Tor Browser Forensics

Priya P Sajan[a], C. Balan[b], M.J. Devi Priya[c], A.L. Sreedeep[d]

[a]C-DAC, Thiruvanathapuram, Kerala, India. E-mail: priyasajan@cdac.in
[b]C-DAC, Thiruvanathapuram, Kerala, India. E-mail: cbalan@cdac.in
[c]C-DAC, Thiruvanathapuram, Kerala, India. E-mail: devimpriya96@gmail.com
[d]C-DAC, Thiruvanathapuram, Kerala, India. E-mail: sreedeep@cdac.in

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract: The TOR Browser is a web browser that anonymizes traffic on the web with the help of Tor network by easily hiding the identity in online platform. It uses onion routing protocol to make use of internet in possible private mode with multiple levels of encryption. These features are being misused for committing many illegal activities such as black market and cyber terrorism. TOR browser takes out all the browsing data and other traces from the network thereby making investigators job a difficult one. This research paper eyes on extracting and analyzing any possible artifacts generated by the TOR browser on local system files and memory dump.

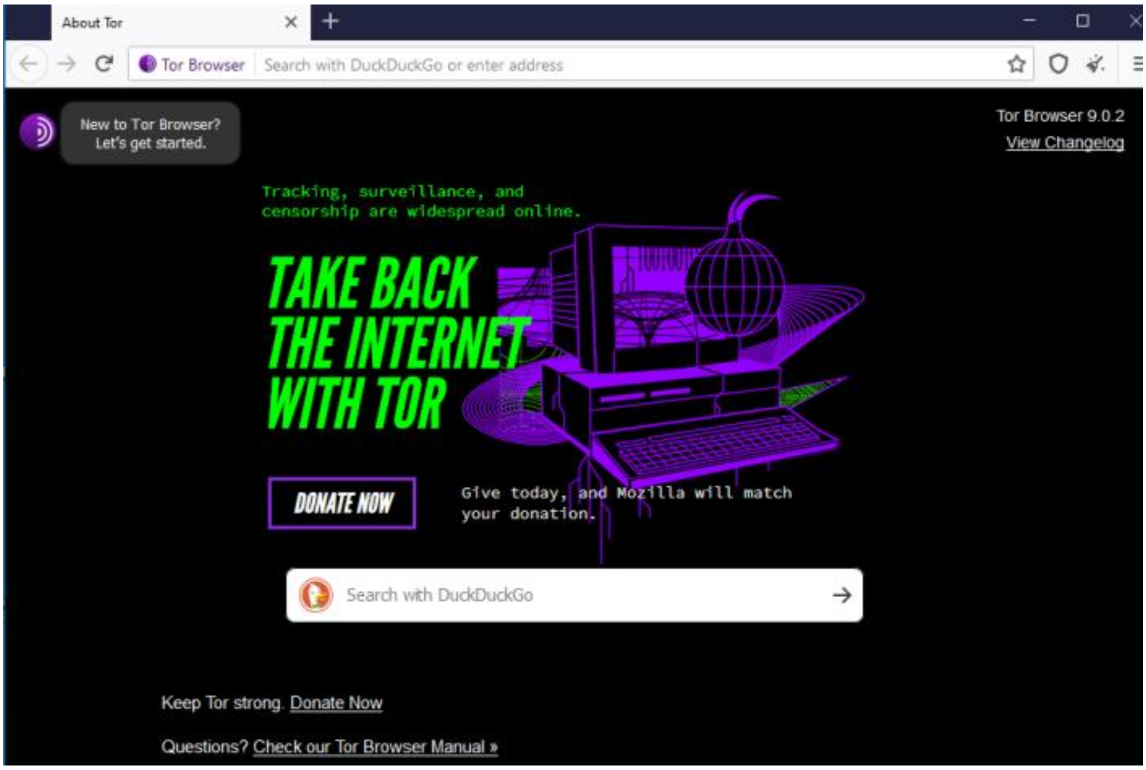
Keywords: Tor Browser, Onion Routing Protocol, Local System Files, Memory Dump.

1. Introduction

Internet can be categorized as Surface web, Dark web and Deep web. Surface web or World Wide Web comprises only 4% of the Internet. Deep web holds about 90% of Internet contents. Deep Web represents part of web that has not yet been indexed by common search engines. Remaining 6% is hosted on dark web. Dark Web accommodates a set of publicly accessible content that are hosted on websites whose IP address is hidden but to which anyone can access it as long as it knows the address. The contents in Dark Web are encrypted which makes them more to be associated with drug trafficking, cyber terrorism, blackmailing etc. Figure 1 shows different layers of dark web.

Figure 1. Different layers of dark web

Accessibility to dark web is possible only through sophisticated web browsers like TOR, I2P/ISP, Tails, Whonix, Subgraph etc. Tor Browser, as shown in Figure 2, is nothing but an extended version of Mozilla Firefox intended for anonymous and secure connection to the Internet more specifically to the dark web. TOR stands for The Onion Routing which is freely available and can be used in Windows, Linux and Mac platforms.



New to Tor Browser? Let's get started.

Tracking, surveillance, and censorship are widespread online.

TAKE BACK THE INTERNET WITH TOR

Give today, and Mozilla will match your donation.

Keep Tor strong. Donate Now

Questions? Check our Tor Browser Manual »

Figure 2. Tor Browser

Tor is an open source privacy network that allows anonymous usage of internet. Tor protects user's identity and secure network by encryption. Users that engage in digital marketplaces, digital payments, and community forums are demanding more anonymity in the way their online communications and transactions are shared. Data anonymization platforms are meeting these demands in the forms of dark wallets and underground networks. Tor is one of such underground networks that was implemented for the purpose of protecting users' identities.

But this advantage of Tor is often abused to commit illegal activities such as drug trafficking, gambling, sale of arms and violent activities etc.

2. The Onion Routing

The Tor browser is an implementation of Onion routing in Firefox browser which encrypts the data and is run by some volunteer nodes. Tor browser supports the onion sites. The components of Onion Routing include:

- Initiator: The sending application.
- Responder: The receiving application.
- Destination: The receiving end.
- Directory node: A node storing the information of other nodes in OR.
- Entry node: The first node in a chain.
- Exit node: Last node in a chain.
- Relay node: Intermediate node between entry and exit node.

The working of Tor consists of four stages:

i. Network establishment: The network topology is defined and connections between neighboring nodes are established continuously.
ii. Connection establishment: The directory sends the list of other nodes to be added in the chain or circuit. The initiator responds with the selected nodes. In this stage, the connection between the selected nodes will be created and key exchange will also happen here.
iii. Data exchange: Actual data is exchanged between the initiator and responder.
iv. Connection crack: After all the data exchange, the connection will be lost.

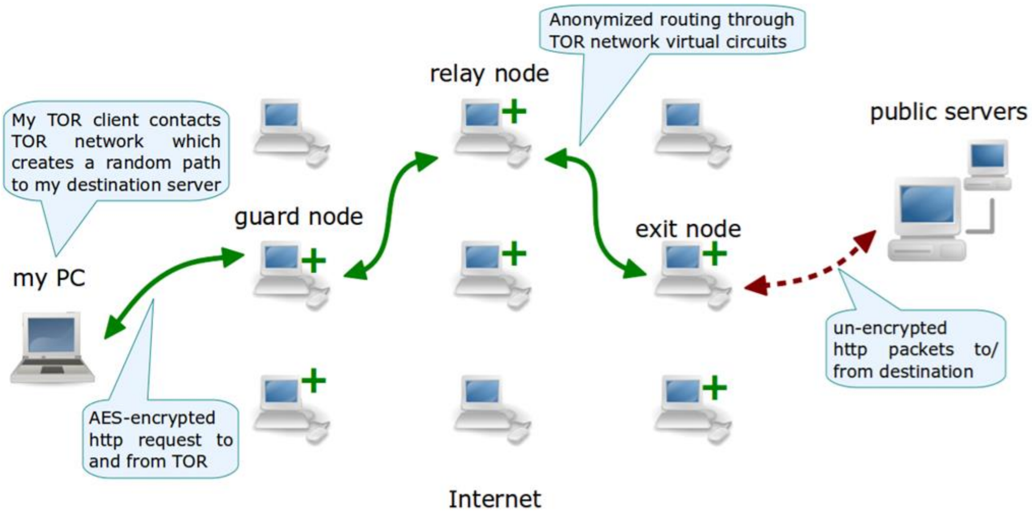

Anonymized routing through TOR network virtual circuits
relay node
My TOR client contacts TOR network which creates a random path to my destination server
public servers
guard node
exit node
my PC
un-encrypted http packets to/ from destination
AES-encrypted http request to and from TOR
Internet

Fig. 3. Working of Tor

3. Related Works

David Goldschlag et al. [1] discussed about onion routing, their network architecture and its basic functions. Roger Dingledine et al. [2] described about Tor browser, its working, advantages, design goals and how it withstands different attacks. Ohana et al. [5] proposed a new methodology for analyzing private and portable web browsing artifacts. Said et al. [6] investigated the effectiveness of the privacy mode feature in three widely used Web browsers, and outlines how to investigate when these browsers have been used to perform a criminal or illegal act. H. Chivers [7] discussed how browsing artifacts are stored in Windows file system and analyzed what all artifacts can be recovered from browsers. Ghafarian[8] explained about analysing privacy of private browsing mode through memory forensics.

K. Satvat et al. [9] evaluates security of private browsing across major browsers and from multiple angles and some of the attacks have been experimentally verified with countermeasures proposed. Filleau et al.[10] discusses on artifacts left behind due to private browsing mode. J. Oh et al. [11] experienced on advanced evidence collection and analysis of web browser activity in multiple aspects. Ming-jung et al.[12] analyses the tor browser artifacts and recovered the browsing histories from the memory dump. Kim et al.[13] focused on sgx-tor, which is a secure and practical tor anonymity network with sgx enclaves.

Ling et al.[14] conducted discovery, blocking, and traceback of malicious traffic over Tor. Matt mulr[15] used some experiments to analyze what all artifacts can be obtained and provides results for evidence trails which can be used within real-life investigations. Natalija et al. [16] discussed on anonymity of tor users demystified. Muhammad et al.[17] proposed an active attack scenario of Tor browser using some unpopular ports and described a technique that allows to increase the scalability of this type of attacks. Mattia epifani Sans eu[18] discusses about the artifacts of Tor browser and its location in windows OS.

From the literature surveys conducted, the perspective of Tor browser in the eyes of an investigator is that it clears all its browsing artifacts. This research paper proposes a system by which Tor browser artifacts can be collected effectively which can be further used for investigation.

4. Problem Statement

Tor browser possesses default security features which turn out investigation to be a laborious task. The prime issue confronted by investigators is to trace out evidences from Tor Browser as it clears all browsing data including cache and cookies. This research paper attempts to explore more about Tor browser, analyze system files and memory dump to dig out evidentiary data generated because of Tor browsing activities. Forensics analysis of Tor was initiated with the memory dump. Tools used for this purpose are

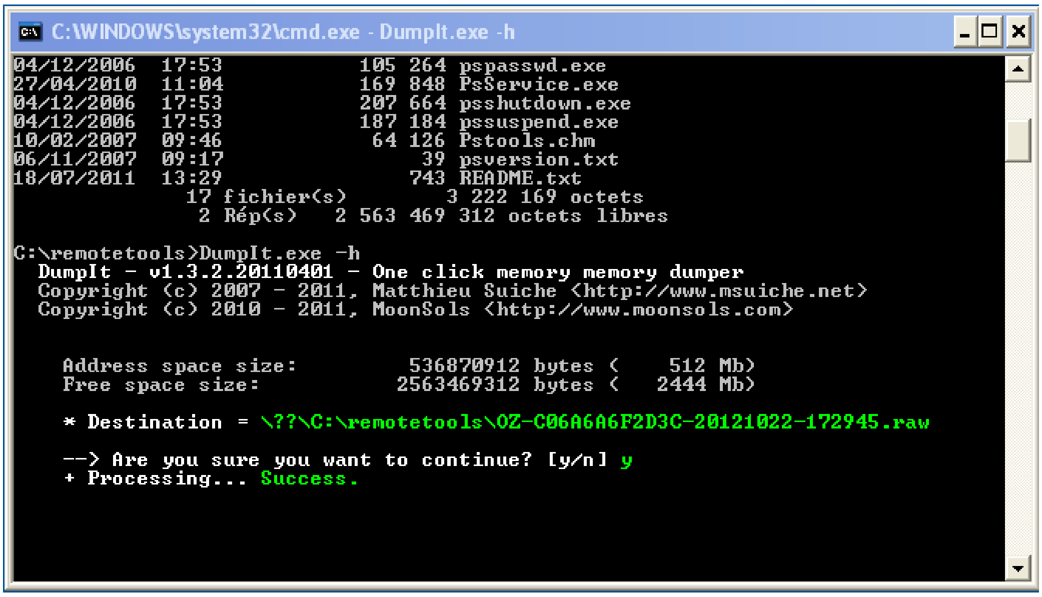
1. Dumpit : For taking the memory dump
2. Volatility : Analyzing the dump
3. Win-LiFT : Memory dump analysis
4. HxD : Hex viewer

5. Analysis of Tor Artifacts

Extraction of Tor browser related artifacts were channeled on memory dump and system local files like prefetch and browser locations.

Memory Dump Acquisition

Figure 4 shows the memory dump acquisition using the DumpIt tool. The raw memory dump is generated in the current directory itself.



```
04/12/2006  17:53            105 264 pspasswd.exe
27/04/2010  11:04            169 848 PsService.exe
04/12/2006  17:53            207 664 psshutdown.exe
04/12/2006  17:53            187 184 pssuspend.exe
10/02/2007  09:46             64 126 Pstools.chm
06/11/2007  09:17                 39 psversion.txt
18/07/2011  13:29                743 README.txt
              17 fichier(s)        3 222 169 octets
               2 Rép(s)   2 563 469 312 octets libres

C:\remotetools>DumpIt.exe -h
  DumpIt - v1.3.2.20110401 - One click memory memory dumper
  Copyright (c) 2007 - 2011, Matthieu Suiche <http://www.msuiche.net>
  Copyright (c) 2010 - 2011, MoonSols <http://www.moonsols.com>


    Address space size:         536870912 bytes (    512 Mb)
    Free space size:           2563469312 bytes (   2444 Mb)

    * Destination = \??\C:\remotetools\OZ-C06A6A6F2D3C-20121022-172945.raw

    --> Are you sure you want to continue? [y/n] y
    + Processing... Success.
```

Figure 4. Dumpit

Figure 5 shows the GUI version of volatility to load the memory dump.

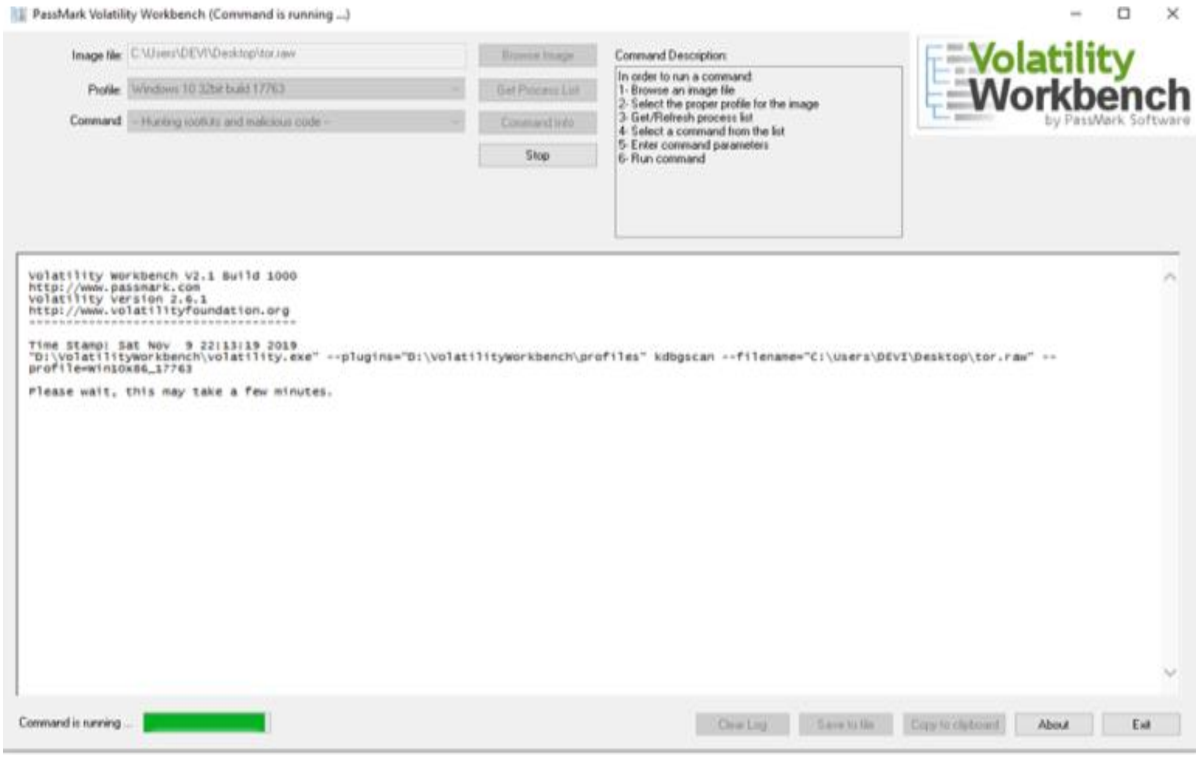


Image file: C:\Users\DEVI\Desktop\tor.raw

Profile: Windows 10 32bit build 17763

Command: -- Hunting rootkits and malicious code --

Command Description:

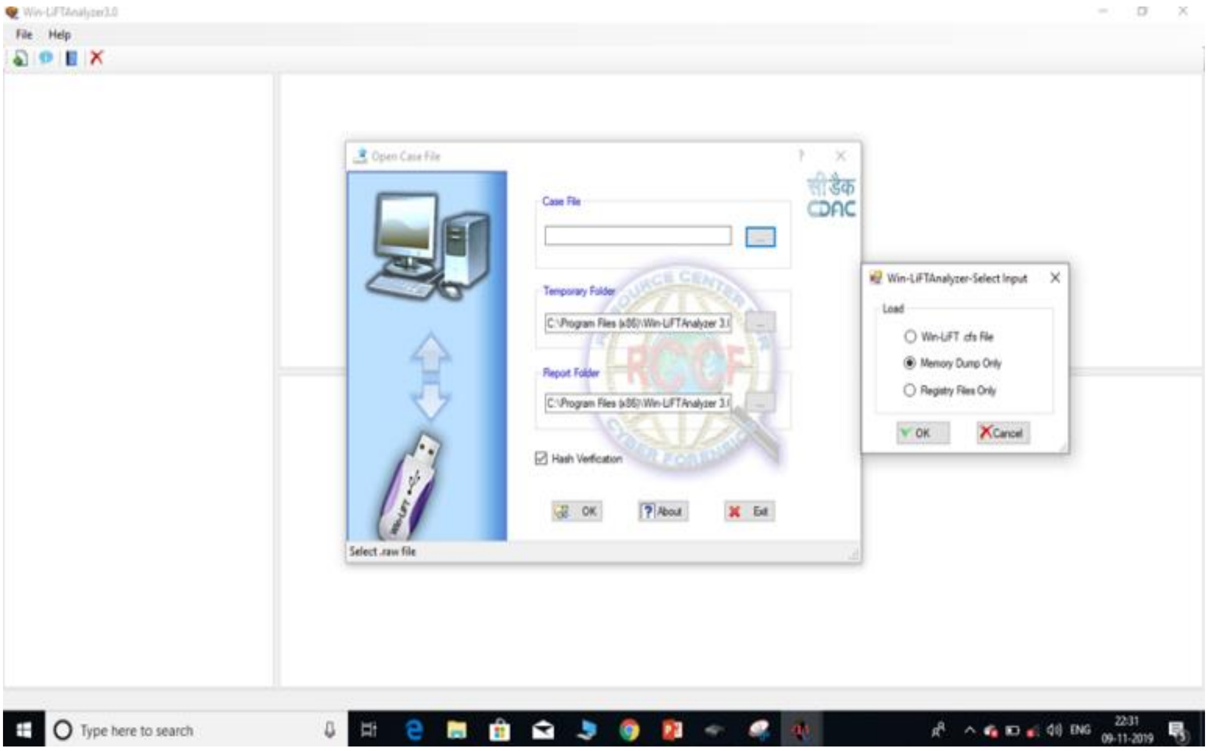
In order to run a command
1- Browse an image file
2- Select the proper profile for the image
3- Get/Refresh process list
4- Select a command from the list
5- Enter command parameters
6- Run command

```
Volatility Workbench V2.1 Build 1000
http://www.passmark.com
Volatility Version 2.6.1
http://www.volatilityfoundation.org
***********************************

Time Stamp: Sat Nov  9 22:13:19 2019
"D:\VolatilityWorkbench\volatility.exe" --plugins="D:\VolatilityWorkbench\profiles" kdbgscan --filename="C:\Users\DEVI\Desktop\tor.raw" --
profile=Win10x86_17763

Please wait, this may take a few minutes.
```

Command is running ...

Figure 5. Volatility GUI

Figure 6 shows the GUI of Win-LiFT memory dump analyzer.



Case File:

Temporary Folder: C:\Program Files (x86)\Win-LiFTAnalyzer 3.(

Report Folder: C:\Program Files (x86)\Win-LiFTAnalyzer 3.(

☑ Hash Verification

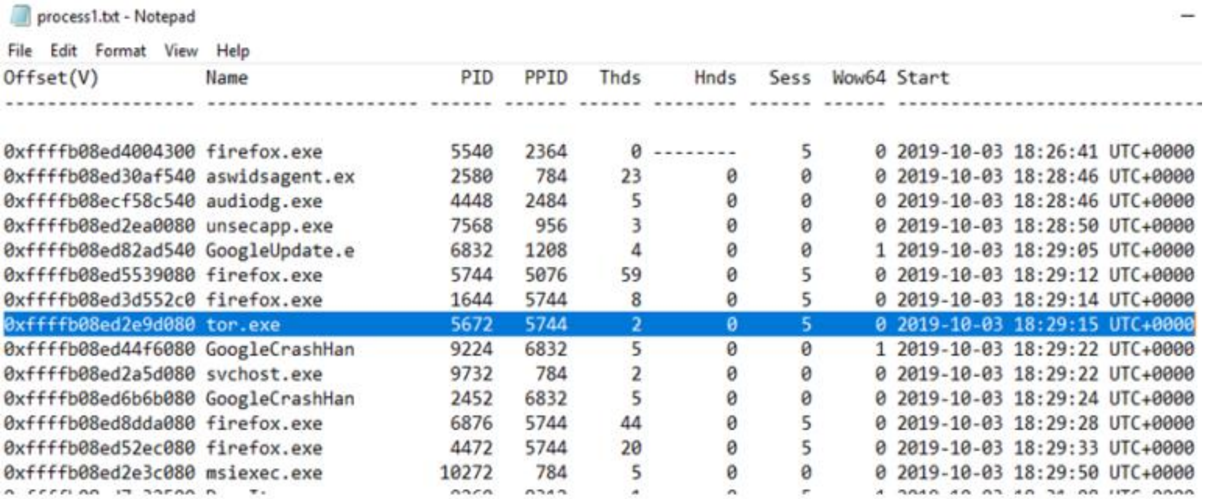
○ Win-LiFT .cfs File
● Memory Dump Only
○ Registry Files Only

Figure 6. Win-LiFT GUI

Memory Dump Analysis

Process List

Figure 7 shows the process list extracted from the memory dump using volatility which shows tor.exe has run whose process id is 5627 and parent process id 5744.

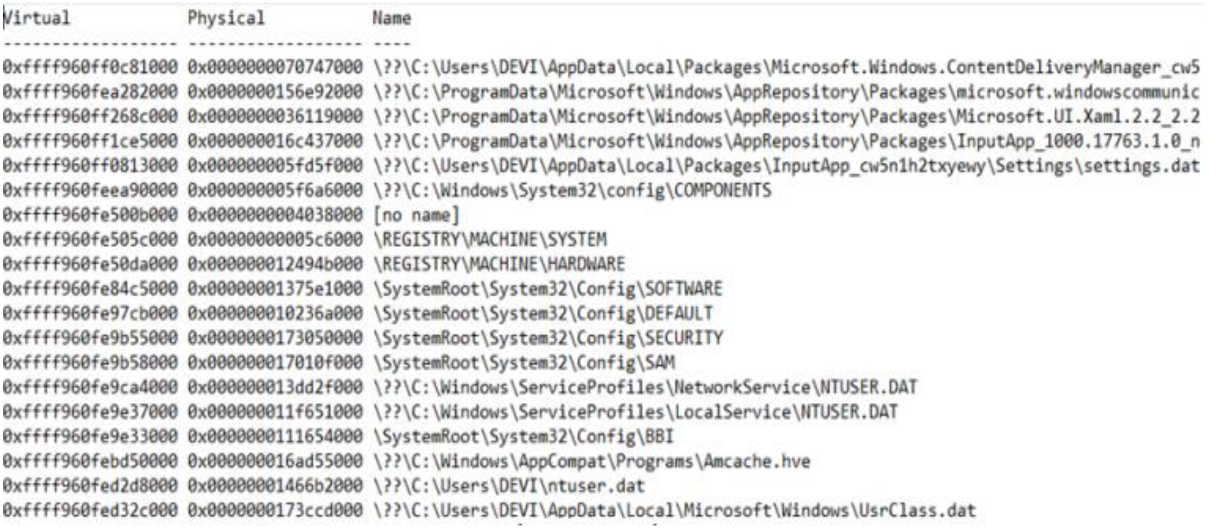


```
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------

0xffffb08ed4004300 firefox.exe            5540   2364      0 --------      5      0 2019-10-03 18:26:41 UTC+0000
0xffffb08ed30af540 aswidsagent.ex         2580    784     23        0      0      0 2019-10-03 18:28:46 UTC+0000
0xffffb08ecf58c540 audiodg.exe            4448   2484      5        0      0      0 2019-10-03 18:28:46 UTC+0000
0xffffb08ed2ea0080 unsecapp.exe           7568    956      3        0      0      0 2019-10-03 18:28:50 UTC+0000
0xffffb08ed82ad540 GoogleUpdate.e         6832   1208      4        0      0      1 2019-10-03 18:29:05 UTC+0000
0xffffb08ed5539080 firefox.exe            5744   5076     59        0      5      0 2019-10-03 18:29:12 UTC+0000
0xffffb08ed3d552c0 firefox.exe            1644   5744      8        0      5      0 2019-10-03 18:29:14 UTC+0000
0xffffb08ed2e9d080 tor.exe                5672   5744      2        0      5      0 2019-10-03 18:29:15 UTC+0000
0xffffb08ed44f6080 GoogleCrashHan         9224   6832      5        0      0      1 2019-10-03 18:29:22 UTC+0000
0xffffb08ed2a5d080 svchost.exe            9732    784      2        0      0      0 2019-10-03 18:29:22 UTC+0000
0xffffb08ed6b6b080 GoogleCrashHan         2452   6832      5        0      0      0 2019-10-03 18:29:24 UTC+0000
0xffffb08ed8dda080 firefox.exe            6876   5744     44        0      5      0 2019-10-03 18:29:28 UTC+0000
0xffffb08ed52ec080 firefox.exe            4472   5744     20        0      5      0 2019-10-03 18:29:33 UTC+0000
0xffffb08ed2e3c080 msiexec.exe           10272    784      5        0      0      0 2019-10-03 18:29:50 UTC+0000
```

Figure 7. pslist output

Registry Hives

Figure 8 shows the registry hives been extracted from the memory dump using volatility specifying to the process id 5672 which is tor.exe.

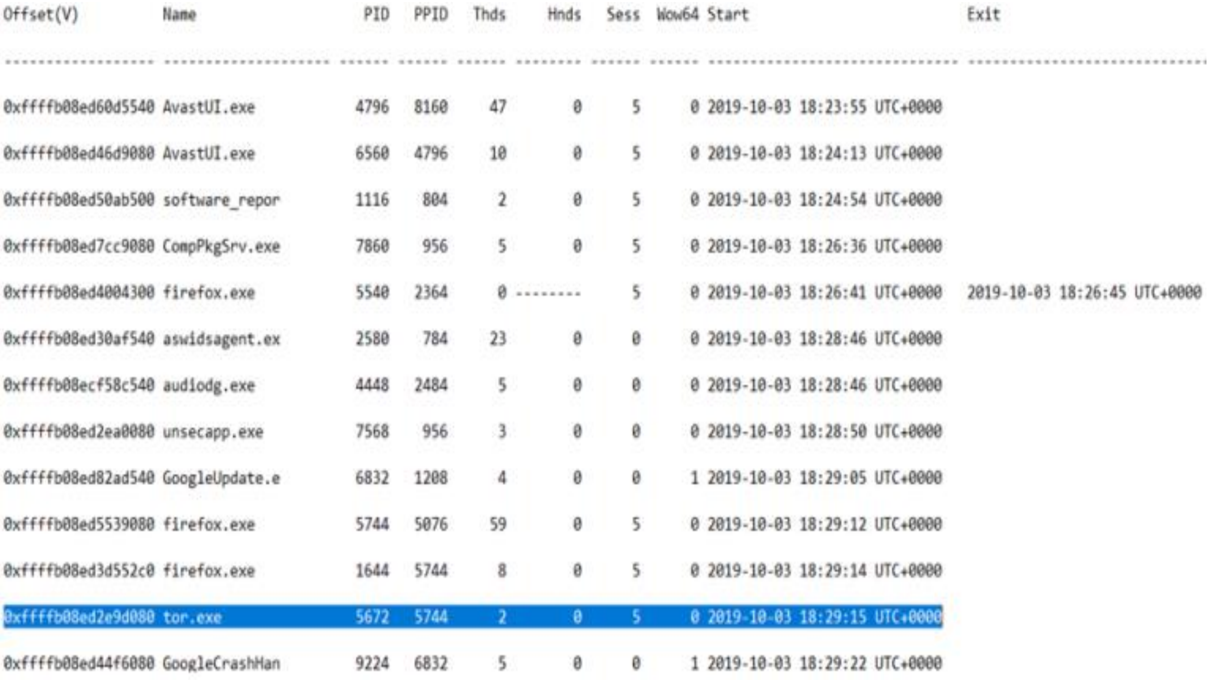


```
Virtual            Physical           Name
------------------ ------------------ ----
0xffff960ff0c81000 0x0000000070747000 \??\C:\Users\DEVI\AppData\Local\Packages\Microsoft.Windows.ContentDeliveryManager_cw5
0xffff960fea282000 0x0000000156e92000 \??\C:\ProgramData\Microsoft\Windows\AppRepository\Packages\microsoft.windowscommunic
0xffff960ff268c000 0x0000000036119000 \??\C:\ProgramData\Microsoft\Windows\AppRepository\Packages\Microsoft.UI.Xaml.2.2_2.2
0xffff960ff1ce5000 0x000000016c437000 \??\C:\ProgramData\Microsoft\Windows\AppRepository\Packages\InputApp_1000.17763.1.0_n
0xffff960ff0813000 0x000000005fd5f000 \??\C:\Users\DEVI\AppData\Local\Packages\InputApp_cw5n1h2txyewy\Settings\settings.dat
0xffff960feea90000 0x000000005f6a6000 \??\C:\Windows\System32\config\COMPONENTS
0xffff960fe500b000 0x0000000004038000 [no name]
0xffff960fe505c000 0x00000000005c6000 \REGISTRY\MACHINE\SYSTEM
0xffff960fe50da000 0x000000012494b000 \REGISTRY\MACHINE\HARDWARE
0xffff960fe84c5000 0x00000001375e1000 \SystemRoot\System32\Config\SOFTWARE
0xffff960fe97cb000 0x000000010236a000 \SystemRoot\System32\Config\DEFAULT
0xffff960fe9b55000 0x0000000173050000 \SystemRoot\System32\Config\SECURITY
0xffff960fe9b58000 0x000000017010f000 \SystemRoot\System32\Config\SAM
0xffff960fe9ca4000 0x000000013dd2f000 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0xffff960fe9e37000 0x000000011f651000 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0xffff960fe9e33000 0x0000000111654000 \SystemRoot\System32\Config\BBI
0xffff960febd50000 0x000000016ad55000 \??\C:\Windows\AppCompat\Programs\Amcache.hve
0xffff960fed2d8000 0x00000001466b2000 \??\C:\Users\DEVI\ntuser.dat
0xffff960fed32c000 0x0000000173ccd000 \??\C:\Users\DEVI\AppData\Local\Microsoft\Windows\UsrClass.dat
```

Figure 8. Registry hives

Threads

Figure 9 shows the number of threads run by tor.exe. Totally two threads were extracted by volatility using the command “D:\VolatilityWorkbench\volatility.exe”–plugins=”D:\VolatilityWorkbench\profiles” pslist filename =”C:\Users\username\Desktop\tor.raw” –profile=Win10x64 17763 –kdbg=0xf807606ac5e0



```
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------

0xffffb08ed60d5540 AvastUI.exe            4796   8160     47        0      5      0 2019-10-03 18:23:55 UTC+0000

0xffffb08ed46d9080 AvastUI.exe            6560   4796     10        0      5      0 2019-10-03 18:24:13 UTC+0000

0xffffb08ed50ab500 software_repor         1116    804      2        0      5      0 2019-10-03 18:24:54 UTC+0000

0xffffb08ed7cc9080 CompPkgSrv.exe         7860    956      5        0      5      0 2019-10-03 18:26:36 UTC+0000

0xffffb08ed4004300 firefox.exe            5540   2364      0 --------      5      0 2019-10-03 18:26:41 UTC+0000   2019-10-03 18:26:45 UTC+0000

0xffffb08ed30af540 aswidsagent.ex         2580    784     23        0      0      0 2019-10-03 18:28:46 UTC+0000

0xffffb08ecf58c540 audiodg.exe            4448   2484      5        0      0      0 2019-10-03 18:28:46 UTC+0000

0xffffb08ed2ea0080 unsecapp.exe           7568    956      3        0      0      0 2019-10-03 18:28:50 UTC+0000

0xffffb08ed82ad540 GoogleUpdate.e         6832   1208      4        0      0      1 2019-10-03 18:29:05 UTC+0000

0xffffb08ed5539080 firefox.exe            5744   5076     59        0      5      0 2019-10-03 18:29:12 UTC+0000

0xffffb08ed3d552c0 firefox.exe            1644   5744      8        0      5      0 2019-10-03 18:29:14 UTC+0000

0xffffb08ed2e9d080 tor.exe                5672   5744      2        0      5      0 2019-10-03 18:29:15 UTC+0000

0xffffb08ed44f6080 GoogleCrashHan         9224   6832      5        0      0      1 2019-10-03 18:29:22 UTC+0000
```

Figure 9. Threads

Handles

Figure 10 shows the resources used by the process 5672(tor.exe). This handle list is extracted by the volatility (GUI) from the memory dump.

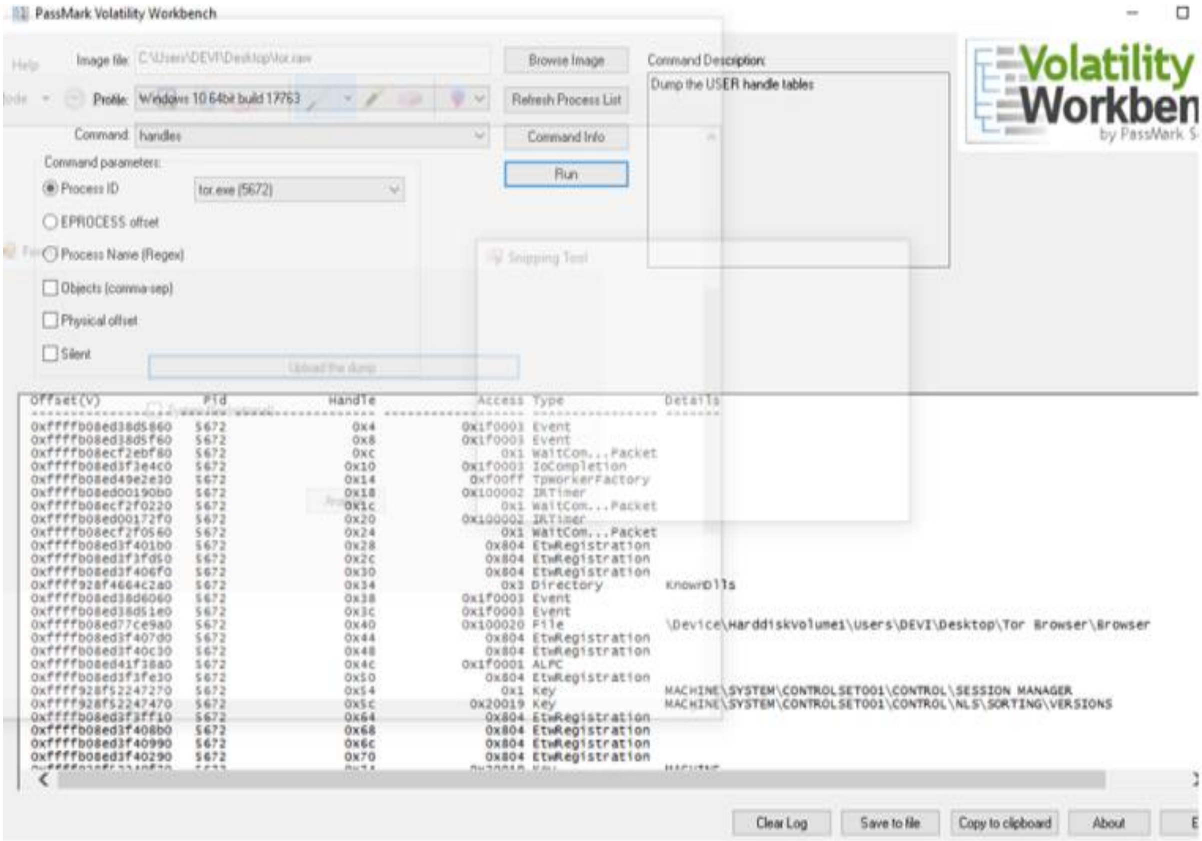


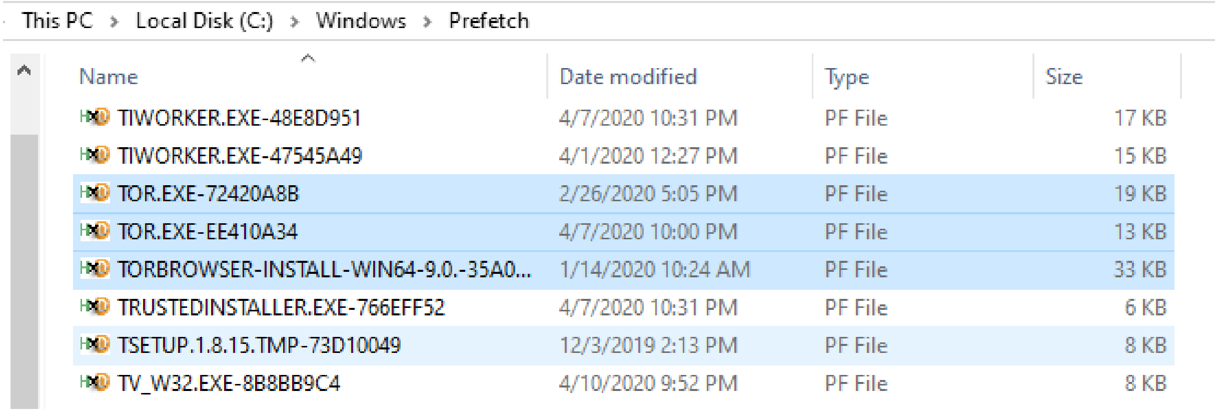
Image file: C:\Users\DEVI\Desktop\tor.raw
Profile: Windows 10 64bit build 17763
Command: handles
Command Description: Dump the USER handle tables
Command parameters:
● Process ID tor.exe (5672)
○ EPROCESS offset
○ Process Name (Regex)
□ Objects (comma-sep)
□ Physical offset
□ Silent

```
Offset(V)               Pid          Handle         Access Type             Details
------------------ ------ ------------------ ---------- ---------------- -------
0xffffb08ed38d5860   5672              0x4   0x1f0003 Event
0xffffb08ed38d5f60   5672              0x8   0x1f0003 Event
0xffffb08ecf2ebf80   5672              0xc        0x1 WaitCom...Packet
0xffffb08ed3f3e4c0   5672             0x10   0x1f0003 IoCompletion
0xffffb08ed49e2e30   5672             0x14    0xf00ff TpWorkerFactory
0xffffb08ed00190b0   5672             0x18   0x100002 IRTimer
0xffffb08ecf2f0220   5672             0x1c        0x1 WaitCom...Packet
0xffffb08ed00172f0   5672             0x20   0x100002 IRTimer
0xffffb08ecf2f0560   5672             0x24        0x1 WaitCom...Packet
0xffffb08ed3f401b0   5672             0x28      0x804 EtwRegistration
0xffffb08ed3f3fd50   5672             0x2c      0x804 EtwRegistration
0xffffb08ed3f406f0   5672             0x30      0x804 EtwRegistration
0xffff928f4664c2a0   5672             0x34        0x3 Directory         KnownDlls
0xffffb08ed38d6060   5672             0x38   0x1f0003 Event
0xffffb08ed38d51e0   5672             0x3c   0x1f0003 Event
0xffffb08ed77ce9a0   5672             0x40   0x100020 File              \Device\HarddiskVolume1\Users\DEVI\Desktop\Tor Browser\Browser
0xffffb08ed3f407d0   5672             0x44      0x804 EtwRegistration
0xffffb08ed3f40c30   5672             0x48      0x804 EtwRegistration
0xffffb08ed41f38a0   5672             0x4c   0x1f0001 ALPC
0xffffb08ed3f3fe30   5672             0x50      0x804 EtwRegistration
0xffff928f52247270   5672             0x54        0x1 Key               MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER
0xffff928f52247470   5672             0x5c    0x20019 Key               MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\VERSIONS
0xffffb08ed3f3ff10   5672             0x64      0x804 EtwRegistration
0xffffb08ed3f408b0   5672             0x68      0x804 EtwRegistration
0xffffb08ed3f40990   5672             0x6c      0x804 EtwRegistration
0xffffb08ed3f40290   5672             0x70      0x804 EtwRegistration
```

Figure 10. Handles

Dlls and Prefetch Files

Dynamic Link Library (dll) contains a library of functions that can be accessed by a windows application. When a program is launched, links to the necessary .dll files are created. If a static link is created, the .dll files will be in use as long as the program is active. If a dynamic link is created, the .dll files will only be used when needed. These dlls can be found from prefetch file. Figure 11 shows the prefetch files of Tor.

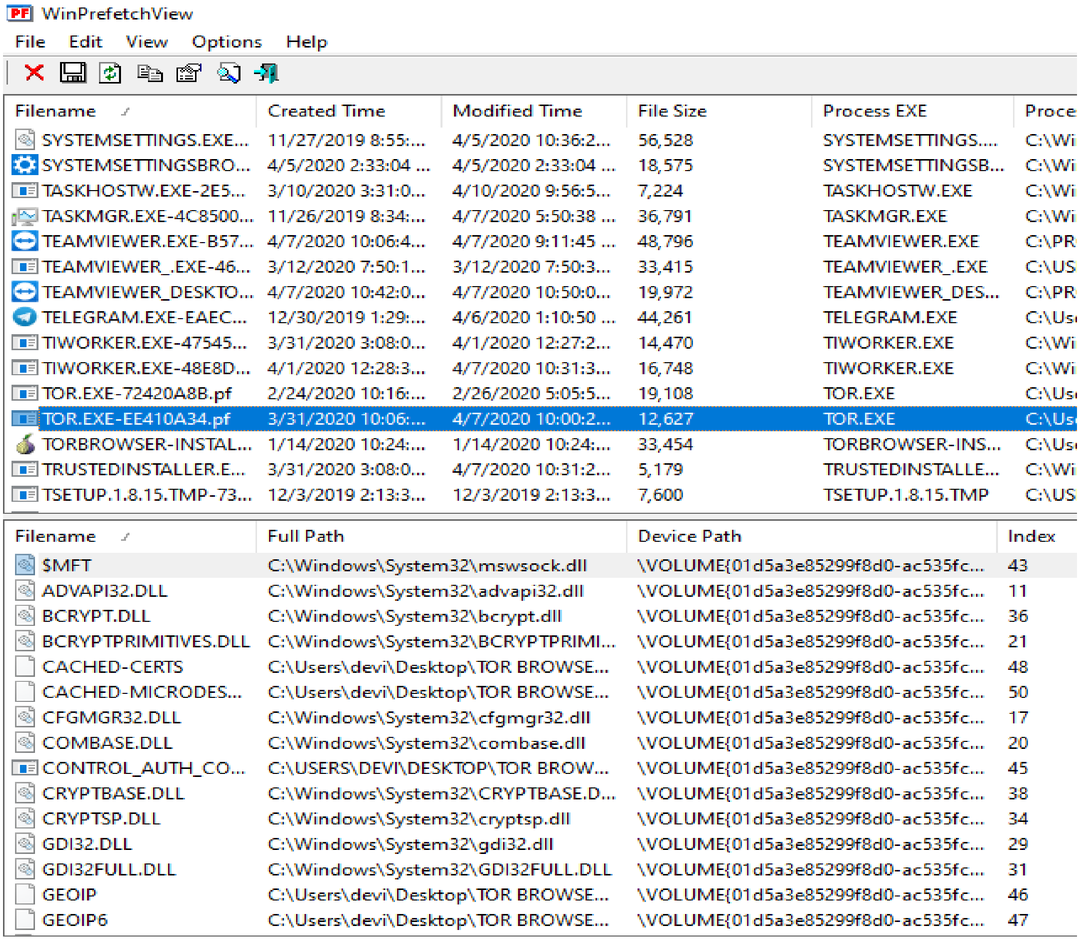


This PC › Local Disk (C:) › Windows › Prefetch

Name	Date modified	Type	Size
TIWORKER.EXE-48E8D951	4/7/2020 10:31 PM	PF File	17 KB
TIWORKER.EXE-47545A49	4/1/2020 12:27 PM	PF File	15 KB
TOR.EXE-72420A8B	2/26/2020 5:05 PM	PF File	19 KB
TOR.EXE-EE410A34	4/7/2020 10:00 PM	PF File	13 KB
TORBROWSER-INSTALL-WIN64-9.0.-35A0...	1/14/2020 10:24 AM	PF File	33 KB
TRUSTEDINSTALLER.EXE-766EFF52	4/7/2020 10:31 PM	PF File	6 KB
TSETUP.1.8.15.TMP-73D10049	12/3/2019 2:13 PM	PF File	8 KB
TV_W32.EXE-8B8BB9C4	4/10/2020 9:52 PM	PF File	8 KB

Figure 11. Prefetch Files

Location of prefetch file is C:\Windows\Prefetch. The tool named winprefetchview is used to view the prefetch files. Figure 12 shows the dlls extracted from the Tor's prefetch file.



Filename	Created Time	Modified Time	File Size	Process EXE	Proce
SYSTEMSETTINGS.EXE...	11/27/2019 8:55:...	4/5/2020 10:36:2...	56,528	SYSTEMSETTINGS....	C:\Wi
SYSTEMSETTINGSBRO...	4/5/2020 2:33:04 ...	4/5/2020 2:33:04 ...	18,575	SYSTEMSETTINGSB...	C:\Wi
TASKHOSTW.EXE-2E5...	3/10/2020 3:31:0...	4/10/2020 9:56:5...	7,224	TASKHOSTW.EXE	C:\Wi
TASKMGR.EXE-4C8500...	11/26/2019 8:34:...	4/7/2020 5:50:38 ...	36,791	TASKMGR.EXE	C:\Wi
TEAMVIEWER.EXE-B57...	4/7/2020 10:06:4...	4/7/2020 9:11:45 ...	48,796	TEAMVIEWER.EXE	C:\PR
TEAMVIEWER_.EXE-46...	3/12/2020 7:50:1...	3/12/2020 7:50:3...	33,415	TEAMVIEWER_.EXE	C:\US
TEAMVIEWER_DESKTO...	4/7/2020 10:42:0...	4/7/2020 10:50:0...	19,972	TEAMVIEWER_DES...	C:\PR
TELEGRAM.EXE-EAEC...	12/30/2019 1:29:...	4/6/2020 1:10:50 ...	44,261	TELEGRAM.EXE	C:\Us
TIWORKER.EXE-47545...	3/31/2020 3:08:0...	4/1/2020 12:27:2...	14,470	TIWORKER.EXE	C:\Wi
TIWORKER.EXE-48E8D...	4/1/2020 12:28:3...	4/7/2020 10:31:3...	16,748	TIWORKER.EXE	C:\Wi
TOR.EXE-72420A8B.pf	2/24/2020 10:16:...	2/26/2020 5:05:5...	19,108	TOR.EXE	C:\Us
TOR.EXE-EE410A34.pf	3/31/2020 10:06:...	4/7/2020 10:00:2...	12,627	TOR.EXE	C:\Us
TORBROWSER-INSTAL...	1/14/2020 10:24:...	1/14/2020 10:24:...	33,454	TORBROWSER-INS...	C:\Us
TRUSTEDINSTALLER.E...	3/31/2020 3:08:0...	4/7/2020 10:31:2...	5,179	TRUSTEDINSTALLE...	C:\Wi
TSETUP.1.8.15.TMP-73...	12/3/2019 2:13:3...	12/3/2019 2:13:3...	7,600	TSETUP.1.8.15.TMP	C:\US

Filename	Full Path	Device Path	Index
$MFT	C:\Windows\System32\mswsock.dll	\VOLUME{01d5a3e85299f8d0-ac535fc...	43
ADVAPI32.DLL	C:\Windows\System32\advapi32.dll	\VOLUME{01d5a3e85299f8d0-ac535fc...	11
BCRYPT.DLL	C:\Windows\System32\bcrypt.dll	\VOLUME{01d5a3e85299f8d0-ac535fc...	36
BCRYPTPRIMITIVES.DLL	C:\Windows\System32\BCRYPTPRIMI...	\VOLUME{01d5a3e85299f8d0-ac535fc...	21
CACHED-CERTS	C:\Users\devi\Desktop\TOR BROWSE...	\VOLUME{01d5a3e85299f8d0-ac535fc...	48
CACHED-MICRODES...	C:\Users\devi\Desktop\TOR BROWSE...	\VOLUME{01d5a3e85299f8d0-ac535fc...	50
CFGMGR32.DLL	C:\Windows\System32\cfgmgr32.dll	\VOLUME{01d5a3e85299f8d0-ac535fc...	17
COMBASE.DLL	C:\Windows\System32\combase.dll	\VOLUME{01d5a3e85299f8d0-ac535fc...	20
CONTROL_AUTH_CO...	C:\USERS\DEVI\DESKTOP\TOR BROW...	\VOLUME{01d5a3e85299f8d0-ac535fc...	45
CRYPTBASE.DLL	C:\Windows\System32\CRYPTBASE.D...	\VOLUME{01d5a3e85299f8d0-ac535fc...	38
CRYPTSP.DLL	C:\Windows\System32\cryptsp.dll	\VOLUME{01d5a3e85299f8d0-ac535fc...	34
GDI32.DLL	C:\Windows\System32\gdi32.dll	\VOLUME{01d5a3e85299f8d0-ac535fc...	29
GDI32FULL.DLL	C:\Windows\System32\GDI32FULL.DLL	\VOLUME{01d5a3e85299f8d0-ac535fc...	31
GEOIP	C:\Users\devi\Desktop\TOR BROWSE...	\VOLUME{01d5a3e85299f8d0-ac535fc...	46
GEOIP6	C:\Users\devi\Desktop\TOR BROWSE...	\VOLUME{01d5a3e85299f8d0-ac535fc...	47

Figure 12. List of Dlls

Places.sqlite Files

These are Firefox files holding record of visited websites, bookmarks, keywords etc. Being an extended version of Firefox, Tor has this file in the profile folder C:\Users\USERNAME\Desktop\TorBrowser\Data\Browser\profile.default. The tool SQLite viewer has been used to recover bookmarks and frequently visited sites even after uninstalling the application. Figure 13shows the screen shot of places.sqlite files viewed in SQLite viewer.

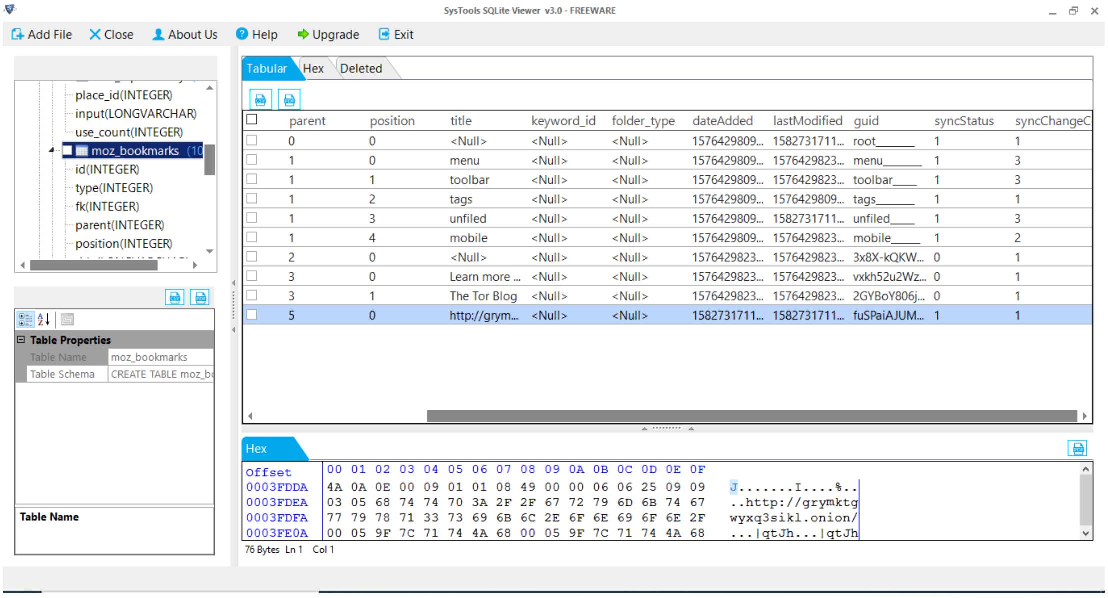


Table Name	moz_bookmarks
Table Schema	CREATE TABLE moz_b

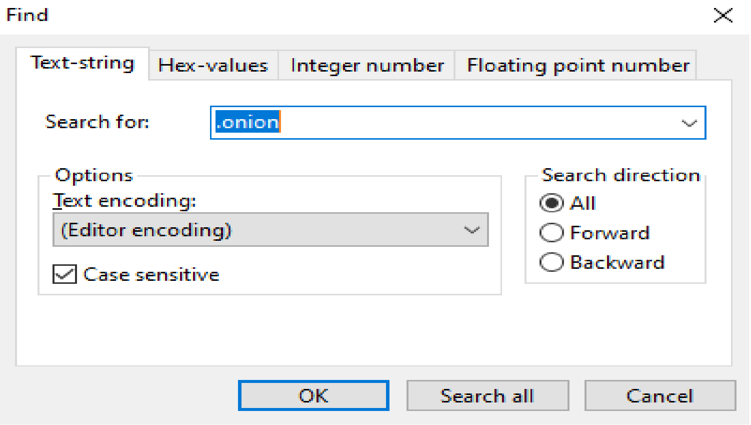
□	parent	position	title	keyword_id	folder_type	dateAdded	lastModified	guid	syncStatus	syncChangeC
□	0	0	<Null>	<Null>	<Null>	1576429809...	1582731711...	root______	1	1
□	1	0	menu	<Null>	<Null>	1576429809...	1576429823...	menu______	1	3
□	1	1	toolbar	<Null>	<Null>	1576429809...	1576429823...	toolbar____	1	3
□	1	2	tags	<Null>	<Null>	1576429809...	1576429809...	tags______	1	1
□	1	3	unfiled	<Null>	<Null>	1576429809...	1582731711...	unfiled____	1	3
□	1	4	mobile	<Null>	<Null>	1576429809...	1576429823...	mobile_____	1	2
□	2	0	<Null>	<Null>	<Null>	1576429823...	1576429823...	3x8X-kQKW...	0	1
□	3	0	Learn more ...	<Null>	<Null>	1576429823...	1576429823...	vxkh52u2Wz...	0	1
□	3	1	The Tor Blog	<Null>	<Null>	1576429823...	1576429823...	2GYBoY806j...	0	1
□	5	0	http://grym...	<Null>	<Null>	1582731711...	1582731711...	fuSPaiAJUM...	1	1

```
Offset    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0003FDDA  4A 0A 0E 00 09 01 01 08 49 00 00 06 06 25 09 09   J.......I....%..
0003FDEA  03 05 68 74 74 70 3A 2F 2F 67 72 79 6D 6B 74 67   ..http://grymktg
0003FDFA  77 79 78 71 33 73 69 6B 6C 2E 6F 6E 69 6F 6E 2F   wyxq3sikl.onion/
0003FE0A  00 05 9F 7C 71 74 4A 68 00 05 9F 7C 71 74 4A 68   ...|qtJh...|qtJh
76 Bytes  Ln 1   Col 1
```

Figure 13. places.sqlite viewed in SQLite viewer

Visited Websites

The Dump file's Hex view is viewed using HxD. Using keyword search (Tor or .onion) visited websites can be viewed. Figure 14 and Figure 15 shows the keyword search and its output respectively.



Find

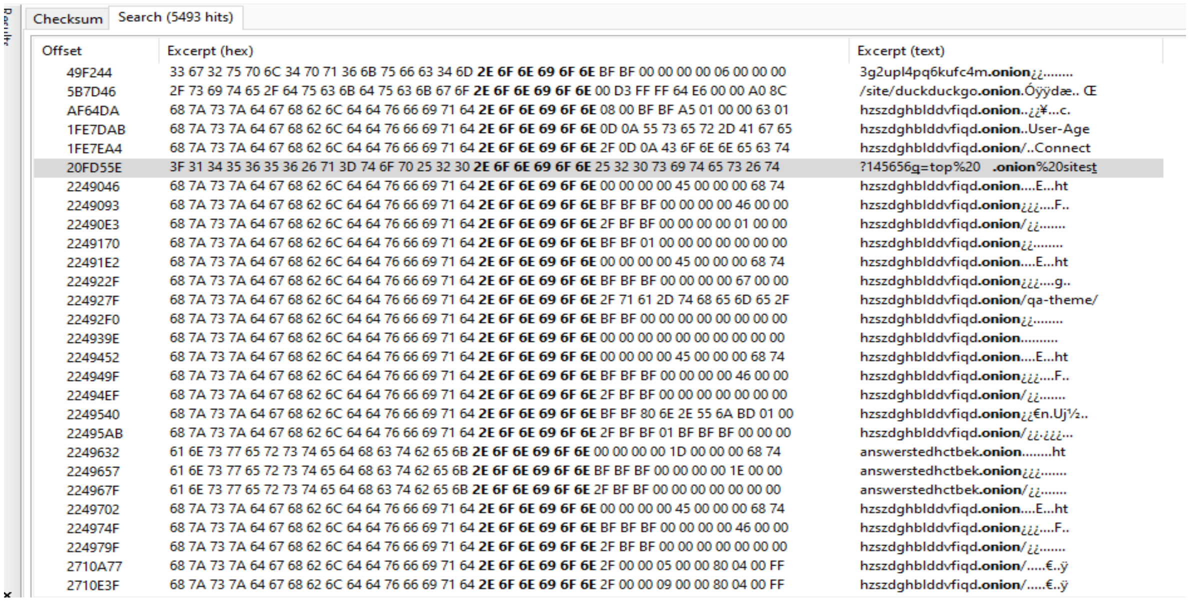
Text-string | Hex-values | Integer number | Floating point number

Search for: .onion

Options
Text encoding: (Editor encoding)
☑ Case sensitive

Search direction
● All
○ Forward
○ Backward

Figure 14. Keyword Search



Offset	Excerpt (hex)	Excerpt (text)
49F244	33 67 32 75 70 6C 34 70 71 36 6B 75 66 63 34 6D **2E 6F 6E 69 6F 6E** BF BF 00 00 00 00 06 00 00 00	3g2upl4pq6kufc4m**.onion**¿¿........
5B7D46	2F 73 69 74 65 2F 64 75 63 6B 64 75 63 6B 67 6F **2E 6F 6E 69 6F 6E** 00 D3 FF FF 64 E6 00 00 A0 8C	/site/duckduckgo**.onion**.Óÿÿdæ.. Œ
AF64DA	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 08 00 BF BF A5 01 00 00 63 01	hzszdghblddvfiqd**.onion**..¿¿¥...c.
1FE7DAB	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 0D 0A 55 73 65 72 2D 41 67 65	hzszdghblddvfiqd**.onion**..User-Age
1FE7EA4	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 2F 0D 0A 43 6F 6E 6E 65 63 74	hzszdghblddvfiqd**.onion**/..Connect
20FD55E	3F 31 34 35 36 35 36 26 71 3D 74 6F 70 25 32 30 **2E 6F 6E 69 6F 6E** 25 32 30 73 69 74 65 73 26 74	?145656q=top%20 **.onion**%20sitest
2249046	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 00 00 00 00 45 00 00 00 68 74	hzszdghblddvfiqd**.onion**....E...ht
2249093	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** BF BF BF 00 00 00 00 46 00 00	hzszdghblddvfiqd**.onion**¿¿¿....F..
22490E3	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 2F BF BF 00 00 00 00 01 00 00	hzszdghblddvfiqd**.onion**/¿¿.......
2249170	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** BF BF 01 00 00 00 00 00 00 00	hzszdghblddvfiqd**.onion**¿¿........
22491E2	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 00 00 00 00 45 00 00 00 68 74	hzszdghblddvfiqd**.onion**....E...ht
224922F	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** BF BF BF 00 00 00 00 67 00 00	hzszdghblddvfiqd**.onion**¿¿¿....g..
224927F	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 2F 71 61 2D 74 68 65 6D 65 2F	hzszdghblddvfiqd**.onion**/qa-theme/
22492F0	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** BF BF 00 00 00 00 00 00 00 00	hzszdghblddvfiqd**.onion**¿¿........
224939E	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 00 00 00 00 00 00 00 00 00 00	hzszdghblddvfiqd**.onion**..........
2249452	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 00 00 00 00 45 00 00 00 68 74	hzszdghblddvfiqd**.onion**....E...ht
224949F	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** BF BF BF 00 00 00 00 46 00 00	hzszdghblddvfiqd**.onion**¿¿¿....F..
22494EF	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 2F BF BF 00 00 00 00 00 00 00	hzszdghblddvfiqd**.onion**/¿¿.......
2249540	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** BF BF 80 6E 2E 55 6A BD 01 00	hzszdghblddvfiqd**.onion**¿¿€n.Uj½..
22495AB	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 2F BF BF 01 BF BF BF 00 00 00	hzszdghblddvfiqd**.onion**/¿¿.¿¿¿...
2249632	61 6E 73 77 65 72 73 74 65 64 68 63 74 62 65 6B **2E 6F 6E 69 6F 6E** 00 00 00 00 1D 00 00 00 68 74	answerstedhctbek**.onion**........ht
2249657	61 6E 73 77 65 72 73 74 65 64 68 63 74 62 65 6B **2E 6F 6E 69 6F 6E** BF BF BF 00 00 00 00 1E 00 00	answerstedhctbek**.onion**¿¿¿.......
224967F	61 6E 73 77 65 72 73 74 65 64 68 63 74 62 65 6B **2E 6F 6E 69 6F 6E** 2F BF BF 00 00 00 00 00 00 00	answerstedhctbek**.onion**/¿¿.......
2249702	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 00 00 00 00 45 00 00 00 68 74	hzszdghblddvfiqd**.onion**....E...ht
224974F	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** BF BF BF 00 00 00 00 46 00 00	hzszdghblddvfiqd**.onion**¿¿¿....F..
224979F	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 2F BF BF 00 00 00 00 00 00 00	hzszdghblddvfiqd**.onion**/¿¿.......
2710A77	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 2F 00 00 05 00 00 80 04 00 FF	hzszdghblddvfiqd**.onion**/.....€..ÿ
2710E3F	68 7A 73 7A 64 67 68 62 6C 64 64 76 66 69 71 64 **2E 6F 6E 69 6F 6E** 2F 00 00 09 00 00 80 04 00 FF	hzszdghblddvfiqd**.onion**/.....€..ÿ

Figure 15. Output for keyword search

The result points out the visited .onion websites and searches which contain the keyword .onion.

6. Conclusion and Future Scope

Tor browser maintains anonymity in internet, keep privacy and protect data with double layer encryption. As the security features of Tor browser increases, it becomes more fragile for committing illegal activities. Hence analysis of Tor artifacts from memory dump and system local files are expected to play a crucial role in the investigation point of view. Future work eyes on extending the analysis of Tor evidences from network packets captured.

References

1. D.M. Goldschlag, M.G. Reed, P.F. Syverson, "Hiding routing information", *International workshop on information hiding*, Springer, 1996, pp. 137– 150.
2. R. Dingledine, N. Mathewson, P. Syverson, "Tor: the second-generation onion router" *Technical report, Naval Research Lab*, Washington, dc, 2004.
3. https:// www.torproject.org
4. https://metrics.torproject.org

5. D.J. Ohana, N. Shashidhar, “Do private and portable web browsers leave incriminating evidence? a forensic analysis of residual artifacts from private and portable web browsing sessions”, *Journal of Information Security* 2013 (2013) 6.
6. H. Said, N. Al mutawa, I. Al Awadhi, M. Guimaraes, “Forensic analysis of private browsing artifacts”, 2011 *International Conference on Innovations in Information Trechnology (iit), ieee,* 2011, pp. 197–202.
7. H. Chivers, private browsing, “A window of forensic opportunity”, *digital invest. 11*(2014) 20–29. 35.
    A. Ghafarian, S. Seno, “Analysis of privacy of private browsing mode through memory forensics”, *International Journal of Computer Applications 132*(2015).
8. K. Satvat, M. Forshaw, F. Hao, E. Toreini, *“On the privacy of private browsing – a forensic approach, data privacy management and autonomous spontaneous security”,* Springer, 2014, pp. 380–389.
9. J. Filleau, M. Zizyte, *“What private browsing leaves behind”,* Springer, 2014, pp. 380–389.
10. J. Oh, S. Lee, “Advanced evidence collection and analysis of web browser activity”, *Digital Invest.* 8 (2011) s62–s70.
11. Ming-jung Chiu huang, Yu-lun wan, “Tor browser forensics in exploring invisible evidence”, 2018 *IEEE International Conference on Systems, Man, and Cybernetics*.
12. Seongmin kim, Juhyeng han, Jaehyeong ha, Taesoo kim, and Dongsu han, “sgx-tor: a secure and practical tor anonymity network with sgx enclaves”, *IEEE/ACM Transactions on Networking,* Vol. 26, No. 5, October 2018.
13. Zhen ling, Junzhou Luo, Kui Wu, Wei Yu, and Xinwen Fu, “TORWARD: discovery, blocking, and traceback of malicious traffic over tor”, *IEEE Transactions on Information Forensics and Security,* Vol. 10, No. 12, December 2015.
14. Matt Mulr, Petra Lelmich, William J Buchanan, “A forensic audit of the tor browser bundle”, *Article in Digital Investigation·* March 2019 Research Gate.
15. Natalija Vlajic, Pooria Adani, Ethan Nguyen, “Anonymity of tor users demystified”, *International Conference on Computational Science and Computational Intelligence* 2017.
16. Muhammad Aliyu Sulaiman, Sami Zhioua, “Attacking tor through unpopular ports”, *IEEE 33rd International Conference on Distributed Computing Systems Workshops* 2013.
17. Mattia epifani Sans eu, *“Tor forensics on windows os”,* Digital Forensics Summit 1 Prague, 5 October 2014.