

I2P Forensic Analysis



Sneha Soney, C. Balan, Priya P. Sajan, Elizabeth Rose Lalson

Abstract: I2P is an anonymous P2P distributed communication layer used to send messages to each other anonymously and safely. It is built on top of the internet and can be considered as an internet within the internet. Even though I2P is developed with an intention to create censorship resistant environment for the free flow of information, it is misused for illegal activities now a days. The possible misuses are less known among law enforcement agencies and existing industry approved software programs have no detection functionality for I2P. Because of the increased use of I2P in criminal purposes, there is a need for methods and tools to acquire and analyze digital evidence related to I2P. We conducted a detailed live memory dump analysis in order to find out the I2P related artifacts from a host machine. Furthermore, we propose a tool that will analyze the memory dump and system local files to find out the I2P related artifacts and provide a detailed report to the investigator.

Keywords: I2P, P2P, Artifacts, Memory Analysis

I. INTRODUCTION

Darkweb is turning out to be a familiar term nowadays. It invokes some common impressions like the place of shady dealings, anonymous communications, illicit materials etc. The term “darknet” was coined within the seventies to designate networks isolated from ARPANET (the government based military/academical network that evolved in to internet), for security functions. Darknet could receive data from ARPANET but did not appear in the network lists and would not answer pings or other inquiries. I2P is an anonymous network, demonstrating a basic layer that users can use to ship messages to each other anonymously and securely. It is intended to be used with nodes that have I2P system installed. I2P can relay traffic via multiple nodes using tunnels and encapsulated data messages which are routed until the destination is reached. It is a message based and fully distributed system that does not rely on centralized directory on servers to keep track of participating nodes and network

performance. I2P uses garlic routing for anonymous communication, which is a variant of Tor’s onion routing.

The success of the recent enforcement operations that have cracked down on criminal services operative on the Tor network is currently inflicting a shift within the criminal landscape. Recently, much of the online criminal activities happens through I2P network. The Libertas Market, a Tor-based portal for selling illegal products, permanently abandoned the Tor network for I2P. Cryptowall 3.0 ransomware and Dyre banking Trojan used I2P for their communication. The malware named i2Ninja, was noticed available at an underground Russian cyber-crime forum. For all major browsers (Internet Explorer, Firefox and Chrome), i2Ninja is capable of HTML injection and form grabbing. The malware toolkit uses I2P networking layer to mask communications between infected machines and the botnet’s command-and-control server. Because of increase in cyber-crimes using I2P there is a need for methods and tools to acquire and analyse digital evidence related to the same. Anonymous communications is a challenge to forensic investigators because the evidence collection is difficult in such cases. Hence there raised a need to combine the analysis of host, memory and network to find artifacts of I2P from a computer system.

II. INVISIBLE INTERNET PROJECT

Invisible Internet Project (I2P) is an anonymous peer to peer network layer upon which any number of anonymous applications can work. The applications in I2P provide internet activities like anonymous web browsing, chat, file sharing, email, blogs and many more. The network has been in active development since 2003. All the work done on I2P is open source and freely available on website. The software participating in the I2P network is called a router. This software is separated from anonymous endpoints or destinations associated with different applications. So an end user will have several local destinations on their router. A tunnel created through an explicitly selected list of routers are used for sending messages. The message can only be sent in one direction. Another tunnel is required to send messages back. Inbound and outbound tunnels are used in I2P communication. Layered encryption is used to ensure the anonymity of the communications. Garlic routing, a variant of onion routing is used in I2P. It will encrypt multiple messages together. As a result traffic analysis become difficult and speed of data transfer also increases.

III. RELATED WORKS

Maxim and Behnam [1] investigate the characteristics of I2P network in order to outline the problems and methods in detection of I2P artifacts.

Manuscript received on February 10, 2020.

Revised Manuscript received on February 20, 2020.

Manuscript published on March 30, 2020.

* Correspondence Author

Sneha Soney*, Department of Computer Science, ER & DCI Institute of Technology, Thiruvananthapuram, India.

Email: snehasoney428@gmail.com.

C. Balan, Knowledge Resource Centre, CDAC, Thiruvananthapuram, India. Email: cbalan@cdac.in.

Priya P. Sajan, Knowledge Resource Centre, CDAC, Thiruvananthapuram, India. Email: priyasajan@cdac.in.

Elizabeth Rose Lalson, Department of Computer Science, ER & DCI Institute of Technology, Thiruvananthapuram, India.

Email: elizabeth@cdac.in.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

They put forward the methods like investigation of I2P installers, detection via known hash set library, comparison of address books, takeover of existing registrars and mirroring of the eepsites. Juan et al. [2] take the first step towards the linkability of users and applications in the I2P network.

They conducted a group-based characterization, where they determine the extent to which a group of users is responsible for the file-sharing activity of the overall I2P. Juan et al. [3] have also designed and successfully implemented and deployed the first large scale monitoring architecture for the I2P network, mainly focused on anonymous file-sharing. They arrive at the conclusion that group based identification is possible, even if single user identification is not.

Based on the analysis on the I2P data, Khalid [4] concluded that the resource sharing (bandwidth participation) of the users on the I2P network improves the anonymity level of the users. On the other hand, using the default setting for not using the shared client tunnels for all applications seems to reduce the anonymity level and enables the application profiling ability of a potential attacker. Yue Gao et al. [5] proposed three approaches to discover eepSites: (1) running floodfill routers, (2) gathering hosts.txt files actively and (3) crawling popular portal eepSites. From the realworld experiments, the combination of the three methods in total discovered 1861 online eepSites covering over 80% of all eepsites in I2P network. Mathias Ehlert [6] conducted tests to analyze latency and bandwidth drawbacks while using I2P to surf the internet anonymously.

Preethi and Dinesha [7] conducted a comparative study of different packet sniffers, protocol analyzers and intrusion detection system that can be used for forensic analysis of darknet. They compared the packet sniffers and protocol analyzers like Tcpdump, Windump, Fiddler, Network Miner, Wireshark/tshark, Capsa, Netsniff-ng and intrusion detection systems like Snort, Suricata, Bro IDS, Security Onion, OpenWIPS-NG, Kismet and NetDetector. Dr. Digvijaysinh [8] proposed techniques for Tor browser and Bitcoin transaction forensics. According to him, the evidences related to Tor browser can be extracted from RAM forensics, registry changes, network forensics and database changes. Bitcoin transaction forensics can be carried out by extracting forensic artifacts from installed Bitcoin wallet application on user's system. Afzaal et. al. [9] in his work conducted an in depth comparative study on the two popular anonymous systems I2P and Tor.

IV. PROBLEM STATEMENT

Computer forensics is a branch of forensic science that deals with the evidence collection from computers or digital storage mediums. The main objective of computer forensics is to identify, collect, preserve and analyse data in a way that preserves the integrity of the evidence gathered so it may be used effectively in a legal case. Many forensic tools are currently available which facilitates the digital investigations.

The investigations related to anonymous networks like I2P is always a headache to the law enforcement agencies. The use of I2P networks in criminal activities are increasing day by day. Current widely accepted forensic tools like Encase or

FTK will not address the detection of I2P by default and the possible use of I2P in cybercrimes are less known among investigators. So there raised a need for an effective analysis tool that detect and analyze I2P related artifacts from a computer system.

The proposed system took the physical memory dump and system local files as input and generate a report containing I2P related artifacts as output. The input files are hashed in order to ensure integrity. The artifacts were collected from the input files and analyzed for the I2P related evidences. A report containing I2P artifacts were generated for the user. Fig. 1. Shows the block diagram of the proposed system.

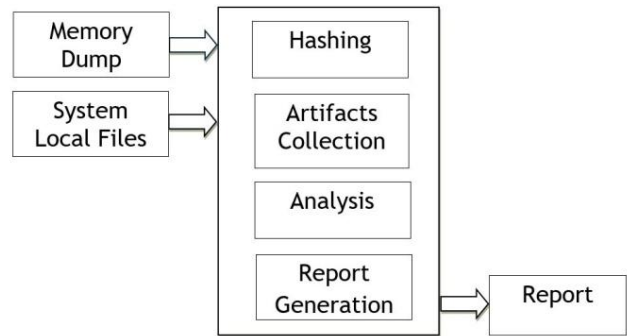


Fig 1. Proposed System

V. FORENSIC ANALYSIS OF I2P

I2P router software focuses heavily on network traffic reliability, rather than on locally stored data on participating I2P nodes. The local data is therefore stored without encryption and can be used by forensic analysts investigating a seized device [1]. Physical memory dump and system local files were analyzed in order to find out artifacts related to I2P.

A. Memory Analysis

Any information in RAM would possibly be lost once the device is turned off. Such information might include passwords, running processes, open sockets, contents of the clipboard, etc. All that information has to be collected before the system is powered down or transported. If the hard drive is encrypted, the capture of volatile data is even more crucial, since the hard drive information without it may not be available to the forensic investigator

In order to get I2P related artifacts the physical memory dump was taken using the OSForensics tool from Passmark software. Fig 2. shows the user interface of OSForensics tool.

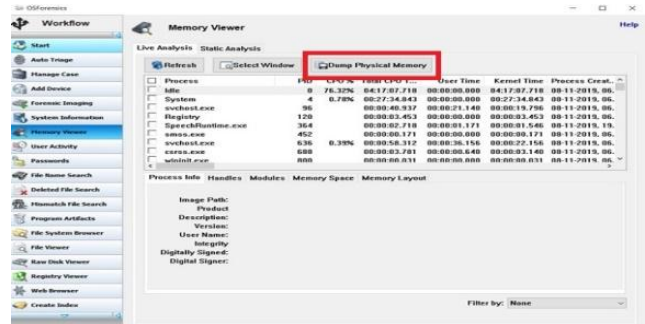


Fig. 2. User interface of OSForensics tool

The physical memory dump was analyzed using Volatility tool. Fig 3. Shows the user interface of Volatility tool.

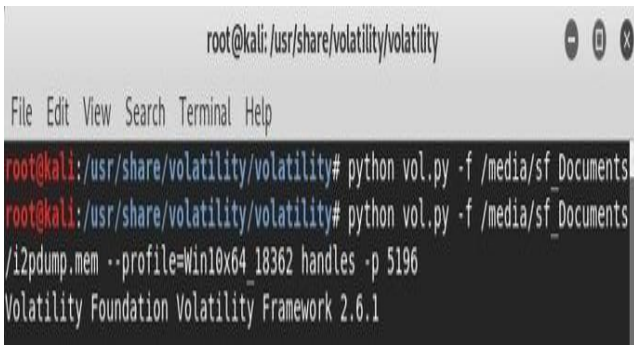


Fig 3. User interface of Volatility tool

B. System Local Files Analysis

a) *Addressbook*: The associations between an I2P domain name and I2P network identifier is stored in a file called the addressbook. In order to access an eepsite that is not present in the addressbook, the users have to manually add the eepsite address to the hosts.txt file or through an automatic subscription mechanism. The subscription details can be obtained from the addressbook which can be used in forensic investigations. The location of addressbook is C:\Users<username>\AppData\ Roaming\I2P\ addressbook as shown in the Fig. 4.

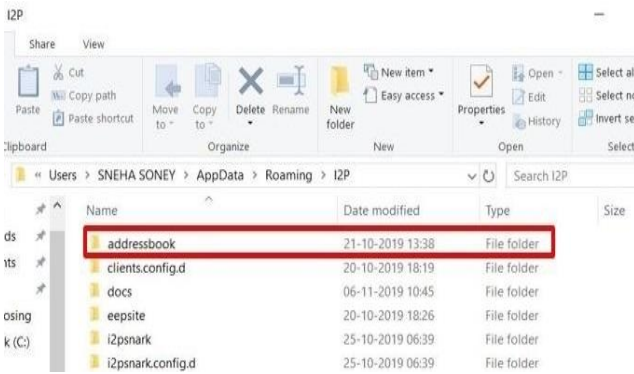


Fig 4. Location of Addressbook

The contents of addressbook is shown in the Fig. 5.

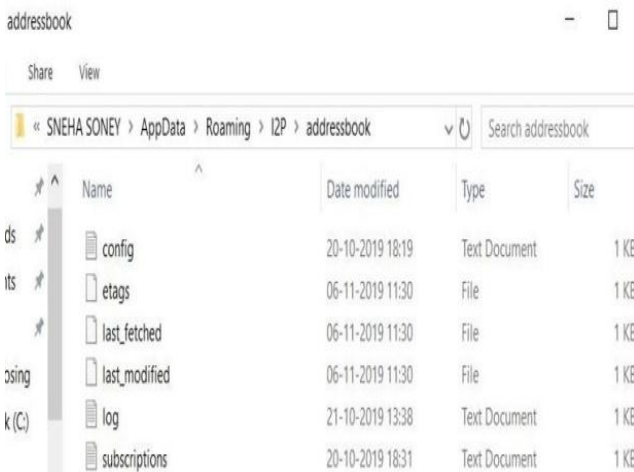


Fig 5. Addressbook Contents

b) *I2P Firefox Browser Profile Folder*: The I2P browser profile was downloaded from the official site to access I2P. The

location of profile folder is C: \ Users\< username >\AppData \ Local \I2PBrowser-Launcher\firefox.profile.i2p. as shown in the Fig. 6.

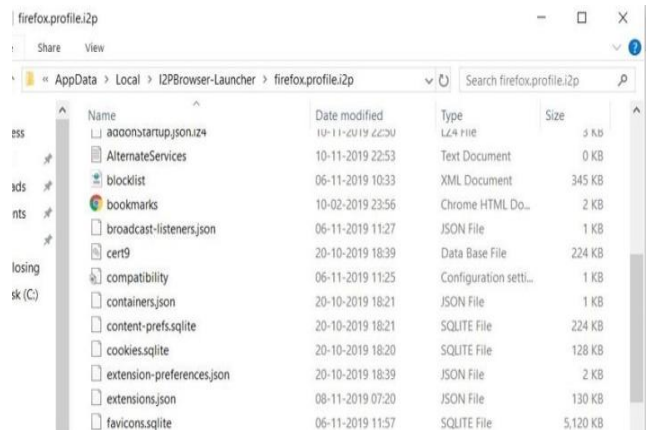


Fig 6. Location of I2P Firefox Browser Profile Folder

VI. ANALYSIS AND DISCUSSION OF RESULTS

Physical memory dump and system local files were analyzed in order to find out artifacts related to I2P. As a result process related information is obtained from the physical memory analysis and browsing related artifacts from the system local files analysis.

A. Analysis Results

In real world scenarios, knowledge about a router participating in an anonymization network is enough to undertake actions against citizens. The physical memory was analysed using the ‘Volatility’ tool. The results are as follows:

a) *pslist*: It will list all the running processes by following the EPROCESS list. The ‘I2Psvc.exe’ is listed in the output as shown in Fig. 7. I2Psvc.exe is usually located in the ‘C:\ Program Files\i2p’ folder. The .exe extension of a filename indicates an executable file. I2Psvc.exe is developed by Tanuki Software Ltd. and is a component of Java Service Wrapper Community.



Fig 7. Output of pslist

b) *pstree*: It will print the process list as a tree. Fig. 8. shows the output of pstree. ‘I2Psvc.exe’ and its child processes are listed in the output. Child processes are indicated using indentation and periods.



Fig 8. Output of pstree

c) *Psscan*: It is a pool scanner for process objects. This can find processes that previously terminated and processes that have been hidden or unlinked by a rootkit. The output of psscan includes 'conhost.exe' which is a child process of 'I2Psvc.exe', as shown in Fig. 9.

0x000000007012e400	firefox.exe	6220	2700	0x00000000140f5b002	2019-11-06 05:55:22 UTC+0000
0x000000007012e400	SystemSettings	10844	84	0x00000000793059002	2019-11-06 04:59:51 UTC+0000
0x000000007194c0	conhost.exe	6404	8764	0x000000001363c0002	2019-11-06 05:54:18 UTC+0000
0x00000000705cc100	svchost.exe	6580	876	0x00000000230709002	2019-11-05 18:22:07 UTC+0000

Fig 9. Output of psscan

d) *psxview*: It will find hidden processes with various process listings. It will give a comparison of various process enumeration techniques that will help in identifying variances. As shown in Fig. 10. 'I2Psvc.exe' is listed in the output of psxview.

0x000000001e03b3c0	svchost.exe	1736	True	True	True	False	True	True	False
0x000000001e5b4d380	svchost.exe	1380	True	True	False	False	True	True	False
0x000000001e1963c0	svchost.exe	2220	True	True	True	False	True	True	False
0x000000001516ce4c0	I2Psvc.exe	8764	True	False	False	False	True	True	False
0x00000000141a430c0	RAVBog64.exe	9404	True	False	False	False	True	True	False

Fig 10. Output of psxview

e) *sessions*: It will list the details of user logon sessions. The 'I2Psvc.exe' is listed in the output as shown in Fig. 11.

Process:	9360	Microsoft.Phot	2019-11-06 05:43:59 UTC+0000
Process:	9068	RuntimeBroker	2019-11-06 05:44:03 UTC+0000
Process:	8764	I2Psvc.exe	2019-11-06 05:54:18 UTC+0000
Process:	6404	conhost.exe	2019-11-06 05:54:18 UTC+0000
Process:	7592	java.exe	2019-11-06 05:54:19 UTC+0000

Fig 11. Output of sessions

f) *threads*: A thread is a path of execution within a process. A process can contain multiple threads. The output of threads list the associated threads of the I2P process as shown in the Fig. 12.

<p>ETHREAD: 0xffffd0097a171080 Pid: 8764 Tid: 10472</p> <p>Tags:</p> <p>Created: 2019-11-06 05:54:18 UTC+0000</p> <p>Exited: 1970-01-01 00:00:00 UTC+0000</p> <p>Owning Process: I2Psvc.exe</p> <p>Attached Process: I2Psvc.exe</p> <p>State: Waiting:DelayExecution</p> <p>BasePriority: 0x8</p> <p>Priority: 0x8</p> <p>TEB: 0x006e4000</p> <p>StartAddress: 0xfffff918cc53a61f0 UNKNOWN</p> <p>Win32Thread: 0xffffd0097a1d1650</p> <p>CrossThreadFlags: PS_CROSS_THREAD_FLAGS_DEADTHREAD</p> <p>0xfffff918cc53a61f0 98 CNDDE</p> <p>0xfffff918cc53a61f1 16 DB 0x16</p> <p>0xfffff918cc53a61f2 17 DB 0x17</p> <p>0xfffff918cc53a61f3 7a09 JP 0xfffff918cc53a61f0</p> <p>0xfffff918cc53a61f5 d0ff SAR BH, 0x1</p> <p>0xfffff918cc53a61f7 f5077 CALL OWORD [RAX+0]</p> <p>0xfffff918cc53a61fa 3ac5 CMP AL, CH</p> <p>0xfffff918cc53a61fc 8c91ffff0062 MOV [RCX+0x6200ffff] 8c91ffff0062</p> <p>0xfffff918cc53a6202 3ac5 CMP AL, CH</p> <p>0xfffff918cc53a6204 8c DB 0x8c</p> <p>0xfffff918cc53a6205 91 XCHG ECX, EAX</p> <p>0xfffff918cc53a6206 ff DB 0xff</p> <p>0xfffff918cc53a6207 ff DB 0xff</p> <p>ETHREAD: 0xffffd0097a0f080 Pid: 8764 Tid: 12026</p> <p>Tags:</p> <p>Created: 2019-11-06 05:54:18 UTC+0000</p>	<p>ETHREAD: 0xffffd00972114080 Pid: 8764 Tid: 6399</p> <p>Tags:</p> <p>Created: 2019-11-06 05:54:18 UTC+0000</p> <p>Exited: 1970-01-01 00:00:00 UTC+0000</p> <p>Owning Process: I2Psvc.exe</p> <p>Attached Process: I2Psvc.exe</p> <p>State: Waiting:MrQueue</p> <p>BasePriority: 0x8</p> <p>Priority: 0x8</p> <p>TEB: 0x006f0000</p> <p>StartAddress: 0xfffffb02ceb0 mtdll.dll</p> <p>Win32Thread: 0x00000000</p> <p>CrossThreadFlags: PS_CROSS_THREAD_FLAGS_DEADTHREAD</p> <p>0xfffffb02ceb0 4883ec78 SUB RSP, 0x78</p> <p>0xfffffb02ceb0 48bc99 MOV RAX, RCX</p> <p>0xfffffb02ceb0 48b0f0 MOV RAX, [RIP+0xf8032]</p> <p>0xfffffb02ceb0 4805c8 TEST RAX, RAX</p> <p>0xfffffb02ceb0 7a18 JZ 0xfffffb02ced3</p> <p>0xfffffb02ceb0 48bc22 MOV RB, R0X</p> <p>0xfffffb02ceb0 48 0b 0x48</p> <p>0xfffffb02ceb0 48 0b 0x80</p> <p>ETHREAD: 0xffffd009795e9080 Pid: 8764 Tid: 6580</p> <p>Tags:</p> <p>Created: 2019-11-06 05:54:18 UTC+0000</p>
---	---

Fig 12. Output of threads

g) *dlllist*: It will print the list of loaded DLLs. DLL stands for "Dynamic Link Library". A DLL (.dll) file contains a library of functions and other information that a Windows program can access. When a program is launched, static links or dynamic links to the necessary .dll files are created. Fig. 13. shows the output of the dlllist.

Base	Size	LoadCount	LoadTime	Path
0x00007ff60810000	0x90000	0xffff	2019-11-06 05:54:18 UTC+0000	C:\Program Files\I2P\I2Psvc.exe
0x00007ff607f0000	0x1f0000	0xffff	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\ntdll.dll
0x00007ff607e0000	0x20000	0xffff	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\KERNEL32.DLL
0x00007ff607d0000	0x220000	0xffff	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\KERNELBASE.dll
0x00007ff607c0000	0x6e5000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\SHELL32.dll
0x00007ff607b0000	0xfa000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\userbase.dll
0x00007ff607a0000	0x1b000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\UPDR.dll
0x00007ff60790000	0x4a000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\cfgmgr32.dll
0x00007ff60780000	0x29000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\shcore.dll
0x00007ff60770000	0x9e0000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\USER32.dll
0x00007ff60760000	0x120000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\USERPCRT.dll
0x00007ff60750000	0x336000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\combase.dll
0x00007ff60740000	0x80000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\bcryptPrimitives.dll
0x00007ff60730000	0x77f000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\windows.storage.dll
0x00007ff60720000	0x3e000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\usp10.dll
0x00007ff60710000	0x97000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\sechost.dll
0x00007ff60700000	0xa3000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\advapi32.dll
0x00007ff606f0000	0x1f000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\propapi.dll
0x00007ff606e0000	0x4a000	0x6	2019-11-06 05:54:18 UTC+0000	C:\WINDOWS\SYSTEM32\powerprof.dll

Fig 13. Output of dlllist

h) *handles*: It will print the list of open handles for each process. Fig. 14. shows the output of handles. A handle is an abstract reference to a resource in computer programming. Common resource handles includes file descriptors, network sockets, connections to database, process identifiers (PIDs), and job IDs.

Offset (v)	Pid	Handle	Access Type	Details
0xffffd0097f0b5e00	8764	0x4	0x1f0003	Event
0xffffd0097f0b5f00	8764	0x8	0x1f0003	Event
0xffffd0097f0b6000	8764	0xc	0x1	WaitCom...Packet
0xffffd0097f0b6100	8764	0x10	0x100020	IoCompletion
0xffffd0097f0b6200	8764	0x14	0x100ff	WorkerFactory
0xffffd0097f0b6300	8764	0x18	0x100002	IRTLiner
0xffffd0097f0b6400	8764	0x1c	0x1	WaitCom...Packet
0xffffd0097f0b6500	8764	0x20	0x100002	IRTLiner
0xffffd0097f0b6600	8764	0x24	0x1	WaitCom...Packet
0xffffd0097f0b6700	8764	0x28	0x004	EthRegistration
0xffffd0097f0b6800	8764	0x2c	0x004	EthRegistration
0xffffd0097f0b6900	8764	0x30	0x004	EthRegistration
0xffffd0097f0b6a00	8764	0x34	0x3	Directory
0xffffd0097f0b6b00	8764	0x38	0x1f0003	Event
0xffffd0097f0b6c00	8764	0x3c	0x1f0003	Event
0xffffd0097f0b6d00	8764	0x40	0x100020	File
0xffffd0097f0b6e00	8764	0x44	0x12019f	File
0xffffd0097f0b6f00	8764	0x48	0x12019f	File
0xffffd0097f0b7000	8764	0x4c	0x1f0001	ALPC
0xffffd0097f0b7100	8764	0x50	0x004	EthRegistration
0xffffd0097f0b7200	8764	0x54	0x004	EthRegistration
0xffffd0097f0b7300	8764	0x58	0x004	EthRegistration
0xffffd0097f0b7400	8764	0x5c	0x1f0003	IoCompletion
0xffffd0097f0b7500	8764	0x60	0x100ff	WorkerFactory
0xffffd0097f0b7600	8764	0x64	0x100002	IRTLiner
0xffffd0097f0b7700	8764	0x68	0x1	WaitCom...Packet
0xffffd0097f0b7800	8764	0x6c	0x100002	IRTLiner
0xffffd0097f0b7900	8764	0x70	0x1	WaitCom...Packet
0xffffd0097f0b7a00	8764	0x74	0x004	EthRegistration
0xffffd0097f0b7b00	8764	0x78	0x004	EthRegistration
0xffffd0097f0b7c00	8764	0x7c	0x004	EthRegistration
0xffffd0097f0b7d00	8764	0x80	0x1	Key
0xffffd0097f0b7e00	8764	0x84	0x1	Key

Fig 14. Output of handles

i) *shellbags*: It print the shellbags information. Shellbags facilitate track views, sizes and positions of a folder window once viewed through Windows Explorer. In order to uninstall I2P, an uninstaller program should be run. The access of the uninstaller through Windows Explorer will be listed in the shellbags output as shown in the Fig. 15.

Access Date	File Attr	Path
2019-10-20 12:36:10 UTC+0000	DIR	C:\Program Files\I2P\I2P Website
2019-10-20 11:51:10 UTC+0000	DIR	C:\Program Files\I2P\I2P Uninstaller

Fig 15. Output of shellbags

VII. CONCLUSION AND FUTURE SCOPE

I2P is an anonymous network that is built up over the internet. Due to the increased use of I2P in criminal activities there arise a need for methods that detect the presence of I2P in the host machine and the related artifacts. We conducted an in depth analysis of the physical memory dump and system local files in order to find out the presence of I2P in a Windows 10 machine. From the analysis, we obtained process related information from the memory dump and browsing related artifacts from the system local files. Furthermore, we are expecting the presence of more I2P related information from system files like event logs and from network traffic analysis.



REFERENCES

1. Maxim Wilson, Behnam Bazli, "Forensic analysis of I2P activities," 22nd International Conference on Automation and Computing (ICAC), 2016
2. Juan Pablo Timpanaro, Isabelle Chrisment, Olivier Festor," Group-Based Characterization for the I2P Anonymous File-Sharing Environment", 6th International Conference on New Technologies, Mobility and Security (NTMS), 2014
3. Juan Pablo Timpanaro, Isabelle Chrisment, Olivier Festor,"A Bird's Eye View on the I2P Anonymous File-sharing Environment", The 6th International Conference on Network and System Security, Nov 2012, Wu Yi Shan, China, 2012
4. Khalid Shahbar ,A. Nur Zincir-Heywood,"Effects of Shared Bandwidth on Anonymity of the I2P Network Users.", 2017 IEEE Symposium on Security and Privacy Workshops,2017
5. Yue Gao, Qingfeng Tan, Jinqiao Shi, Xuebin Wang, Muqian Chen, "Large-Scale Discovery and Empirical Analysis for I2P EepSites", 2017 IEEE Symposium on Computers and Communications (ISCC), 2017
6. M. Ehlert, "I2P usability vs. Tor usability: a bandwidth and latency comparison", [online] Available: <http://userpage.fu>.
7. Preeti S. Joshi1,Dinesha H.A, "Study Report of existing forensic tools and technologies to identify Darknet", IJCSE 2018
8. Dr. Digvijaysinh Rathod," Darknet Forensics", International Journal of Emerging Trends Technology in Computer Science (IJETTCS),2017
9. Afzaal Ali, Maria Khan, Muhammad Saddique, Umar Pirzada, Muhammad Zohaib, Imran Ahmad, Narayan Debnath," TOR vs I2P: A Comparative Study",IEEE, 2016
10. <https://geti2p.net/en/>
11. <https://en.wikipedia.org/wiki/I2P>

AUTHORS PROFILE



Sneha Soney, is currently doing her Masters Degree in Cyber Forensics and Information Security at ER & DCI Institute of Technology, Thiruvananthapuram, Kerala, India. She has a BTech in Computer Science and Engineering. Currently, she is pursuing an internship at H&R Block (India) Pvt Ltd. Her areas of interest include Cyber Forensics and Cyber Security.



C Balan, is currently working as Associate Director in C-DAC, Thiruvananthapuram. He possesses 21 years of experience in Cyber Forensics and Artificial Intelligence. He led a team of scientists in designing and developing indigenous cyber forensics software tools for disk imaging, data recovery and analysis and data integrity checking. He has provided technical support to various Law enforcement agencies for cyber-crime investigation. Authored number of International and national technical papers on cyberforensics.



Priya P Sajan, is currently working as Project Engineer in C-DAC, Thiruvananthapuram, Kerala, India. She has 10 years of experience in Cyber Forensics and Cyber Security. She is pursuing her PhD in Parallelization of Image Segmentation in Multicore Systems. She had published 9 papers in various Scopus indexed journals.



Elizabeth Rose Lalson, is currently working as Assistant Professor in ER&DCI Institute of Technology, C-DAC Thiruvananthapuram, Kerala, India. She pursued her BTech and MTech from Cochin University of Science and Technology. She has over 7 years of experience in Teaching. Published about seven papers in national and international journals and conference. Her areas of interest include cryptography, cyber forensics and cyber security.