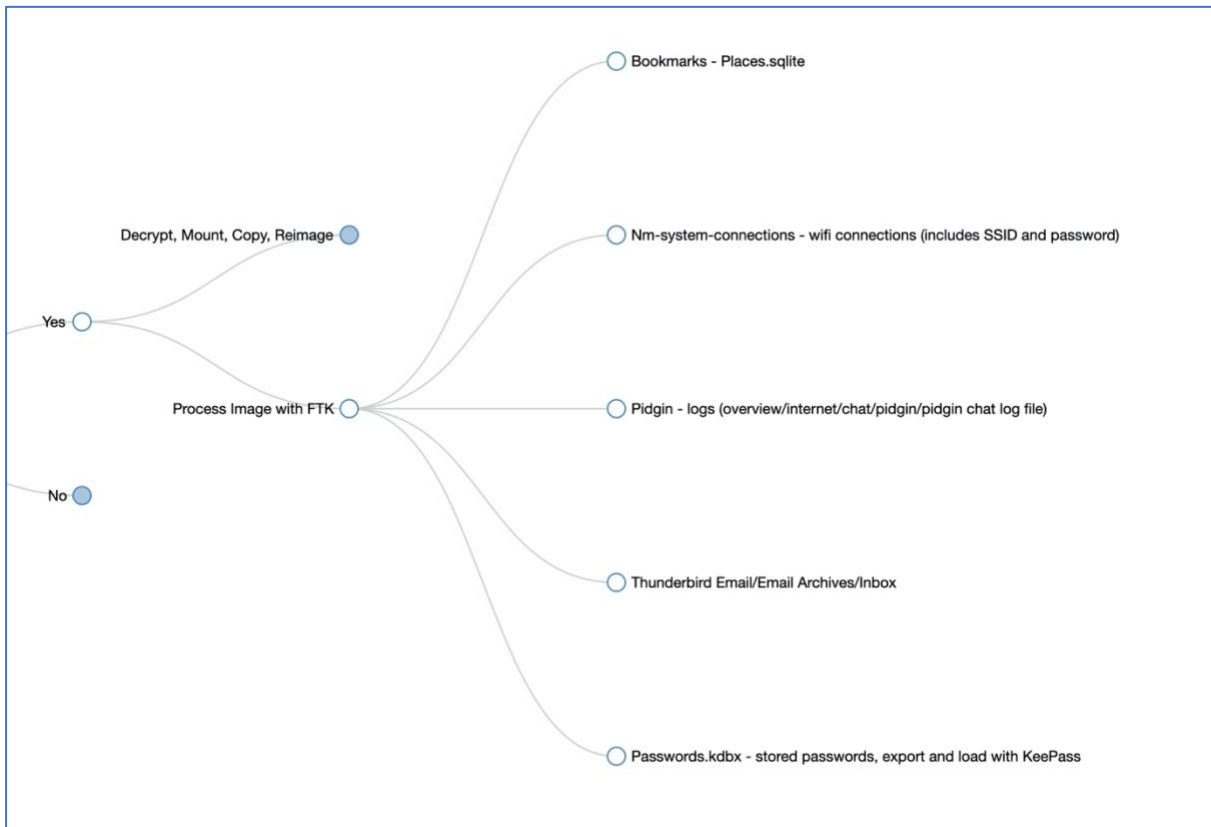


Figure 91 - Tails Persistent Storage Branching

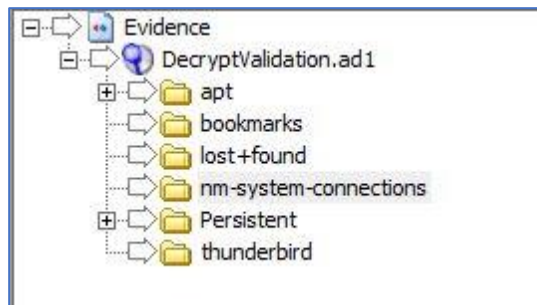


Figure 92 - Tails Persistent Storage Artifacts

Firefox SQLite Places Database				
Bookmarks				
Title	Parent	Created	Modified	URL
		9/26/2020 5:47:54 PM -0500	9/26/2020 5:56:30 PM -0500	
menu		9/26/2020 5:47:54 PM -0500	9/26/2020 5:47:54 PM -0500	
toolbar		9/26/2020 5:47:54 PM -0500	9/26/2020 5:47:55 PM -0500	
tags		9/26/2020 5:47:54 PM -0500	9/26/2020 5:47:54 PM -0500	
unfiled		9/26/2020 5:47:54 PM -0500	9/26/2020 5:56:30 PM -0500	
mobile		9/26/2020 5:47:54 PM -0500	9/26/2020 5:47:54 PM -0500	
	menu	9/26/2020 5:47:54 PM -0500	9/26/2020 5:47:54 PM -0500	
Learn more about Tor	toolbar	9/26/2020 5:47:55 PM -0500	9/26/2020 5:47:55 PM -0500	https://www.torproject.org/
The Tor Blog	toolbar	9/26/2020 5:47:55 PM -0500	9/26/2020 5:47:55 PM -0500	https://blog.torproject.org/
NCIDE - Northern California Illicit Digital Economy Task Force	unfiled	9/26/2020 5:50:56 PM -0500	9/26/2020 5:50:56 PM -0500	http://ncidelf3j26mdt.vf.onion/
Hidden Answers	unfiled	9/26/2020 5:54:22 PM -0500	9/26/2020 5:54:22 PM -0500	http://answerszuvs3gg2l64e6hmnryudl5zgrwm3vh65hszsdghbddvfiqd.onion/
Title	Parent	Created	Modified	URL
Dark Eye - The DarkNet Monitor	unfiled	9/26/2020 5:56:30 PM -0500	9/26/2020 5:56:30 PM -0500	http://darkeypxw7cuu2cppnjlqqaav6j42gyt43clcn4vjjf7llfyly5cxid.onion/

Figure 93 - places.sqlite Artifacts

```

[connection]
id=mywifilab
uuid=52f72402-7c8d-4497-af4b-b7a578aa2b7e
type=wifi
permissions=

[wifi]
mac-address=AC:BC:32:BF:C6:8D
mac-address-blacklist=
mode=infrastructure
ssid=mywifilab

[wifi-security]
auth-alg=open
key-mgmt=wpa-psk
psk=Fails8!vineyards

[ipv4]

```

Figure 94 - Nm-system-connections Artifact

Previous Case Comparison

Using a case previously worked at the Digital Forensics Lab at DSU, a comparison was made of the artifacts identified in the original casework with artifacts identified using the created framework. This particular case involved a user with a laptop with Windows 7 Home installed who had launched the Tor Browser from a SD card. The suspect in this case was thought to be purchasing drugs on the dark web. Indications of CCleaner, a program used to remove things like unwanted files, temporary internet files, and unneeded registry entries, was

identified during the exam which could limit some artifacts, however despite this several artifacts were still identified (“CCleaner.com - What is CCleaner?,” n.d.).

Using the framework during the examination of the SD card revealed new artifacts such as Firefox add-ons in the *extensions.json* file. Additionally, and more importantly, using the Dark Web Artifact Framework, the serial number of the SD card was located and matched to artifacts located on the hard drive proving the user launched Tor from that particular SD card.

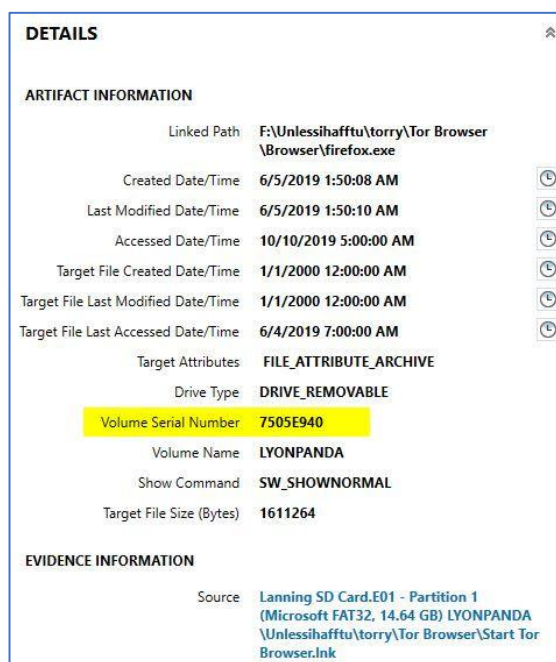


Figure 95 - SD Card Serial Number

Following the Dark Web Artifact Framework during the examination of the hard drive of the laptop revealed artifacts in locations that were not discovered during the initial examination. Files that were located using the framework that were not indicated in the initial report include *\$UsnJrnl:\$J*, *UsrClass.dat*, *Amcache.hve*, *Security.evtx*, *Application.evtx*, and *MPLLog-07132009-221054.log*. During the examination of the registry, new artifacts such as *PCI* and *PCISTOR* were located since the user launched Tor from an SD card. Keyword searches conducted during the initial examination produced four Tor browser artifacts, where using the Dark Web Artifact Framework produced all of those artifacts plus the eight additional artifacts listed above. This evaluation demonstrates that the Dark Web Artifact Framework allows an investigator to discover more artifacts than by keyword searches alone. The scope of the original investigation included artifacts outside of those identified in the Dark Web Artifact Framework such as cryptocurrency and drug references making comparing analyst time with

and without the framework difficult. However, using the framework allows an analyst to point directly to these artifacts. This direct linkage minimizes the necessity to further determine whether these artifacts could be left behind for other reasons, thereby reducing false positives and saving analysts valuable time.

Expert Opinion of South Dakota DCI ICAC Team Members

The primary stakeholders of this research are identified as the investigators trying to solve crimes committed using dark web sites such as buying illegal drugs, identity theft, weapons, terrorism, illegal pornography, and other criminal material. To assess whether the designed framework contributes to the goal of the stakeholders, the expert opinion of practicing digital forensic investigators was sought. The South Dakota Division of Criminal Investigation (DCI) Internet Crimes Against Children (ICAC) task force focuses on the investigation of children who are exploited via the internet or electronic means (*DIVISION OF CRIMINAL INVESTIGATION*, n.d.). Five members of the ICAC task force were asked to evaluate the Dark Web Artifact Framework and give their expert opinion on the ease, adaptability, and usefulness of the framework. The credentials of the ICAC task force members as they provided them are found in Appendix B. Collectively, these five individuals have over 63 years of law enforcement experience and over 28 years of experience in digital forensics.

To complete the assessment of the Dark Web Artifact Framework these investigators were each asked the same set of questions to evaluate the effectiveness of the framework. The list of questions posed can be found in Appendix C. The responses to these questions were collected, analyzed, and are summarized below:

The step-by-step design of the framework makes it easy to follow and navigate. It is a roadmap for locating dark web artifacts, with the breakdown between operating systems providing additional focus and granularity. The framework provides efficiency for locating artifacts, or ruling out the presence of artifacts, if following the framework returns a null result. The interactive, decision-making steps brings awareness of multiple scenarios that may exist during an investigation.

The framework could be adapted to anything that there is information and a need for. For example, further branching could be added to accommodate changes or updates to specific

operating systems, platforms, or conditions. Android and iOS options could be added as more apps are available to access the dark web using mobile devices. The framework could also be adapted to accommodate other scenarios, such as providing step-by-step instructions for performing similar tasks with different software programs.

The links to further documentation and resources makes the framework useful and gives detailed information about each artifact. This additional information provides supporting documentation for investigators when completing forensic examination reports to support their findings. The inclusion of macOS provides investigators who have less experience with macOS a way to quickly navigate to those artifacts. Each examination that is performed is different so using the framework is a way to quickly locate dark web artifact locations.

The rise in the number of devices requiring examination combined with the increasing amount of data stored on these devices makes an “at your fingertips” framework a time saver. The framework gives investigators a detailed listing of relevant artifacts that can be used as a checklist to ensure artifacts are not overlooked. It eliminates the need to manually search through the file structure. Investigators may be less familiar with file structures for some operating systems, so the framework makes it easy to quickly determine if those artifacts are present on a system. It also reduces the need to rely on keyword searches which may or may not accurately reflect the presence of all artifacts.

Further enhancements could be made by having the links be more focused to provide more specific, detailed information or to have information on artifact-specific anomalies regarding operating system versions.

Summary of Treatment Validation

The treatment validation steps described above demonstrate that the created Dark Web Artifact Framework meets the requirements set forth by the researcher.

Easy to follow ✓

In each validation step listed above, the Dark Web Artifact Framework was easy to follow when locating artifacts. Each step demonstrated the ease of locating artifacts and when new artifacts were discovered they were added to the framework.

Consider both Windows and macOS operating systems ✓

Both Windows PC and macOS operating systems were considered and researched extensively while creating this framework. When designing the Dark Web Artifact Framework, data was generated and drives imaged and investigated on both systems. This was demonstrated in the first validation step using a volunteer user to generate dark web data on both a Windows PC and macOS system.

Consider multiple ways to access the dark web using Tor ✓

The dark web was accessed in multiple ways on each operating system when designing the Dark Web Artifact Framework. First, each was booted using a Tails stand-alone operating system. Next, within each operating system Tor was launched from a USB drive. Last, within each operating system Tor was installed and launched locally. This was demonstrated in the second validation step when comparing data from a previously worked case using the designed framework. The previous case involved a Windows PC with a removable SD card that was used to launch Tor.

Be adaptable to future platforms ✓

The Dark Web Artifact Framework was designed to branch into categories prior to pointing to a specific folder or file location.

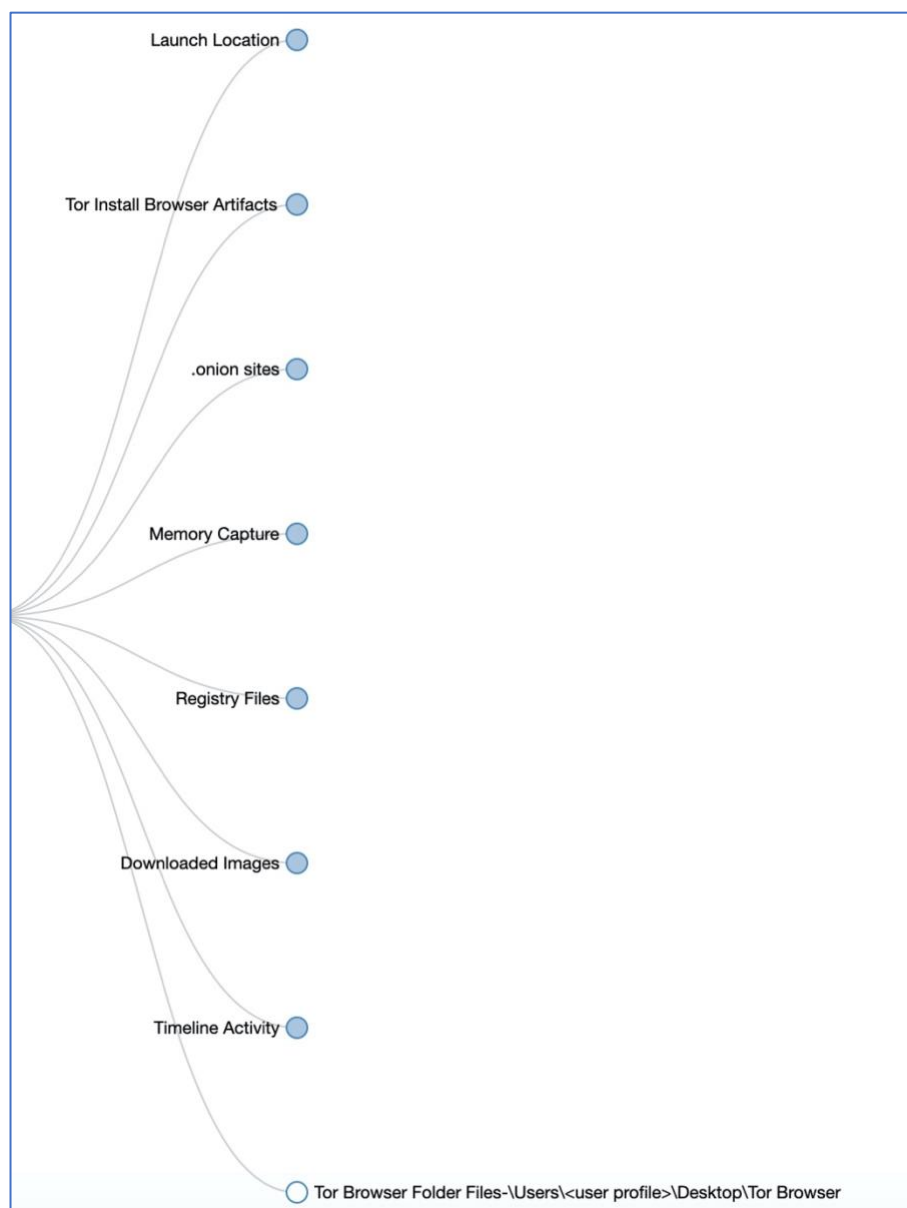


Figure 96 - Windows Artifact Categories

Future updates of either Windows PC or macOS operating systems may necessitate different files or folder locations but the categories will likely remain the same. If the location of artifacts changes to different folders or files, only the end branching of the framework would require updating rather than starting over completely. The Windows categories of launch location, Tor install browser artifacts, .onion sites, memory capture, registry files, images downloaded, the Windows timeline and Tor Browser install files will all be relevant even if the folder or file locations change. The same is true for the macOS categories of the Tor Browser

user files, .onion sites, downloaded images, and memory capture. This similarity of categories between operating systems demonstrates the adaptability to other platforms.

Additional branching of the framework can be added to account for other platforms. Linux, virtual environments such as Whonix, and mobile operating systems of Android or iOS could be added to create their own branching.

CHAPTER 6 – CONCLUSION

This chapter discusses the accomplishment of the objective of this research, the contribution, and limitations and future research. The dark web has become synonymous with nefarious and illicit content in forum sites and underground marketplaces that sell weapons, illegal drugs, stolen credit cards, stolen user credentials, child pornography, and more. Finding traces of dark web activity indicating a connection to illegal material may be the difference between a suspect being charged with a crime or being allowed to go free to continue to commit more of these crimes. Meeting the objective of this research and making a contribution by assisting in providing evidence that a user has accessed dark web market sites to acquire illegal goods or services is valuable to investigators of these cases.

Objective Accomplishment

The primary objective of this project was to design a reusable framework that can be used by digital forensic investigators when searching for host-based artifacts on a computer where the use of the dark web is suspected in the commission of a crime. The objective was accomplished with the Dark Web Artifact Framework created using the OSINT Framework as a template. When using the framework, it provides choices to an investigator on the type of system being investigated and the method used to access the dark web. If during the investigation it is discovered that artifacts are not being located using one method, it is easy for the investigator to go back to another branch, or open multiple branches, in an attempt to locate more or different artifacts. In addition to host-based artifacts, the Dark Web Artifact Framework provides branching to locate artifacts left on removable media such as USB drives or SD cards. The framework also provides the means to link removable media to the computer by locating the serial number on artifacts both on the removable media and the hard drive. Lastly, the Dark Web Artifact Framework also provides a roadmap for investigators to identify and decrypt persistent storage on a Tails USB drive.

Contribution

A digital component can be found in nearly every crime committed today. The amount of data that can be found on these devices is often measured in terabytes rather than gigabytes leaving investigators thousands or hundreds of thousands of artifacts to pour through on a single device. The reusable, paperless, comprehensive framework that was designed for this research provides investigators with a map to follow to locate the necessary artifacts to determine if the system being investigated has been used to access the dark web for the purpose of committing a crime. Addressing both Windows PC and macOS was necessary as cases often include evidence of both types. A resource such as this framework provides a time-saving method for investigators to use.

Limitations and Future Research

One limitation of this research are the methods considered of accessing the dark web. Both Windows PC and macOS were considered and accessed the dark web in multiple ways of using a bootable operating system in Tails, launching Tor from a USB drive or SD card, and installing Tor Browser locally. There are additional ways to access the dark web which were not considered such as using a Linux machine, using a virtual machine such as Whonix, using Orbot from an Android phone, using Onion Browser from an iPhone, or using an alternate operating system such as Qubes. Future research to enhance the Dark Web Artifact Framework would include the addition of other operating systems such as those mentioned above, in particular mobile systems of Android and iOS. Mobile devices are increasingly the focus of investigations with storage capacity of up to 512GB. This future research would be added as further branching to the existing framework. A second limitation is that the focus of this research was on host-based artifacts, network-based artifacts were not considered for this research.

In the creation of this framework, a process itself was created that will contribute to future works. The yes/no, if/then structure of the framework is adaptable to fit with workflows in any area that would benefit from a recurring process.

This research addressed both Windows PC and macOS systems, however the purpose of this research was not to compare the quantity of artifacts left on each operating system.

Future research to consider would be quantifying the artifacts left behind on each operating system to determine the footprint left behind when installing Tor. Removing Tor from each operating system by following the directions on the Tor Project website (“UNINSTALLING | Tor Project | Tor Browser Manual,” n.d.) removes the Tor install files but leaves many artifacts behind. There are resources to be found for removing traces of Tor from each operating system that could be combined into a helpful go-to guide for removing all traces of Tor from a system.

An additional future enhancement to consider is a method to take the information found using the framework and provide an automatic method for transferring the information found to the case report that investigators write. Case reports are a vitally important part of an investigation that tells the story of the case. An automatic means of transferring the information from the framework to the case report would be another time-saving and valuable step.

REFERENCES

- 5 Dark Web Browsers for Deep Web Browsing in 2018. (n.d.). Retrieved October 6, 2019, from <https://www.thedarkweblinks.com/dark-web-browsers/>
- About DARPA. (n.d.). Retrieved September 28, 2019, from <https://www.darpa.mil/about-us/about-darpa>
- Abraham, S., Silva, T., Decourcy, R., & Cardon, J. (n.d.). *Tails & Tor ... and other tools for Safeguarding Online Activities Forensic Investigations*. Retrieved from <https://arxiv.org/pdf/1710.08705.pdf>
- Al Jawaheri, H., Al Sabah, M., Boshmaf, Y., & Erbad, A. (2020). Deanonymizing Tor hidden service users through Bitcoin transactions analysis. *Computers & Security*, 89, 101684. <https://doi.org/10.1016/j.cose.2019.101684>
- Amazing Facts and Figures About the Evolution of Hard Disk Drives - Pingdom Royal. (2019). Retrieved September 28, 2019, from <https://royal.pingdom.com/amazing-facts-and-figures-about-the-evolution-of-hard-disk-drives/>
- Anash, K. (2019). Playpen Member Pedophile Brothers Sentenced to More Than 44 Years in Prison | DarknetStats. Retrieved October 6, 2019, from <https://www.darknetstats.com/playpen-member-pedophile-brothers-sentenced-to-more-than-44-years-in-prison/>
- Baryamureeba, V., & Tushabe, F. (2004). The enhanced digital investigation process model. *Proceedings of the Digital Forensic Research Conference, DFRWS 2004 USA*, 1–9.
- Bazli, B., Wilson, M., & Hurst, W. (2017). The dark side of I2P, a forensic analysis case study. *Systems Science and Control Engineering*, 5(1), 278–286. <https://doi.org/10.1080/21642583.2017.1331770>
- Bio — John W Creswell. (n.d.). Retrieved March 8, 2020, from <https://www.johnwreswell.com/bio>
- Cardenas-Haro, J. A., & Dawson, M. (2016). Tails linux operating system: The amnesiac incognito system in times of high surveillance, its security flaws, limitations, and strengths in the fight for democracy. *Security Solutions for Hyperconnectivity and the Internet of Things*, 260–271. <https://doi.org/10.4018/978-1-5225-0741-3.ch010>
- Carrier, B., & Spafford, E. (2004). *DIGITAL FORENSIC RESEARCH CONFERENCE An*

- Event-Based Digital Forensic Investigation Framework*. Retrieved from https://www.dfrws.org/sites/default/files/session-files/paper-an_event-based_digital_forensic_investigation_framework.pdf
- Cattle, A. E. (n.d.). *DIGITAL TAHRIR SQUARE: AN ANALYSIS OF HUMAN RIGHTS AND THE INTERNET EXAMINED THROUGH THE LENS OF THE EGYPTIAN ARAB SPRING*. Retrieved from <http://www.telegraph.co.uk/>
- CCleaner.com - What is CCleaner? (n.d.). Retrieved September 27, 2020, from <https://www.ccleaner.com/docs/ccleaner/introducing-ccleaner/what-is-ccleaner>
- Central Intelligence Agency. (2019). CIA's Latest Layer: An Onion Site — Central Intelligence Agency. Retrieved January 30, 2020, from <https://www.cia.gov/news-information/press-releases-statements/2019-press-releases-statements/ciagov-over-tor.html>
- Controlling Saved Application States - krypted. (n.d.). Retrieved August 6, 2020, from <https://krypted.com/mac-security/controlling-saved-application-states/>
- Cox, J. (n.d.). "Operation Hyperion" Targets Suspected Dark Web Users Around the World - VICE. Retrieved December 26, 2019, from https://www.vice.com/en_us/article/z438d8/operation-hyperion-targets-suspected-dark-web-users-around-the-world
- Cox, J. (2016). After High Profile Busts, Dozens of Dark Web Child Porn Sites Remain - VICE. Retrieved October 6, 2019, from https://www.vice.com/en_us/article/8q8exb/after-high-profile-busts-dozens-of-dark-web-child-porn-sites-remain
- Creswell, J. W. (2014). *Research Design Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.
- cryptsetup. (n.d.). Retrieved September 27, 2020, from <https://gitlab.com/cryptsetup/cryptsetup/>
- Darcie, W., Boggs, R. J., Sammons, J. M., & Fenger, T. (2015). *Online Anonymity: Forensic Analysis of the Tor Browser Bundle*. Retrieved from http://www.marshall.edu/forensics/files/WinklerDarcie_ResearchPaper_8-6-141.pdf
- Dayalamurthy, D. (2013). Forensic Memory Dump Analysis And Recovery Of The Artefacts Of Using Tor Bundle Browser – The Need. *Australian Digital Forensics Conference*.

<https://doi.org/10.4225/75/57b3c7f3fb86e>

Décary-Hétu, D., & Giommoni, L. (2017). Do police crackdowns disrupt drug cryptomarkets?

A longitudinal analysis of the effects of Operation Onymous. *Crime, Law and Social Change*, 67(1), 55–75. <https://doi.org/10.1007/s10611-016-9644-4>

Dingledine, R., Mathewson, N., & Syverson, P. (2004). Tor: The second-generation onion router. *SSYM'04 Proceedings of the 13th Conference on USENIX Security Symposium*, 13(03), 21. <https://doi.org/10.1.1.4.6896>

DIVISION OF CRIMINAL INVESTIGATION. (n.d.). Retrieved from

[https://atg.sd.gov/docs/2011 Division of Criminal Investigation Annual Report.pdf](https://atg.sd.gov/docs/2011%20Division%20of%20Criminal%20Investigation%20Annual%20Report.pdf)

Doran, M. D. (2014). *A FORENSIC LOOK AT BITCOIN CRYPTOCURRENCY* by Michael Dennis Doran A Capstone Project Submitted to the Faculty of Utica College May 2014 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Cybersecurity. (May).

Dr. Digvijaysinh Rathod. (2017). Darknet Forensics. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 6(4, July-August 2017), 77–79. Retrieved from <https://github.com/HelloZeroNet/ZeroNet>

Epp, T. (2019). Sioux Falls man convicted in nationwide federal “dark web,” Bitcoin drug probe | News | KELO Newstalk 1320 107.9. Retrieved October 6, 2019, from <https://kelo.com/news/articles/2019/may/31/sioux-falls-man-convicted-in-nationwide-federal-dark-web-bitcoin-drug-probe/>

European Monitoring Centre for Drugs and Drug Addiction. (2017). *Drugs and the darknet*. <https://doi.org/10.2810/783427>

FBI. (2018). Operation Disarray — FBI. Retrieved December 26, 2019, from <https://www.fbi.gov/news/stories/operation-disarray-040318>

Federal Bureau of Investigation. (2017). ‘Playpen’ Creator Sentenced to 30 Years — FBI. Retrieved October 6, 2019, from <https://www.fbi.gov/news/stories/playpen-creator-sentenced-to-30-years>

Feds Dismantled the Dark-Web Drug Trade—but It’s Already Rebuilding | WIRED. (2019). Retrieved October 6, 2019, from <https://www.wired.com/story/dark-web-drug-takedowns-deepdotweb-rebound/>

Find and install add-ons to add features to Firefox | Firefox Help. (n.d.). Retrieved August 23,

2020, from <https://support.mozilla.org/en-US/kb/find-and-install-add-ons-add-features-to-firefox>

- Forensic Focus. (2012). Digital Forensics is not just HOW but WHY | Forensic Focus - Articles. Retrieved January 30, 2020, from Forensic Focus website: <https://articles.forensicfocus.com/2012/07/03/digital-forensics-is-not-just-how-but-why/>
- Fosburgh, L. (1973). Chief Teller Is Accused of Theft Of \$1.5-Million at a Bank Here - The New York Times. Retrieved December 8, 2019, from The New York Times website: <https://www.nytimes.com/1973/03/23/archives/chief-teller-is-accused-of-theft-of-15million-at-a-bank-here-teller.html>
- Harris, R. (2006). Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. *Digital Investigation*, 3(SUPPL.), 44–49. <https://doi.org/10.1016/j.diin.2006.06.005>
- hashcat - advanced password recovery. (n.d.). Retrieved September 5, 2020, from <https://hashcat.net/hashcat/>
- Hay Newman, L. (2019). The CIA Sets Up Shop on Tor, the Anonymous Internet | WIRED. Retrieved December 9, 2019, from Wired website: <https://www.wired.com/story/cia-sets-up-shop-on-tor/>
- Hayes, D. (2015). Computer Forensics Critical in the Trial of Silk Road’s Ross Ulbricht – Homeland Security Today. Retrieved February 11, 2020, from Homeland Security Today.US website: <https://www.hstoday.us/columns/best-practices/computer-forensics-critical-in-the-trial-of-silk-road-s-ross-ulbricht/>
- Hetherington, C. (2019). *personal communication*.
- Hevner, A., March, S. T., Park, J., & Ram, S. (1996). Design Science in Information Systems Research. *AI and Society*, 10(2), 199–217. <https://doi.org/10.1007/BF01205282>
- Honan, M. (2012). Kill the Password: A String of Characters Won’t Protect You | WIRED. Retrieved December 8, 2019, from <https://www.wired.com/2012/11/ff-mat-honan-password-hacker/>
- How to: Use Tor on macOS | Surveillance Self-Defense. (n.d.). Retrieved August 23, 2020, from <https://ssd.eff.org/en/module/how-use-tor-macos>
- How to Access the Dark Net and Deep Web Safely - Step by Step Guide. (n.d.). Retrieved December 29, 2019, from <https://www.comparitech.com/blog/vpn-privacy/how-to->

- access-the-deep-web-and-darknet/
IACIS - History. (n.d.). Retrieved September 22, 2019, from <https://www.iacis.com/about-2/history/>
- Jacoby Mentor, C., & Chow, M. (2016). *The Onion Router and the Darkweb*. Retrieved from <https://guardianproject>.
- Jadoon, A. K., Iqbal, W., Amjad, M. F., Afzal, H., & Bangash, Y. A. (2019). Forensic Analysis of Tor Browser: A Case Study for Privacy and Anonymity on the Web. *Forensic Science International*, 299(August), 59–73.
<https://doi.org/10.1016/j.forsciint.2019.03.030>
- Kennecke, A. (n.d.). California man accused of shipping fentanyl through mail, including to South Dakota residents | KELOLAND.com. Retrieved October 11, 2019, from 2019 website: <https://www.keloland.com/news/opioid-crisis/california-man-accused-of-shipping-millions-of-fentanyl-pills-through-mail-including-to-south-dakota-residents-through-dark-web/>
- Kent, K., Chevalier, S., Grance, T., & Dang, H. (2006). Guide to Integrating Forensic Techniques into Incident Response. *The National Institute of Standards and Technology*.
- Klein, M. (2019). SF man admits to buying meth online and selling it in South Dakota | KELOLAND.com. Retrieved October 11, 2019, from <https://www.keloland.com/uncategorized/sf-man-admits-to-buying-meth-online-and-selling-it-in-south-dakota/>
- Knowledge is Power! Using the macOS/iOS knowledgeC.db Database to Determine Precise User and Application Usage — mac4n6.com. (n.d.). Retrieved August 23, 2020, from <https://www.mac4n6.com/blog/2018/8/5/knowledge-is-power-using-the-knowledgecdb-database-on-macos-and-ios-to-determine-precise-user-and-application-usage>
- Kohen, I. (2017). Darknet Chronicles Pt 1: Clearnet vs Darknet. Retrieved September 12, 2019, from Business 2 Community website:
<https://www.business2community.com/cybersecurity/darknet-chronicles-pt-1-clearnet-vs-darknet-01972328>
- Kohn, Michael; Eloff, JHP; Olivier, M. (2006). *Framework for a Digital Forensic Investigation*. <https://doi.org/10.14943/jjvr.64.suppl.s33>
- Kothari, C. R. (2004). *Research Methodology : Methods & Techniques*. New Delhi: New Age

International.

- KSFY News. (2018). Push to fight opioids in South Dakota results in cases against 19 alleged traffickers. Retrieved April 12, 2019, from KSFY News website:
<https://www.ksfy.com/content/news/Push-to-fight-opioids-in-South-Dakota-results-in-cases-against-19-alleged-traffickers-501259392.html>
- Lange, R. (2019). *personal communication*.
- Lewis, G. (2020). South Dakota Residents Handed Prison Sentences For Roles In Dark Web Meth Distribution.
- Liles, S., Crimmins, D., Falk, C., Fowler, S., Gravel, C., Kouremetis, M., ... Liles, S. (2015). US Bank of Cyber. *Proceedings of the 16th Annual Information Security Symposium*, 30.
- Mac OS Daily Logs | Salt Forensics. (n.d.). Retrieved August 23, 2020, from
<https://salt4n6.com/2018/12/11/mac-os-daily-logs/>
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15, 251–266.
 Retrieved from
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.826.5567&rep=rep1&type=pdf>
- Miller, S. G., Hong, T. W., Wiley, B., Sandberg, O., & Clarke, I. (2002). Protecting Free with Freenet. *IEEE Internet Computing*, (6:1), 40–49.
- Moore, D., & Rid, T. (2016). Cryptopolitik and the Darknet. *Survival*, 58(1), 7–38.
<https://doi.org/10.1080/00396338.2016.1142085>
- Negi, N. (2017). Comparison of Anonymous Communication Networks-Tor, I2P, Freenet. *International Research Journal of Engineering and Technology*. Retrieved from
<https://www.torproject.org/index.html.en>
- Net Marketshare. (n.d.). Operating system market share. Retrieved October 27, 2019, from
<https://netmarketshare.com/operating-system-market-share.aspx>
- Nordine, J. (n.d.). *OSINT Framework*. Retrieved from <https://github.com/lockfale/osint-framework>
- Oliver, C., Brannon, S., & Song, T. (2008). *Usab5601.Pdf*. Retrieved from
<http://www.justice.gov/sites/default/files/usao/legacy/2008/02/04/usab5601.pdf>
- Osborne, G., Turnbull, B., & Slay, J. (2010). The “Explore, Investigate and Correlate” (EIC)

- conceptual framework for digital forensics information visualisation. *ARES 2010 - 5th International Conference on Availability, Reliability, and Security*, pp. 629–634.
<https://doi.org/10.1109/ARES.2010.74>
- Paul, K. (2018). Ancient Artifacts vs. Digital Artifacts: New Tools for Unmasking the Sale of Illicit Antiquities on the Dark Web. *Arts*, 7(2), 12. <https://doi.org/10.3390/arts7020012>
- Perry, Mike; Clark, Erinn; Murdoch, Steven; Koppen, G. (2018). The Design and Implementation of the Tor Browser [DRAFT].
- Pollitt, M., & History, M. P. A. (2010). *A History of Digital Forensics*. 3–15.
https://doi.org/10.1007/978-3-642-15506-2_1i
- Pollitt, M. M. (1995). 18th National Information Systems Security Conference, October 10-13, 1995, Baltimore Convention Cen.pdf. *Computer Forensics: An Approach to Evidence in Cyberspace*, 487–491.
- Popular Computer Forensics Top 21 Tools. (2019). Retrieved September 28, 2019, from Infosec website: <https://resources.infosecinstitute.com/computer-forensics-tools/#gref>
- Poulsen, K. (2013). FBI Admits It Controlled Tor Servers Behind Mass Malware Attack | WIRED. Retrieved October 22, 2019, from Wired website:
<https://www.wired.com/2013/09/freedom-hosting-fbi/>
- Rajasekar, S., Philominathan, P., & Chinnathambi, V. (2014). Research methodology. *Contributions to Management Science*, 75–100. https://doi.org/10.1007/978-3-319-04069-1_4
- Reith, Mark; Carr, Clint; Gunsch, G. (2009). Two models of digital forensic examination. *4th International Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE 2009*, 1(3), 42–53. <https://doi.org/10.1109/SADFE.2009.8>
- Richard, G. G., & Case, A. (2014). In lieu of swap : Analyzing compressed RAM in Mac OS X and Linux. *Digital Investigation*, 11, S3–S12.
<https://doi.org/10.1016/j.diin.2014.05.011>
- Routley, N. (2018). A Data-Driven Look At Dark Web Marketplaces. Retrieved September 8, 2019, from <https://www.visualcapitalist.com/data-driven-look-dark-web-marketplaces/>
- Russell, T. (2020). *personal communication*.
- Scherer, C. (2020). *personal communication*.
- Security, N. (2012). Tor-hidden online narcotics store, ‘The Farmer’s Market’, brought down

- in multinational sting. Retrieved October 2, 2019, from <https://nakedsecurity.sophos.com/2012/04/23/farmers-market-tor-narcotics/>
- Syverson, P. (2005). Onion Routing: Executive Summary. Retrieved September 29, 2019, from <https://www.onion-router.net/Summary.html#Solution>
- Tails - Privacy for anyone anywhere. (n.d.). Retrieved October 11, 2019, from <https://tails.boum.org/>
- The United States Department of Justice. (2020). International Law Enforcement Operation Targeting Opioid Traffickers on the Darknet Results in over 170 Arrests Worldwide and the Seizure of Weapons, Drugs and over \$6.5 Million | OPA | Department of Justice. Retrieved October 3, 2020, from <https://www.justice.gov/opa/pr/international-law-enforcement-operation-targeting-opioid-traffickers-darknet-results-over-170>
- The Washington Post. (n.d.). NSA report on the Tor encrypted network. Retrieved October 26, 2019, from The Washington Post website: <https://www.washingtonpost.com/apps/g/page/world/nsa-report-on-the-tor-encrypted-network/501/>
- TOR History. (n.d.).
- Tor Project. (n.d.). Retrieved from www.torproject.org
- Tor Project | What is GetTor? (n.d.). Retrieved September 14, 2020, from <https://gettor.torproject.org/>
- U.S. Attorney's Office. (2018). First Nationwide Undercover Operation Targeting Darknet Vendors Results in Arrests of More Than 35 Individuals Selling Illicit Goods and the Seizure of Weapons, Drugs, and More Than \$23.6 Million | USAO-SD | Department of Justice. Retrieved April 11, 2019, from United States Department of Justice website: <https://www.justice.gov/usao-sd/pr/first-nationwide-undercover-operation-targeting-darknet-vendors-results-arrests-more-35>
- UNINSTALLING | Tor Project | Tor Browser Manual. (n.d.). Retrieved August 16, 2020, from <https://tb-manual.torproject.org/uninstalling/>
- United States Department of Justice. (2019a). Alleged Dark Web Child Pornography Facilitator Extradited to the United States to Face Federal Charges | OPA | Department of Justice. Retrieved October 6, 2019, from <https://www.justice.gov/opa/pr/alleged-dark-web-child-pornography-facilitator-extradited-united-states-face-federal-charges>

- United States Department of Justice. (2019b). Global investigation of Dark Web drug network leads to arrest of 3 German Nationals. Retrieved January 19, 2020, from Justice News website: <https://www.dea.gov/press-releases/2019/05/03/global-investigation-dark-web-drug-network-leads-arrest-3-german>
- Wadas, D. J. (2018). *Bitcoin and Blockchain Forensics*. (April).
- Ware, C., Mire, B., Kuniavsky, M., & Snyder, C. (n.d.). *INFORMATION VISUALIZATION*.
- Warren, A. (2017). *TOR Browser Artifacts in Windows 10*.
- Wegberg, R. van, Verburch, T., Berg, J. van den, & Staalduinen, M. van. (2017). *Alphabay Exit, Hansa-Down: Dream On? Examining the Effects of Operation Bayonet on Dream Market*. Retrieved from <https://www.tno.nl/media/10032/17-9099-factsheetbrochure-dws-05.pdf>
- Welcome to Tor Metrics. (n.d.). Retrieved September 29, 2019, from <https://metrics.torproject.org/>
- Whitcomb, D. (2012). U.S. busts global online drug market, arrests eight - Reuters. Retrieved October 16, 2019, from Reuters website: <https://www.reuters.com/article/net-us-usa-drugs-internet/u-s-busts-global-online-drug-market-arrests-eight-idUSBRE83G01Q20120417>
- Whonix. (n.d.). Retrieved October 26, 2019, from <https://www.whonix.org>
- Wieringa, J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Berlin: Springer.
- Zajác, R. (2017). Silk Road: The market beyond the reach of the state. *Information Society*, 33(1), 23–34. <https://doi.org/10.1080/01972243.2016.1248612>

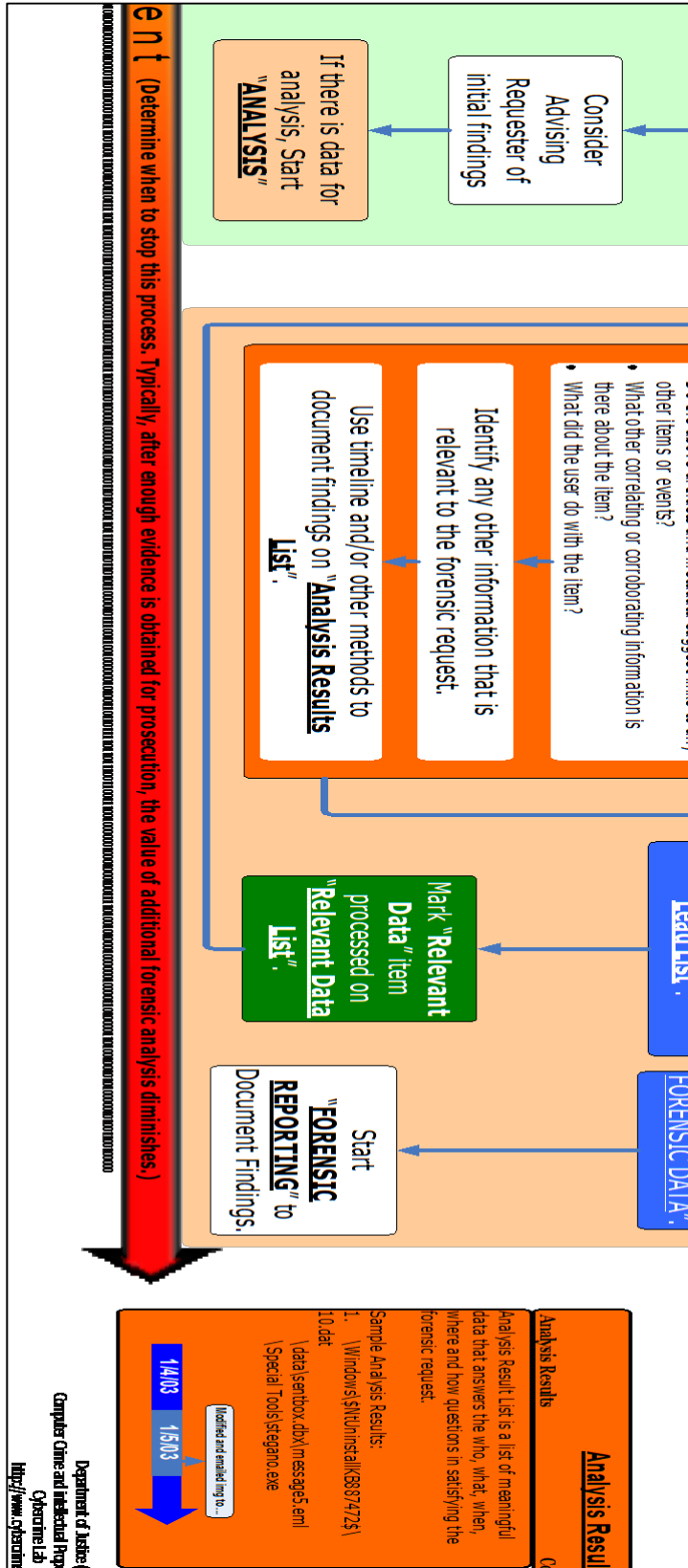


Figure 98 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)

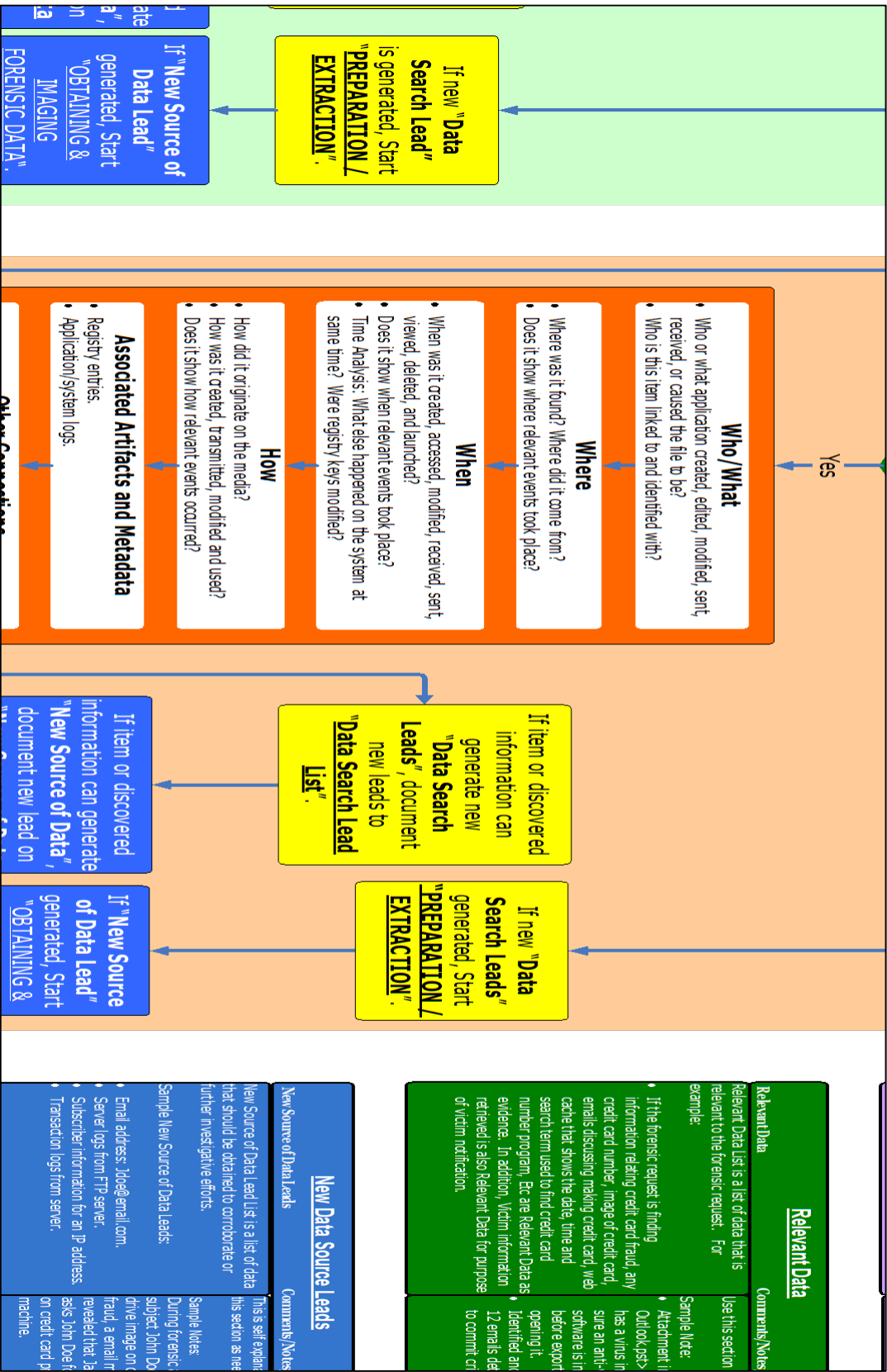


Figure 99 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)

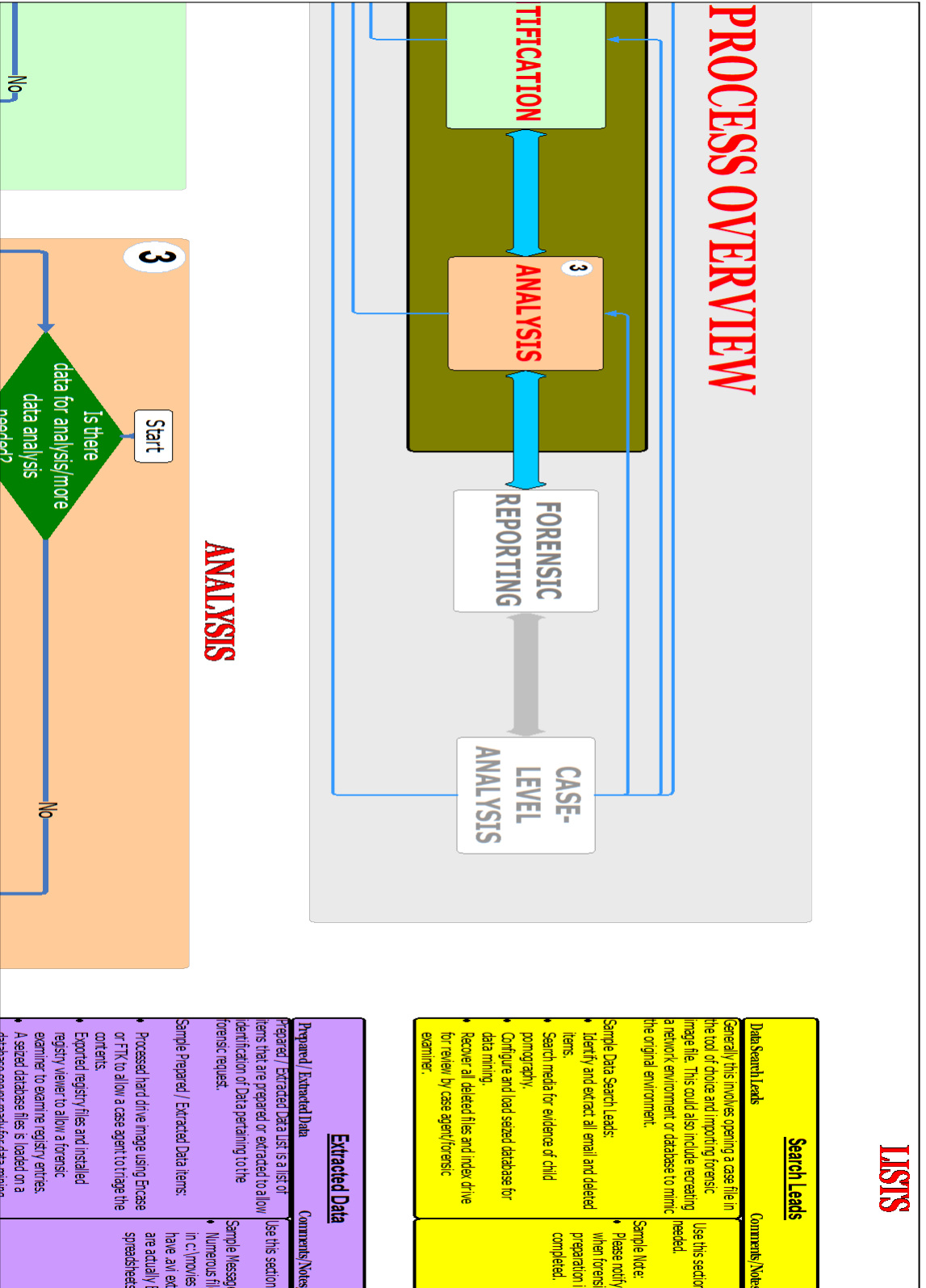


Figure 100 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)



Figures 101 & 102 - Digital Forensic Analysis Methodology Process Overview (Justice, 2008)

APPENDIX B: DIGITAL FORENSIC INVESTIGATOR CREDENTIALS

Brent Gromer - Supervisory Special Agent / Internet Crimes Against Children Commander, South Dakota Division of Criminal Investigation

24 years law enforcement experience. 11 years experience as a Digital Forensics Examiner/Investigator, 6 years ICAC Commander for State of South Dakota, Testified extensively as an expert in Digital Forensics and Investigations.

Toby Russell - Special Agent, South Dakota Division of Criminal Investigation

Employed as a law enforcement officer in the State of South Dakota since June of 1998. Attended and completed an eight-week South Dakota Basic Law Enforcement certification in South Dakota. Employed as a Patrol Officer and as a Detective with the Mitchell Police Department for 14 years and employed as a Special Agent with the South Dakota Division of Criminal Investigation (DCI) for the past eight years.

As a Special Agent with the DCI, is assigned to the South Dakota Internet Crimes Against Children (ICAC) Task Force a member of the ICAC Task Force since 2007. Over 1300 hours of general law enforcement training. Also, over 210 hours of specialized training related to conducting Internet and digital evidence related investigations and over 475 hours of specialized training in computer forensics and digital evidence forensics. Conducted hundreds of Internet and digital evidence investigations and conducted over 1300 computer forensice examinations, cell phone examinations and digital evidence examinations.

Hollie Strand - Computer Forensic Analyst, Pennington County Sheriff's Office, Rapid City, SD

Education:

2003-2005 Masters of Science - Forensic Science –Nebraska Wesleyan University

Completed 12-2005 Emphasis in Investigative Sciences

2000-2003 Masters of Science - Counseling - South Dakota State University

Completed 07-2003 Emphasis in Correctional Counseling

1995-2000 Bachelors of Science - Interdisciplinary Studies, South Dakota School of Mines and Technology

Completed 05-2000 Emphasis in Psychology/Criminology

1993-1995 Diploma, Central High School

Completed 05-1995

Employment:

2015 – Present - Computer Forensic Analyst, Pennington County Sheriff's Office – Rapid City, SD

Analyze submitted evidence; following acceptable forensic methods in evidence analysis; writing lab reports on the analysis results; testify in court hearings; provide depositions on analysis results and reports; assist with in-service training to law enforcement agencies and State's and United States' attorneys; assist with public education programs; obtain data stored in the form of magnetic or electronic coding on computer media or on media capable of being read by a computer; assist with search warrants and the seizure of evidence from crime scenes.

Kendra Russell - Special Agent, South Dakota Division of Criminal Investigation

Internet Crimes Against Children (ICAC), Investigator & Forensic Examiner (3 years);
Certified Forensic Computer Examiner (CFCE) by International Association of Computer Investigative Specialists (IACIS)

Jackson Brown - Special Agent, South Dakota Division of Criminal Investigation,

Internet Crimes Against Children Task Force

Over 9 years in law enforcement and over 1 year doing digital forensics alongside ICAC Investigations.

APPENDIX C: DARK WEB ARTIFACT FRAMEWORK EVALUATION QUESTIONS

Evaluation for Dark Web Artifact Framework

Dissertation for Arica Kulm

October 2020

Name:

Position:

Background and Credentials:

Questions for evaluation:

- What made the framework easy to follow?

- How could this framework be adapted to future platforms?

- What did you find particularly useful about the framework?

- Describe the differences of doing an investigation with or without the framework.

- What could be added or changed to make it better?