Fight crime. Unravel incidents... one byte at a time.

Copyright SANS Institute
Author Retains Full Rights

Interested in learning more?
Check out the list of upcoming events offering "Advanced Incident Response, Threat Hunting, and Digital Forensics (FOR508)" at http://digital-forensics.sans.org/events/
Abstract

The Tor network is a popular, encrypted, worldwide, anonymizing virtual network in existence since 2002 and is used by all facets of society such as privacy advocates, journalists, governments, and criminals. This paper will provide a forensic analysis of the Tor Browser version 5 client on a Windows 10 host for an individual or group interested in remnants left by the software. This paper will utilize various free and commercial tools to provide a detailed analysis of filesystem artifacts as well as a comparison between pre- and post- connection to the Tor network using memory analysis.
1. Introduction

The Tor project has been a worldwide collaborative effort for over twenty years with roots beginning in the United States Government. The Office of Naval Research funded a project in 1995 (Syverson, 2005b) with the goal of identifying a method ”not specifically to provide anonymous communication, but, to separate identification from routing ” (Syverson, 2005a, para. 5). The work was termed Onion Routing with the initial development milestone called “generation 0” (Syverson, 2005b, para. 4). The Onion Routing’s initial public presentation was at the First Information Hiding Workshop on May 31, 1996 (Syverson, 2005b). In 1997 improvements moved development from generation 0 to generation 1 and Defense Advanced Research Projects Agency (DARPA) became a funding source (Syverson, 2005b). Generation 2 of the code is what is commonly named Tor (Syverson, 2005b). Tor is an acronym for “The onion routing” even though it does not follow acronym conventions in capitalization ("Tor FAQ," n.d.). In 2002 generation 2 was born as a fork of code “originally produced by Matej Pfajfar at Cambridge University for his undergraduate final-year project” (Syverson, 2005b, para. 24). Code development moved in 2003 to torproject.org and the Tor network was fully deployed (Syverson, 2005b).

Onion routing is a simple concept in that the end user, or *initiator* of network traffic, encrypts traffic with multiple layers. A layer of encryption exists for each hop inside the Tor network as denoted on the left in Figure 1. As the encrypted traffic moves through the Tor network, each node removes one layer of encryption, analogous to removing an onion layer. At the last Tor network node, the final layer of encryption is removed and the traffic proceeds out onto the Internet. The first node going into the Tor network, node A in Figure 1, is called the entry relay.

Figure 1. A packet is encrypted once for each hop in the Tor Network. At each Tor node, the outermost layer of encryption is removed.
or entry guard (“Tor FAQ,” n.d.). The last node, node C in Figure 1, is called the exit relay or exit node (“Tor FAQ,” n.d.).

The Tor Browser Bundle, currently called Tor Browser, attempts to achieve the simplest method for connecting users to Tor (Peery, 2014). In terms of software development, Tor dates back to at least a minimal version number of 1.0 as of March of 2008 (Phobos, 2008). The current version of 6.0.8 is based upon Mozilla’s Firefox Extended Release Support (ESR) and includes “Torbutton, TorLauncher, NoScript, and HTTPS-Everywhere” (“What is Tor Browser,” n.d.).

This paper will begin by giving an overview of steps taken in performing a Tor Browser installation and subsequent connection to the Tor network. The Tor Browser will be installed on a Windows 10 Virtual Machine (VM). Once the relevant image snapshots have been created, an in-depth look at the filesystem artifacts will be shown. Subsequently, an analysis of the in-memory artifacts will be performed. Lastly, the paper will be an overview of the Tor Browser’s anti-forensics approach will be provided.

## 2. Forensic Approach

The disk images that will be used in the analysis are snapshots that were created using VMWare Fusion version 8.5.3. The Operating System (OS) used was a clean 64-bit installation of Windows 8.1 Pro, subsequently upgraded to a 64-bit Windows 10 Pro. The OS was patched to kernel version 10.0.10586.17. The user warren seen throughout this paper was a user account with administrative privileges.

To make the analysis easier, a full clone of the VM was made to have a clean starting point with the snapshots. The first snapshot of the clone was made immediately after the cloning was performed. The second snapshot was taken was after the Tor Browser software was installed. A third snapshot was made while a connection to the Tor network was active.

The computer used to perform the analysis was a Windows 7 Home Edition SIFT workstation provided in the SANS FOR408 class disc version 6.0, dated September 2012. The commercial X-Ways Forensics version 17.3 SR 4 was used along with open source tools that will be mentioned throughout this paper. The version of the Tor Browser installed was version 5.0_en-US.

The reason an older Tor Browser version was used for analysis in this paper is in following the Ethical Tor Research Guidelines whose general principle is that “experimentation

Aron Warren, aronwarren@gmail.com
does not justify endangering people” (Ailanthus, 2015). By using an older version, it was this author’s hope that any vulnerabilities have been mitigated allowing for end-user’s time to upgrade to newer versions. The analysis done here was considered a basic template with the potential to be performed with the most current Tor Browser version.

3. Installation of Regshot

A basic Windows forensic step is to obtain the registry settings. The registry before and after installation of the Tor Browser software can yield an understanding of how the software installation changes the system. Regshot, shown in Figure 2, is open source software that performs registry snapshots (regshot, 2016). HAL9000 says that we “simply create the 1st shot, install the software or run the program you want to watch, and then press 2nd shot” (Hal9000, 2016).

4. Installation of Tor Browser

Older versions of the Tor Browser are difficult to find but can be obtained from https://archive.torproject.org/tor-package-archive/torbrowser/5.0/. Pretty Good Privacy (PGP) was used, as shown in Figure 3 below, to verify the software signatures and ensure the version downloaded is a verified package (“How to verify signatures for packages,” n.d.).

```
warren$ gpg --keyserver pool.sks-keyserver.net --recv-keys 0x4E2C6E8793298290
```

```
gpg: requesting key 93298290 from hkp server pool.sks-keyserver.net

gpg: key 93298290: public key "Tor Browser Developers (signing key) <torbrowser@torproject.org>" imported

gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model

gpg: depth: 0 valid: 2 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 2u

gpg: next trustdb check due at 2021-06-03

gpg: Total number processed: 1

gpg: imported: 1 (RSA: 1)
```
Figure 3. PGP verifies a file’s content against a signature signed by a key. This ensures that the file has not been altered.

Since the package has been verified, as shown with the words “Good signature” in Figure 3, the next step of installation was as easy as double-clicking the executable. For this installation, the user’s desktop was chosen to make it easy to find forensically.

Regshot was used again and the second shot was taken. In comparing the differences between the first and second shots only one related entry was found, and it showed the location the installation binary was launched from was a VMware shared folder, as shown in Figure 4:

Figure 4. The installation path of the Tor Browser installer.

5. Filesystem Artifacts
5.1 Carving in W-Ways

To perform the filesystem forensics, X-Ways was used to carve the files. X-Ways is compatible with VMDK files that are split into smaller file sizes. All that was needed to be done is “Create New Case” file from the “Case Data” window’s File Menu, as shown in Figure 5.

Next, in the Case Data window selecting “Add Image” from the File menu was done to add the VMDK files. The next step was to select each VM snapshot’s VMDK file.

As shown in Figure 6, each of the VM snapshots was added in with disk image “Virtual Disk-cl1-000003” which is the snapshot that is the post-Tor Browser installation snapshot when the Tor Browser was running and connected to the Tor network.

5.2 Prefetch

One of the artifacts to look for on the filesystem is a prefetch file to indicate the software’s installation location. To find prefetch files, one must traverse to %SystemRoot%\Prefetch. The contents are shown in Figure 7 below ("Prefetch," 2016).
Figure 7. Prefetch files able to be carved from the VM snapshot taken after installation of the Tor Browser.

Right clicking on the prefetch file and selecting Recover/Copy from the context menu presents the option to export the prefetch file. Once the file is recovered decompression must be performed to view the contents of the prefetch file (Picasso, 2015a).

Windows 7 SIFT does not have the OS files for Windows 10 prefetch decompression necessary to run Picasso’s regripper scripts (2015b). The scripts must instead be launched on a Windows 10 machine. The command used is shown in Figure 8:

```
$ python hotoloti/sas/w10pfdecomp.py TORBROWSER-INSTALL-5.0_EN-US.-767F1BA4.pf TORBROWSER-INSTALL-5.0_EN-US.-767F1BA4-uncompressed
```

Figure 8. The command line arguments necessary to decompress the prefetch file on a Windows 10 host.

Running hexdump on the resultant uncompressed file produced the following first five lines shown in Figure 9:

```
00000000 1e 00 00 00 53 43 43 41 11 00 00 00 e2 ad 00 00 00000010 54 00 04 f0 00 52 00 42 00 52 00 4f 00 57 00 53 00 f0 00 57 00 53 00 00000020 45 00 52 00 2d 00 49 00 4e 00 53 00 54 00 41 00 00000030 4c 00 04 00 02 00 03 00 02 00 30 00 30 00 3f 00 45 00 00000040 4e 00 2d 00 55 00 53 00 53 00 02 00 00 00 00 00 a4 1b 7f 76 00000050 00 00 00 00 30 01 00 00 53 00 00 00 00 90 0b 00 00
```

Aron Warren, aronwarren@gmail.com
At offset 0x0000000 for four bytes and at offset 0x0000004 for four bytes the output shown in Figure 10 are the properties of the prefetch file ("Windows Prefetch File Format," 2016).

Starting at location 0x0000010 for 60 bytes was the program identifier ("Windows Prefetch File Format," 2016) as shown in Figure 11:

Using a hex to ASCII converter of the hex in Figure 11 yielded the string shown in Figure 12:

The actual executable’s location was not calculated nor considered in scope for this paper but could be obtained using Metz’s instructions (2016). This information will instead be determined another way in a following section.

5.3 Hives and RegRipper

To analyze the system and user registry hives, which contain artifacts about system and user activity, RegRipper was used ("Registry Hives," n.d.). After installing RegRipper, the next step was to use X-Ways to carve out the System, SAM, and Security hives,
both before and after installation of the Tor Browser. Running RegRipper on each hive (see Figure 13 above) then doing a Unix diff against both files, yielded that the SAM, System, Software, and Security hives had no relevant changes to indicate installation of the software.

One artifact from the SAM hive useful in correlating artifacts was the user warren’s Security IDentifier (SID) shown in Figure 14:

\[ S-1-5-21-445630921-2900216602-2167668200-1001 \]

Figure 14. warren’s SID.

In analyzing the warren’s NTUSER.dat the relevant entries are shown in Figure 15:

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fri Oct 30</td>
<td>07:18:23</td>
<td><code>\\vmware-host\Shared Folders\shared-with-vm\torbrowser-install-5.0_en-US.exe</code></td>
</tr>
<tr>
<td>Sun Oct 9</td>
<td>15:14:09</td>
<td><code>\\vmware-host\Shared Folders\shared-with-vm\torbrowser-install-5.0_en-US.exe (2)</code></td>
</tr>
</tbody>
</table>

Figure 15. NTUSER.dat entries pointing to the installer’s location.

This indicated where the Tor Browser install executable was located. An analysis of the UsrClass.dat yielded nothing pertaining to the Tor Browser.

6. Memory Artifacts

At this point, a switch from filesystem artifacts is made to look at memory artifacts. Memory artifacts were obtained using the VM snapshot memory file and the Volatility suite. To perform that analysis the most current version of Volatility was obtained via the preferred installation instructions (iMHLv2, 2016). The commit ID of 33134a97 was the last committed change of the win10 profile for kernel 10586. Instead of using that kernel version, this example used commit ID of b3cde88 giving access to 0x64 tech preview 14968. The kernel version in that commit is a higher revision than the VM’s version of 10586. Additionally, the volatility community plugins, used to gain access to additional tools (“Volatilityfoundation / community,” 2017) were used at commit ID: 29b07e7

An example command line used for gathering the process tree using the pstree module is shown in figure 16.
Aron Warren, aronwarren@gmail.com

mbp-2:volatility warren$ python2.7 vol.py     
--plugins=~warren/Documents/community-plugins 
--profile=Win10x64     
--filename=/Volumes/WD2TB/clone-of-Win8toWin10//Copy\ of\ all\ Full\ Clone\ Windows\ 8\ contents/Full\ Clone\ of\ Windows\ 8\ x86-Win8ToWin10-Snapshot2.vmem pstree

Figure 16. Volatility command line options used to gather artifacts from memory images.

The following subsections show the artifacts relevant to the Tor Browser analysis captured from various modules:

6.1 getsids

Figure 17 depicts the SIDs associated with the two Process IDs (PIDs) used by the Tor Browser:

<table>
<thead>
<tr>
<th>Process</th>
<th>SIDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>firefox.exe</td>
<td>S-1-5-21-445630921-2900216602-2167668200-1001 (warren)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-21-445630921-2900216602-2167668200-513  (Domain Users)</td>
</tr>
<tr>
<td></td>
<td>S-1-1-0 (Everyone)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-114 (Local Account (Member of Administrators))</td>
</tr>
<tr>
<td></td>
<td>S-1-5-11 (Authenticated Users)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-15 (This Organization)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-113 (Local Account)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-10-199061 (Logon Session)</td>
</tr>
<tr>
<td></td>
<td>S-1-2-0 (Local (Users with the ability to log in locally))</td>
</tr>
<tr>
<td></td>
<td>S-1-5-64-10 (NTLM Authentication)</td>
</tr>
<tr>
<td></td>
<td>S-1-16-8192 (Medium Mandatory Level)</td>
</tr>
<tr>
<td>tor.exe</td>
<td>S-1-5-21-445630921-2900216602-2167668200-1001 (warren)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-21-445630921-2900216602-2167668200-513 (Domain Users)</td>
</tr>
<tr>
<td></td>
<td>S-1-1-0 (Everyone)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-114 (Local Account (Member of Administrators))</td>
</tr>
<tr>
<td></td>
<td>S-1-5-10-199061 (Logon Session)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-32-544 (Administrators)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-32-545 (Users)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-4 (Interactive)</td>
</tr>
<tr>
<td></td>
<td>S-1-2-1 (Console Logon (Users who are logged onto the physical console))</td>
</tr>
<tr>
<td></td>
<td>S-1-5-11 (Authenticated Users)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-15 (This Organization)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-113 (Local Account)</td>
</tr>
<tr>
<td></td>
<td>S-1-5-50-199061 (Logon Session)</td>
</tr>
<tr>
<td></td>
<td>S-1-2-0 (Local (Users with the ability to log in locally))</td>
</tr>
<tr>
<td></td>
<td>S-1-5-64-10 (NTLM Authentication)</td>
</tr>
<tr>
<td></td>
<td>S-1-16-8192 (Medium Mandatory Level)</td>
</tr>
</tbody>
</table>

Figure 17. The SIDS used by both processes.
6.2 dlllist

Following, in Figure 18, are the DLL entries for the two known PIDs which also happen to show the installation location:

```
************************************************************************
firefox.exe pid:   4684
Command line : "C:\Users\warren\Desktop\Tor Browser\Browser\firefox.exe"
Base                Size      LoadCount  Path
------------------  ------------- ---------- ----
0x00000000600000  0x55000   0x0        C:\Users\warren\Desktop\Tor Browser\Browser\firefox.exe
************************************************************************
tor.exe pid:   4476
Command line :
Base                Size          LoadCount  Path
------------------  ------------- ---------- ----
0x00000000880000  0x1fc000      0x0        C:\Users\warren\Desktop\Tor Browser\TorBrowser\Tor\tor.exe
************************************************************************
```

Figure 18. The DLLs used in each process.

6.3 ports open from netscan

Shown in Figure 19 are the open network sockets related to the Tor activity. The Internet Protocol (IP) addresses that resolved to a name containing the word tor are represented as torserver below. The other IP addresses not directly attributable to Tor addresses are represented by unknown. The Tor SOCKS proxy is located on 9050/tcp and the Tor control port is located on 9051/tcp. Ports 9150 and 9151 are used by the Tor Browser Bundle (Crenshaw, 2014).

```
<table>
<thead>
<tr>
<th>Offset(P)</th>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
<th>Pid</th>
<th>Owner</th>
<th>Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xe000177bd570</td>
<td>TCPv4</td>
<td>127.0.0.1:9151</td>
<td>127.0.0.1:49745</td>
<td>ESTABLISHED</td>
<td>-</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0xe00018e99d10</td>
<td>TCPv4</td>
<td>172.16.30.182:49775</td>
<td>unknown4:80</td>
<td>ESTABLISHED</td>
<td>-</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0xe00018992010</td>
<td>TCPv4</td>
<td>127.0.0.1:49700</td>
<td>127.0.0.1:9151</td>
<td>ESTABLISHED</td>
<td>-</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0xe00018c36600</td>
<td>TCPv4</td>
<td>127.0.0.1:9151</td>
<td>127.0.0.1:49701</td>
<td>ESTABLISHED</td>
<td>-</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0xe00018d05010</td>
<td>TCPv4</td>
<td>172.16.30.182:49756</td>
<td>unknown5:443</td>
<td>ESTABLISHED</td>
<td>-</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0xe00018e9ed10</td>
<td>TCPv4</td>
<td>127.0.0.1:49703</td>
<td>127.0.0.1:9151</td>
<td>ESTABLISHED</td>
<td>-</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0xe00019047b40</td>
<td>TCPv4</td>
<td>127.0.0.1:49754</td>
<td>127.0.0.1:9150</td>
<td>ESTABLISHED</td>
<td>-</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0xe000196054c0</td>
<td>TCPv4</td>
<td>127.0.0.1:49702</td>
<td>127.0.0.1:49701</td>
<td>ESTABLISHED</td>
<td>-</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
```
Tor Browser Artifacts in Windows 10

Figure 19. Ports listening on either the Tor Browser ports or the servers involved in the Tor browsing.

6.4 envars

Figure 20 shows the environment variables used by both processes. Environment variables are useful in that it will indicate where the process may look for information tailored for each system. The Path variable is particularly interesting in that for Tor, the path is the installation directory, whereas for Firefox it is not.

<table>
<thead>
<tr>
<th>PID</th>
<th>Process</th>
<th>Block</th>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>ALLUSERSPROFILE</td>
<td>C:\ProgramData</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>APPDATA</td>
<td>C:\Users\warren\AppData\Roaming</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>CommonProgramFiles</td>
<td>C:\Program Files\Common Files</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>CommonProgramFiles (x86)</td>
<td>C:\Program Files (x86)\Common Files</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>COMSPEC</td>
<td>C:\WINDOWS\system32\cmd.exe</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>FPS_BROWSER_APP_PROFILE_STRING</td>
<td>Internet Explorer</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>FPS_BROWSER_PROFILE_STRING</td>
<td>Default</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>HOMEDRIVE</td>
<td>WIN8-2</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>Homedrive</td>
<td>C:\WINDOWS\system32\cmd.exe</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>LOCALAPPDATA</td>
<td>Users\warren\AppData\Local</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>LOGONSERVER</td>
<td>\WIN8-2</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>NUMBER_OF_PROCESSORS</td>
<td>1</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>OS</td>
<td>Windows_NT</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>Path</td>
<td>C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\Win\PowerShell\v1.0</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>PATHEXT</td>
<td>.COM\EXE;BAT;CMD;VBS;VBE;JS;JSE;WSF;WSH;MSC</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>PROCESSOR_ARCHITECTURE</td>
<td>AMD64</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>PROCESSOR_IDENTIFIER</td>
<td>Intel64 Family 6 Model 42 Stepping 7, GenuineIntel</td>
</tr>
<tr>
<td>4684</td>
<td>firefox.exe</td>
<td>0x00000000001d0860</td>
<td>PROCESSOR_LEVEL</td>
<td>6</td>
</tr>
</tbody>
</table>

Aron Warren, aronwarren@gmail.com
Tor Browser Artifacts in Windows 10

Aron Warren, aronwarren@gmail.com
Figure 20. The PATH environment variables where the process will look for executables. Also shown are other variables which may be of interest to the investigator.

6.5 cmdline

The command line for both processes assist in identifying the installation location, except for Tor in this case, is shown in Figure 21:

```
Mozilla Firefox (64-bit) – Tor Browser (32-bit)

firefox.exe pid:   4684
Command line : "C:\Users\warren\Desktop\Tor Browser\Browser\firefox.exe"

Tor Browser (32-bit)

tor.exe pid: 4476
Command line : 
```

Figure 21. The command line used by each process. Note that the tor executable is not displaying any command line.

6.6 dumpfiles

Unfortunately, due to time constraints it was not feasible to dump the individual files from the memory image to see what their contents were. The contents of the individual files possibly would show more artifacts about what was being browsed at the time of the snapshot. For completeness, they are listed in Appendix A.

6.7 vmem privs

The results from the Virtual Memory (vmem) privileges module are located in Appendix B due to the numerous entries.

6.8 vadtree

For both processes, the Virtual Address Descriptor (VAD) ranges can be seen in Figure 22:
6.9 vadinfo

The vadinfo module produced information, specifically of interest is the starting and ending memory address range, of each VAD entry in the vadtree but due to the numerous entries, they are located in Appendix C.

6.10 Failed modules

Unfortunately, several volatility modules failed to parse the memory image and resulted in errors which returned with no useful output. The modules which failed were: psxview, shellbags, sockets, services, userassist, and devicetree. The apihooks module did not fail but instead returned no data, so the suspicion is that the module failed as well.

6 Anti-Forensics

Mike Peery outlines several design requirements for the Tor Browser which are “that [it] defends against both network and local forensic adversaries” (Perry, 2015, para. 8). For instance, “the browser MUST NOT write any information that is derived from or that reveals browsing activity to the disk” (Peery, 2015, para. 15). This statement was confirmed by the fact that while carving and performing cookie, history, and cache recover against Firefox’s artifacts, nothing was found written to disk. An attempt using tools such as MZHistoryView, MZCacheView and MZCookieView from Nirsoft did not provide any results as the author could not get the tools to read the proper DLLs.

The only on-disk artifact found was when a VM snapshot was taken during a connection to Tor with the user surfing to google.com and bookmarking the site. The places.sqlite on-disk did contain the google.com location visited, as shown in Figure 23:

```
root# echo '.dump' | sqlite3 places.sqlite > places.txt
root# more places.txt
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
```
CREATE TABLE moz_places (   id INTEGER PRIMARY KEY, url LONGVARCHAR, title LONGVARCHAR, rev_host LONGVARCHAR, visit_count INTEGER DEFAULT 0, hidden INTEGER DEFAULT 0 NOT NULL, typed INTEGER DEFAULT 0 NOT NULL, favicon_id INTEGER, frecency INTEGER DEFAULT -1 NOT NULL, last_visit_date INTEGER, guid TEXT, foreign_count INTEGER DEFAULT 0 NOT NULL);  
INSERT INTO "moz_places" VALUES(1,'https://www.torproject.org/','null','gro.tcejorprot.www.',0,0,0,1,140 ,null,'sKr_fL_2R5gK',1);  
INSERT INTO "moz_places" VALUES(2,'https://blog.torproject.org/','null','gro.tcejorprot.golb.',0,0,0,2, 140,null,'PChOdf-VbdEN',1);  
INSERT INTO "moz_places" VALUES(3,'place:sort=8&maxResults=10','null',null,0,1,0,null,0,null,'yrrZEMbCO me',1);  
INSERT INTO "moz_places" VALUES(4,'place:folder=BOOKMARKS_MENU&folder=UNFILED_BOOKMARKS&folder=TOOLBAR &queryType=1&sort=12&maxResults=10&excludeQueries=1','null',null,0,1,0,null,0,null,'Gx39ziCQ-gg',1);  
INSERT INTO "moz_places" VALUES(5,'place:type=6&sort=14&maxResults=10','null',null,0,1,0,null,0,null,'Bo bBb40Vn6ot',1);  
INSERT INTO "moz_places" VALUES(6,'https://www.google.com/','null','moc.elgoog.www.',0,1,0,null,- 1,null,'ATecMqhOR884',1);  
CREATE TABLE moz_historyvisits (   id INTEGER PRIMARY KEY, from_visit INTEGER, place_id INTEGER, visit_date INTEGER, visit_type INTEGER, session INTEGER);  
CREATE TABLE moz_inputhistory (   place_id INTEGER NOT NULL, input LONGVARCHAR NOT NULL, use_count INTEGER, PRIMARY KEY (place_id, input));  
CREATE TABLE moz_hosts (  id INTEGER PRIMARY KEY, host TEXT NOT NULL UNIQUE, frecency INTEGER, typed INTEGER NOT NULL DEFAULT 0, prefix TEXT);  
INSERT INTO "moz_hosts" VALUES(1,'torproject.org',140,0,null);  
INSERT INTO "moz_hosts" VALUES(2,'blog.torproject.org',140,0,Null);  
INSERT INTO "moz_hosts" VALUES(3,'google.com', -1,0,null);  
CREATE TABLE moz_bookmarks (   id INTEGER PRIMARY KEY, type INTEGER, fk INTEGER DEFAULT NULL, parent INTEGER, position INTEGER, title LONGVARCHAR, keyword_id INTEGER, folder_type TEXT, dateAdded INTEGER, lastModified INTEGER, guid TEXT);  
INSERT INTO "moz_bookmarks" VALUES(1,2,NULL,0,0,'',null,Null,1476026354653000,1476026354653000,'root______');  
INSERT INTO "moz_bookmarks" VALUES(2,2,NULL,1,0,'Bookmarks Menu',null,null,1476026354653000,1483030789911000,'menu________');  
INSERT INTO "moz_bookmarks" VALUES(3,2,NULL,1,1,'Bookmarks Toolbar',null,null,1476026354653000,1476026355340000,'toolbar_____');  
INSERT INTO "moz_bookmarks" VALUES(4,2,NULL,1,2,'Tags',null,null,1476026354653000,1476026354653000,'tags_______');  
INSERT INTO "moz_bookmarks" VALUES(5,2,NULL,1,3,'Unsorted Bookmarks',null,null,1476026354653000,1476026355325000,'unfiled______');  
INSERT INTO "moz_bookmarks" VALUES(6,1,1,3,'Learn more about Tor',null,null,1476026355340000,1476026355340000,'Naid2pnu7289');  
INSERT INTO "moz_bookmarks" VALUES(7,1,2,3,2,'The Tor Blog',null,null,1476026355340000,1476026355340000,'IxpDbp40F0dD');  
INSERT INTO "moz_bookmarks" VALUES(8,3,NULL,2,2,'null',null,null,1476026355340000,1476026355340000,'9ZBE3mo BwrcJ');
Figures 23. The contents of places.sqlite showing locations browsed and bookmarks set.

Having the browsing history stored on-disk, albeit located solely inside the Tor installation directory, in and of itself is a trade-off between security and functionality. To mitigate this concern and others, tor_opsec (n.d.) gives a full list of steps for proper Operational Security (OPSEC) when using Tor too detailed and numerous to outline here. They are necessary if the end-user is going to maintain maximum anonymity while using Tor.

## 7 Conclusion

This paper began with an overview of The Onion Router (Tor) project and described the subsequent creation of the Tor Browser. A detailed overview of a Tor Browser installation and forensic methodology was provided so that the reader could recreate this analysis. After carving a prefetch file, system and user hives, as well as Mozilla on-disk files, the Tor project’s goal of leaving a minimal footprint on-disk is confirmed by the above filesystem analysis. Memory analysis used provided various artifacts pointing to the installation location of the Tor Browser in addition to Internet locations the browser was connected to. In the end, using the above analysis, dozens of pointers to artifacts is provided to assist other investigators in identifying the location and use of the Tor Browser. Future research using reverse engineering techniques to dump memory locations may prove useful in identifying activity occurring when the third VM snapshot was taken.
References


Tor FAQ. (n.d.). Retrieved from https://www.torproject.org/docs/faq

Tor_opsec. (n.d.) Here is a quick guide to using Tor + OPSEC. Retrieved from https://www.reddit.com/r/TOR/comments/3dq1pg/here_is_a_quick_guide_to_using_tor_opsec/


Aron Warren, aronwarren@gmail.com
Appendix A:

The dumpfiles from subsection 6.6 are in Figure 24:

| ImageSectionObject 0xfffffe0001898d850 4684 \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\firefox.exe |
| DataSectionObject 0xfffffe0001898d850 4684 \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\firefox.exe |
| DataSectionObject 0xfffffe00019b68720 4684 \Device\HarddiskVolume2\Windows\SysWOW64\en-US\UIAutomationCore.dll.mui |
| DataSectionObject 0xfffffe000195a6ba0 4684 \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\profile.default\webappsstore.sqlite-shm |
| DataSectionObject 0xfffffe00019001b90 4684 None |
| DataSectionObject 0xfffffe000195a6ba0 4684 \Device\HarddiskVolume2\Windows\SysWOW64\en-US\rasdlg.dll.mui |
| DataSectionObject 0xfffffe0001904d270 4684 None |
| DataSectionObject 0xfffffe000191e3240 4684 None |
| DataSectionObject 0xfffffe00019ecff20 4684 \Device\HarddiskVolume2\Windows\SysWOW64\d2d1.dll.mui |
| ImageSectionObject 0xfffffe000191db520 4684 None |
| ImageSectionObject 0xfffffe000191db520 4684 None |
| DataSectionObject 0xfffffe000191e3240 4684 None |

Aron Warren, aronwarren@gmail.com
Aron Warren, aronwarren@gmail.com
Tor Browser Artifacts in Windows 10

Aron Warren, aronwarren@gmail.com
Figure 24. The dump file locations are the locations of where the files in memory are located.
Appendix B:

Artifacts found from the vmem privs module from Volatility are shown in Figure 25.

<table>
<thead>
<tr>
<th>Process</th>
<th>Privilege Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>firefox.exe</td>
<td>SeCreateTokenPrivilege</td>
<td>Create a token object</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeAssignPrimaryTokenPrivilege</td>
<td>Replace a process-level token</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeLockMemoryPrivilege</td>
<td>Lock pages in memory</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeIncreaseQuotaPrivilege</td>
<td>Increase quotas</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeMachineAccountPrivilege</td>
<td>Add workstations to the domain</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeTcbPrivilege</td>
<td>Act as part of the operating system</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeSecurityPrivilege</td>
<td>Manage auditing and security log</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeTakeOwnershipPrivilege</td>
<td>Take ownership of files/objects</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeLoadDriverPrivilege</td>
<td>Load and unload device drivers</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeSystemProfilePrivilege</td>
<td>Profile system performance</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeSystemTimePrivilege</td>
<td>Change the system time</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeProfileSingleProcessPrivilege</td>
<td>Profile a single process</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeIncreaseBasePriorityPrivilege</td>
<td>Increase scheduling priority</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeCreatePagefilePrivilege</td>
<td>Create a pagefile</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeCreatePermanentPrivilege</td>
<td>Create permanent shared objects</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeBackupPrivilege</td>
<td>Backup files and directories</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeRestorePrivilege</td>
<td>Restore files and directories</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeShutdownPrivilege</td>
<td>Shut down the system</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeDebugPrivilege</td>
<td>Debug programs</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeAuditPrivilege</td>
<td>Generate security audits</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeTrustedCredManAccessPrivilege</td>
<td>Access Credential Manager as a trusted caller</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeRelabelPrivilege</td>
<td>Modify the mandatory integrity level of an object</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeIncreaseWorkingSetPrivilege</td>
<td>Present Allocate more memory for user applications</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeTimeZone</td>
<td>Adjust the time zone of the computer's internal clock</td>
</tr>
<tr>
<td>firefox.exe</td>
<td>SeCreateSymbolicLinkPrivilege</td>
<td>Required to create a symbolic link</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeCreateTokenPrivilege</td>
<td>Create a token object</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeAssignPrimaryTokenPrivilege</td>
<td>Replace a process-level token</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeLockMemoryPrivilege</td>
<td>Lock pages in memory</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeIncreaseQuotaPrivilege</td>
<td>Increase quotas</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeMachineAccountPrivilege</td>
<td>Add workstations to the domain</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeTcbPrivilege</td>
<td>Act as part of the operating system</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeSecurityPrivilege</td>
<td>Manage auditing and security log</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeTakeOwnershipPrivilege</td>
<td>Take ownership of files/objects</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeLoadDriverPrivilege</td>
<td>Load and unload device drivers</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeSystemProfilePrivilege</td>
<td>Profile system performance</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeSystemTimePrivilege</td>
<td>Change the system time</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeProfileSingleProcessPrivilege</td>
<td>Profile a single process</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeIncreaseBasePriorityPrivilege</td>
<td>Increase scheduling priority</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeCreatePagefilePrivilege</td>
<td>Create a pagefile</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeCreatePermanentPrivilege</td>
<td>Create permanent shared objects</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeBackupPrivilege</td>
<td>Backup files and directories</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeRestorePrivilege</td>
<td>Restore files and directories</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeShutdownPrivilege</td>
<td>Shut down the system</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeDebugPrivilege</td>
<td>Debug programs</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeAuditPrivilege</td>
<td>Generate security audits</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeSystemEnvironmentPrivilege</td>
<td>Edit firmware environment values</td>
</tr>
<tr>
<td>tor.exe</td>
<td>SeChangeNotifyPrivilege</td>
<td>Present, Enabled, Default Receive notifications of changes to files or directories</td>
</tr>
<tr>
<td>Process</td>
<td>Privilege Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>24 SeRemoteShutdownPrivilege</td>
<td>Force shutdown from a remote system</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>25 SeUndockPrivilege</td>
<td>Present computer from docking station</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>26 SeSyncAgentPrivilege</td>
<td>Synch directory service data</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>27 SeEnableDelegationPrivilege</td>
<td>Enable user accounts to be trusted for delegation</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>28 SeManageVolumePrivilege</td>
<td>Manage the files on a volume</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>29 SeImpersonatePrivilege</td>
<td>Impersonate a client after authentication</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>30 SeCreateGlobalPrivilege</td>
<td>Create global objects</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>31 SeTrustedCredManAccessPrivilege</td>
<td>Access Credential Manager as a trusted caller</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>32 SeRelabelPrivilege</td>
<td>Modify the mandatory integrity level of an object</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>33 SeIncreaseWorkingSetPrivilege</td>
<td>Present Allocate more memory for user applications</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>34 SeTimeZonePrivilege</td>
<td>Present Adjust the time zone of the computer's internal clock</td>
</tr>
<tr>
<td>4476 tor.exe</td>
<td>35 SeCreateSymbolicLinkPrivilege</td>
<td>Required to create a symbolic link</td>
</tr>
</tbody>
</table>

*Figure 25. The privileges granted to each process.*
Appendix C:

This section contains the relevant artifacts found from the vadinfo module of Volatility:

```
VAD node @ 0xffffe000189a78b0 Start 0x0000000000060000 End 0x00000000000b4fff Tag Vad Flags: Protection: 7, VadType: 2 Protection: PAGE_EXECUTE_WRITECOPY Vad Type: VadImageMap Control Area @ffffff000189f830 Segment ffffc000f9636b90 NumberOfSectionReferences: 1 NumberOfPfnReferences: 39 NumberOfMappedViews: 1 NumberOfUserReferences: 2 Control Flags: File: 1, Image: 1 FileObject @ffffff000189f850, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\firefox.exe First prototype PTE: ffffc000f3f0c910 Last contiguous PTE: ffffc000f3f0cbb0 Flags2: Inherit: 1, NoValidationNeeded: 1

VAD node @ 0xffffe000175ba930 Start 0x0000000000b80000 End 0x0000000000b87fff Tag Vad Flags: Protection: 4 Protection: PAGE_READWRITE Vad Type: VadNone Control Area @ffffff00018884490 Segment ffffc000fa1efc00 NumberOfSectionReferences: 1 NumberOfPfnReferences: 8 NumberOfMappedViews: 1 NumberOfUserReferences: 2 Control Flags: File: 1 FileObject @ffffff00017dd4090, Name: \Device\HarddiskVolume2\Users\warren\Browser\profile.default\places.sqlite-shm First prototype PTE: ffffc000f0eff8d0 Last contiguous PTE: ffffc000f0eff908 Flags2: Inherit: 1, TrimBehind: 1

VAD node @ 0xffffe00019fccc750 Start 0x0000000007810000 End 0x0000000007817fff Tag Vad Flags: Protection: 4 Protection: PAGE_READWRITE Vad Type: VadNone Control Area @ffffff0001818e00 Segment ffffc000fb6d9b00 NumberOfSectionReferences: 1 NumberOfPfnReferences: 8 NumberOfMappedViews: 1 NumberOfUserReferences: 2 Control Flags: File: 1 FileObject @ffffff00019566a0, Name: \Device\HarddiskVolume2\Users\warren\Browser\profile.default\webappsstore.sqlite-shm First prototype PTE: ffffc000f4d3e820 Last contiguous PTE: ffffc000f4d3e858 Flags2: Inherit: 1, TrimBehind: 1

(Note: X-Ways was not able to find this file in the disk image.)

VAD node @ 0xffffe00017d13930 Start 0x00000000069ad000 End 0x00000000069b48ff Tag Vad Flags: Protection: 7, VadType: 2 Protection: PAGE_EXECUTE_WRITECOPY Vad Type: VadImageMap Control Area @ffffff0001a8790 Segment ffffc000f306c740 NumberOfSectionReferences: 0 NumberOfPfnReferences: 114 NumberOfMappedViews: 1 NumberOfUserReferences: 1 Control Flags: File: 1, Image: 1 FileObject @ffffff00017d949d0, Name: \Device\HarddiskVolume2\Users\warren\Browser\profile.default\freebl3.dll First prototype PTE: ffffc000f4fde9ad0 Last contiguous PTE: ffffc000f4fde9e90 Flags2: Inherit: 1, NoValidationNeeded: 1

VAD node @ 0xffffe00019b30ab0 Start 0x00000000069b50000 End 0x00000000069bc0fff Tag Vad Flags: Protection: 7, VadType: 2 Protection: PAGE_EXECUTE_WRITECOPY Vad Type: VadImageMap Control Area @ffffff0001248a0 Segment ffffc000f4d07760 NumberOfSectionReferences: 0 NumberOfPfnReferences: 82 NumberOfMappedViews: 1 NumberOfUserReferences: 1 Control Flags: File: 1, Image: 1 FileObject @ffffff00018919360, Name: \Device\HarddiskVolume2\Users\warren\Browser\freebl3.dll

Aron Warren, aronwarren@gmail.com
```
First prototype PTE: ffffc000f50b6c70 Last contiguous PTE: ffffc000f50b6ff0
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe00019409170 Start 0x0000000069e90000 End 0x000000006cfdffff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea: ffffc00017de06e0 Segment ffffc000f9668290
NumberOfSectionReferences: 0 NumberOfPfnReferences: 6355
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject: ffffc00019bfe710, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\browser\xul.dll
First prototype PTE: ffffc0000fc76000 Last contiguous PTE: ffffc000fbc8ea78
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a0e8e60 Start 0x000000006e8a0000 End 0x000000006e8dafff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea: ffffc00018902d10 Segment ffffc000f4a5fb90
NumberOfSectionReferences: 0 NumberOfPfnReferences: 33
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject: ffffc0001892ed00, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\browser\components\browsercomps.dll
First prototype PTE: ffffc000f4e3d920 Last contiguous PTE: ffffc000f4e3daf0
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a46d280 Start 0x000000006fb30000 End 0x000000006fb3bffff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea: ffffc00019ef8ae0 Segment ffffc000f54135f0
NumberOfSectionReferences: 0 NumberOfPfnReferences: 10
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject: ffffc0001998c980, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\mozglue.dll
First prototype PTE: ffffc000f4e6a880 Last contiguous PTE: ffffc000f4e6a8d8
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a0177b20 Start 0x000000006ee00000 End 0x000000006edcffff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea: ffffc000194e2870 Segment ffffc000f5928800
NumberOfSectionReferences: 0 NumberOfPfnReferences: 27
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject: ffffc000194ccf20, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\nss3.dll
First prototype PTE: ffffc000f5020910 Last contiguous PTE: ffffc000f5020ff0
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a49c910 Start 0x000000006eb40000 End 0x000000006ec2dfff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea: ffffc000177b3010 Segment ffffc000f5928800
NumberOfSectionReferences: 0 NumberOfPfnReferences: 164
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject: ffffc000195f54820, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\nss3.dll
First prototype PTE: ffffc000f4feca90 Last contiguous PTE: ffffc000f4fe0ff8
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a097100 Start 0x000000006ed20000 End 0x000000006ed54fff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap

Aron Warren, aronwarren@gmail.com
ControlArea @ffffe00017d2dbb0 Segment ffffc000f55311c0
NumberOfSectionReferences: 0 NumberOfPfnReferences: 46
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject @ffffe00013e0350, Name: \Device\Hاردیسکدیوامل2\Users\warren\Desktop\Tor Browser\Browser\ssl3.dll
First prototype PTE: ffffc000fbaa0e50 Last contiguous PTE: ffffc000fbaa0ff0
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a3e0350 Start 0x000000006edd6000 End 0x000000006ed86fff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea @ffffe00019be750 Segment ffffc000f94322a0
NumberOfSectionReferences: 0 NumberOfPfnReferences: 1
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject @ffffe000194abe80, Name: \Device\Hاردیسکدیوامل2\Users\warren\Desktop\Tor Browser\Browser\smime3.dll
First prototype PTE: ffffc000f4fbae960 Last contiguous PTE: ffffc000f4fbae90
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a3e0350 Start 0x000000006edd6000 End 0x000000006ed5fff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea @ffffe00019b82a00 Segment ffffc000f59f4a60
NumberOfSectionReferences: 0 NumberOfPfnReferences: 77
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject @ffffe00019bee090, Name: \Device\Hاردیسکدیوامل2\Users\warren\Desktop\Tor Browser\Browser\msvcr100.dll
First prototype PTE: ffffc000f4f4f5a10 Last contiguous PTE: ffffc000f4f4f5f8
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a3e0350 Start 0x000000006f460000 End 0x000000006f46bfff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea @ffffe00017d8440 Segment ffffc000f4f61ae0
NumberOfSectionReferences: 0 NumberOfPfnReferences: 10
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject @ffffe000191ee4b0, Name: \Device\Hاردیسکدیوامل2\Users\warren\Desktop\Tor Browser\Browser\plds4.dll
First prototype PTE: ffffc000f4f4f5a10 Last contiguous PTE: ffffc000f4f4f5f8
Flags2: Inherit: 1, NoValidationNeeded: 1
VAD node @ 0xffffe0001a3e0350 Start 0x000000006f460000 End 0x000000006f46bfff Tag Vad
Flags: Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea @ffffe000191ee4b0 Segment ffffc000f307e10
NumberOfSectionReferences: 0 NumberOfPfnReferences: 11
NumberOfMappedViews: 1 NumberOfUserReferences: 1
Control Flags: File: 1, Image: 1
FileObject @ffffe000138a090, Name: \Device\Hاردیسکدیوامل2\Users\warren\Desktop\Tor Browser\Browser\plc4.dll
First prototype PTE: ffffc000f4e920 Last contiguous PTE: ffffc000f4e980

Aron Warren, aronwarren@gmail.com
<table>
<thead>
<tr>
<th>Flags2: Inherit: 1, NoValidationNeeded: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD node @ 0xffffe00017570450 Start 0x000000006f470000 End 0x000000006f505fff Tag Vad</td>
</tr>
<tr>
<td>Protection: 7, VadType: 2</td>
</tr>
<tr>
<td>Protection: PAGE_EXECUTE_WRITECOPY</td>
</tr>
<tr>
<td>Vad Type: VadImageMap</td>
</tr>
<tr>
<td>ControlArea @ 0xffffe00019bea450 Segment 0xffffe00019de6a60</td>
</tr>
<tr>
<td>NumberOfSectionReferences: 0 NumberOfPfnReferences: 129</td>
</tr>
<tr>
<td>NumberOfMappedViews: 1 NumberOfUserReferences: 1</td>
</tr>
<tr>
<td>Control Flags: File: 1, Image: 1</td>
</tr>
<tr>
<td>FileObject @ 0xffffe00019be86a0, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\mozgiite3.dll</td>
</tr>
<tr>
<td>First prototype PTE: 0xffffe00019bea450 Last contiguous PTE: 0xffffe00019de6a60</td>
</tr>
<tr>
<td>Flags2: Inherit: 1, NoValidationNeeded: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flags2: Inherit: 1, NoValidationNeeded: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD node @ 0xffffe0001a3b5190 Start 0x000000006f610000 End 0x000000006f62bfff Tag Vad</td>
</tr>
<tr>
<td>Protection: 7, VadType: 2</td>
</tr>
<tr>
<td>Protection: PAGE_EXECUTE_WRITECOPY</td>
</tr>
<tr>
<td>Vad Type: VadImageMap</td>
</tr>
<tr>
<td>ControlArea @ 0xffffe000187350 Segment 0xffffe0000faef2c0</td>
</tr>
<tr>
<td>NumberOfSectionReferences: 0 NumberOfPfnReferences: 19</td>
</tr>
<tr>
<td>NumberOfMappedViews: 1 NumberOfUserReferences: 1</td>
</tr>
<tr>
<td>Control Flags: File: 1, Image: 1</td>
</tr>
<tr>
<td>FileObject @ 0xffffe0001947af20, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\libssp-0.dll</td>
</tr>
<tr>
<td>First prototype PTE: 0xffffe000187350 Last contiguous PTE: 0xffffe0000faef2c0</td>
</tr>
<tr>
<td>Flags2: Inherit: 1, NoValidationNeeded: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flags2: Inherit: 1, NoValidationNeeded: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD node @ 0xffffe00017cb71f0 Start 0x000000008800000 End 0x000000008800000a7bfff Tag Vad</td>
</tr>
<tr>
<td>Protection: 7, VadType: 2</td>
</tr>
<tr>
<td>Protection: PAGE_EXECUTE_WRITECOPY</td>
</tr>
<tr>
<td>Vad Type: VadImageMap</td>
</tr>
<tr>
<td>ControlArea @ 0xffffe00017c73010 Segment 0xffffe000f31ba0f0</td>
</tr>
<tr>
<td>NumberOfSectionReferences: 1 NumberOfPfnReferences: 458</td>
</tr>
<tr>
<td>NumberOfMappedViews: 1 NumberOfUserReferences: 2</td>
</tr>
<tr>
<td>Control Flags: File: 1, Image: 1</td>
</tr>
<tr>
<td>FileObject @ 0xffffe000192fc460, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\TorBrowser\tor.exe</td>
</tr>
<tr>
<td>First prototype PTE: 0xffffe000f31ba0f0 Last contiguous PTE: 0xffffe000f31ba0f0</td>
</tr>
<tr>
<td>Flags2: Inherit: 1, NoValidationNeeded: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flags2: Inherit: 1, NoValidationNeeded: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD node @ 0xffffe0001a583b80 Start 0x0000000069650000 End 0x00000000696500006a7bfff Tag Vad</td>
</tr>
<tr>
<td>Protection: 7, VadType: 2</td>
</tr>
<tr>
<td>Protection: PAGE_EXECUTE_WRITECOPY</td>
</tr>
<tr>
<td>Vad Type: VadImageMap</td>
</tr>
<tr>
<td>ControlArea @ 0xffffe00019c0a430 Segment 0xffffe000f3221c20</td>
</tr>
<tr>
<td>NumberOfSectionReferences: 0 NumberOfPfnReferences: 9</td>
</tr>
<tr>
<td>NumberOfMappedViews: 1 NumberOfUserReferences: 1</td>
</tr>
<tr>
<td>Control Flags: File: 1, Image: 1</td>
</tr>
<tr>
<td>FileObject @ 0xffffe000199372b0, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\TorBrowser\TorBrowser\tor\ssleay32.dll</td>
</tr>
<tr>
<td>First prototype PTE: 0xffffe000f3221c20 Last contiguous PTE: 0xffffe000f3221c20</td>
</tr>
<tr>
<td>Flags2: Inherit: 1, NoValidationNeeded: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flags2: Inherit: 1, NoValidationNeeded: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD node @ 0xffffe0001a59f720 Start 0x0000000069f80000 End 0x0000000069f8000069f8bfff Tag Vad</td>
</tr>
<tr>
<td>Protection: 7, VadType: 2</td>
</tr>
<tr>
<td>Protection: PAGE_EXECUTE_WRITECOPY</td>
</tr>
<tr>
<td>Vad Type: VadImageMap</td>
</tr>
<tr>
<td>ControlArea @ 0xffffe0001898ad10 Segment 0xffffe000f3221a00</td>
</tr>
<tr>
<td>NumberOfSectionReferences: 0 NumberOfPfnReferences: 103</td>
</tr>
<tr>
<td>NumberOfMappedViews: 1 NumberOfUserReferences: 1</td>
</tr>
<tr>
<td>Control Flags: File: 1, Image: 1</td>
</tr>
<tr>
<td>FileObject @ 0xffffe000190be290, Name: \Device\HarddiskVolume2\Users\warren\Desktop\Tor Browser\Browser\TorBrowser\Tor\ssleay32.dll</td>
</tr>
<tr>
<td>First prototype PTE: 0xffffe000f3221a00 Last contiguous PTE: 0xffffe000f3221a00</td>
</tr>
<tr>
<td>Flags2: Inherit: 1, NoValidationNeeded: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flags2: Inherit: 1, NoValidationNeeded: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAD node @ 0xffffe000196084c0 Start 0x0000000069340000 End 0x00000000693400006934bfff Tag Vad</td>
</tr>
<tr>
<td>Protection: 7, VadType: 2</td>
</tr>
<tr>
<td>Protection: PAGE_EXECUTE_WRITECOPY</td>
</tr>
<tr>
<td>Vad Type: VadImageMap</td>
</tr>
<tr>
<td>ControlArea @ 0xffffe00017c9010 Segment 0xffffe000f321d880</td>
</tr>
</tbody>
</table>

Aron Warren, aronwarren@gmail.com
Figure 26. The VAD node’s description.
## Upcoming SANS Forensics Training

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS San Diego 2020</td>
<td>San Diego, CA</td>
<td>Feb 17, 2020 - Feb 22, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Scottsdale 2020</td>
<td>Scottsdale, AZ</td>
<td>Feb 17, 2020 - Feb 22, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Brussels February 2020</td>
<td>Brussels, Belgium</td>
<td>Feb 17, 2020 - Feb 22, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>Open-Source Intelligence Summit &amp; Training 2020</td>
<td>Alexandria, VA</td>
<td>Feb 18, 2020 - Feb 24, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Zurich February 2020</td>
<td>Zurich, Switzerland</td>
<td>Feb 24, 2020 - Feb 29, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Secure India 2020</td>
<td>Bangalore, India</td>
<td>Feb 24, 2020 - Feb 29, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS vLive - FOR578: Cyber Threat Intelligence</td>
<td>FOR578 - 202002,</td>
<td>Feb 25, 2020 - Apr 02, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>Northern VA - Reston Spring 2020 - FOR585: Smartphone</td>
<td>Reston, VA</td>
<td>Mar 02, 2020 - Mar 07, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>Forensic Analysis In-Depth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SANS Northern VA - Reston Spring 2020</td>
<td>Reston, VA</td>
<td>Mar 02, 2020 - Mar 07, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Secure Japan 2020</td>
<td>Tokyo, Japan</td>
<td>Mar 02, 2020 - Mar 14, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Munich March 2020</td>
<td>Munich, Germany</td>
<td>Mar 02, 2020 - Mar 07, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS St. Louis 2020</td>
<td>St. Louis, MO</td>
<td>Mar 08, 2020 - Mar 13, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>Dallas 2020 - FOR500: Windows Forensic Analysis</td>
<td>Dallas, TX</td>
<td>Mar 09, 2020 - Mar 14, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>SANS Dallas 2020</td>
<td>Dallas, TX</td>
<td>Mar 09, 2020 - Mar 14, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Norfolk 2020</td>
<td>Norfolk, VA</td>
<td>Mar 16, 2020 - Mar 21, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS San Francisco Spring 2020</td>
<td>San Francisco, CA</td>
<td>Mar 16, 2020 - Mar 27, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>Mentor Session - FOR508</td>
<td>Sao Paulo, Brazil</td>
<td>Mar 25, 2020 - Mar 28, 2020</td>
<td>Mentor</td>
</tr>
<tr>
<td>SANS Abu Dhabi March 2020</td>
<td>Abu Dhabi, United Arab Emirates</td>
<td>Mar 28, 2020 - Apr 02, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS FOR585 Rome March 2020 (In Italian)</td>
<td>Rome, Italy</td>
<td>Mar 30, 2020 - Apr 04, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Frankfurt March 2020</td>
<td>Frankfurt, Germany</td>
<td>Mar 30, 2020 - Apr 04, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS vLive - FOR508: Advanced Incident Response, Threat Hunting, and Digital Forensics</td>
<td>FOR508 - 202003,</td>
<td>Mar 31, 2020 - May 07, 2020</td>
<td>vLive</td>
</tr>
<tr>
<td>SANS 2020</td>
<td>Orlando, FL</td>
<td>Apr 03, 2020 - Apr 10, 2020</td>
<td>Live Event</td>
</tr>
</tbody>
</table>