On the Memory Artifacts of the Tor Browser Bundle

Atta Al-Khaleel, Duaa Bani-Salameh, Mohammed I. Al-Saleh

Jordan University of Science & Technology
Computer Science Dept.
P.O. Box 3030
Irbid, Jordan 22110
ayalkhaleel12@cit.just.edu.jo, dabanisalameh12@cit.just.edu.jo, misaleh@just.edu.jo

Abstract

Tor is one of the most famous privacy-preserving tools. It creates virtual encrypted tunnels to convey users' data. Tor users improve their privacy against people watching their activities or doing traffic analysis. They enjoy being anonymous while browsing the web or chatting with friends. Tor is being used by variety of people ranging from ordinary individuals to journalists or even governmental organizations. This paper investigates the memory artifacts of the Tor browser Bundle, which is specifically pre-configured to use the Tor network. Although it is hard to analyze Tor's data while being in transit, information is fully exposed in the clients' machines after delivery. That is all data must go through the memory before being processed. We use Tor browser in different experiments to check the possibility of recovering data remnants from the memory. This paper shows that Tor works pretty good in destroying the involved data prior getting closed.

Keywords. Tor network, Memory artifacts, Privacy.

1 Introduction

The Internet has become an essential part of the people's daily life. It is being used in many life aspects such as web browsing, instant messaging, online shopping and banking, communication, social networking, and emailing. The networks are untrusted in the sense that data while being transferred can be intercepted by third parties. So, it is well-known that the Internet is insecure and it is up to the communicating parties how they secure their channels. The three security goals that must be achieved for a system to be secure are: confidentiality, integrity, and availability. Confidentiality simply means that data can only be read by the authorized parties. Integrity means that data can only be modified by the authorized parties. Availability means that the system must guarantee certain level of timely responsiveness. Although privacy somehow is related to security (confidentiality in specific), it also means that third parties should not be able to bind actions or information to a specific person. Given that, privacy and anonymity are also related.

People spend a lot of their time in browsing the web and chatting with friends. They do not like others to know what the websites they are visiting, the videos they are watching, the items they are buying, or the messages they are sending. Furthermore, in many situations, people do not like to expose their physical locations where their activities originate from. Many parties are interested in what people are doing. This includes governments who want to understand people's political trends, companies who want to understand people's shopping trends or habits, or even malicious attackers whose goals are to extort innocents. Some users give up on that while others still seek their privacy. Tor is one of the most famous privacy-preserving tools. In Tor, encrypted data travel through several Tor nodes (or relays) before reaching the ultimate destination, making it hard for eavesdroppers or network traffic analyzer to read the data or even trace it to its origins.

All information need to go through the memory before and after being processed. Consequently, a lot of sensitive and private information, such as encryption keys and viewed images, might be found there if not explicitly destroyed. Some works on memory artifacts have been done [12, 24, 20, 26, 3]. Memory information should be given a special attention as some information can only be found in the memory and never goes to permanent storage. This paper only investigates the memory artifacts of the Tor Browser Bundle.

This paper is organized as follows. In Section 2, we give a brief overview of the Tor network and present our investigation model. This is followed by Section 3 that explains our experimental setup. Our results are shown in Section 4. A discussion and future work are covered in Section 5. This is followed by related work and the conclusion.

2 Tor network and investigation model

This section illustrates how Tor generally works. Furthermore, we highlight our investigation model.

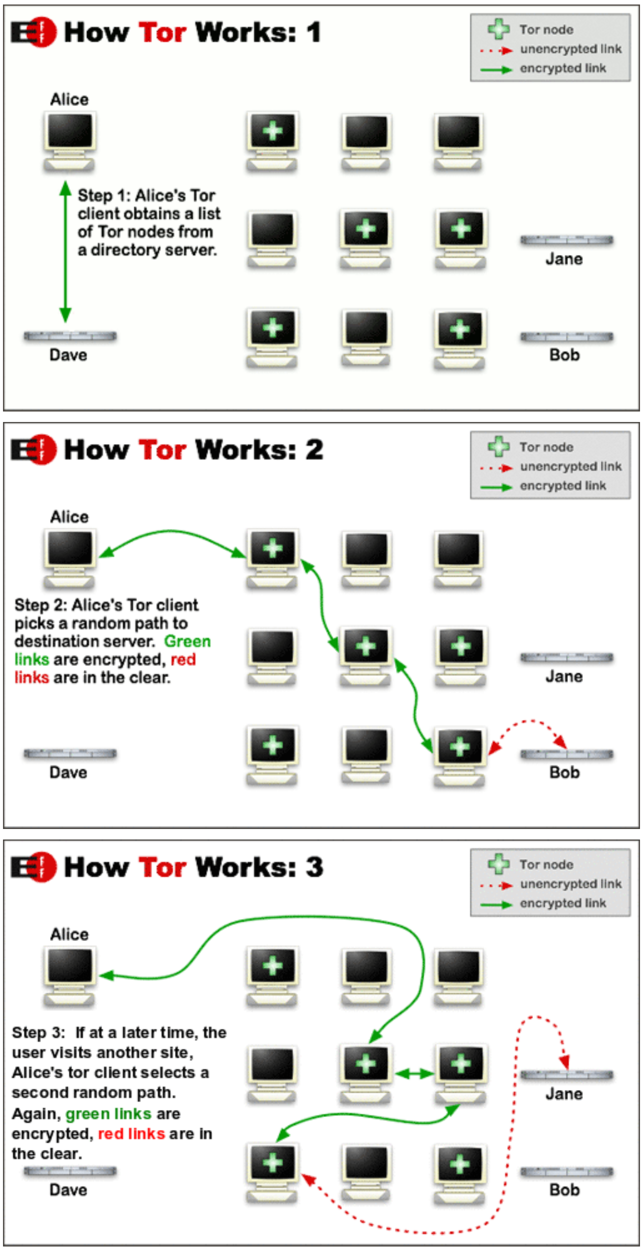
2.1 Tor network

Tor is specifically designed to enable users to enhance their privacy while using the public networks. Users can browse the web and chat with friends without having their privacy compromised to third parties or even to the servers they are communicating with. Network traffic analyzers can sniff people's packets and know accordingly who is talking to whom, at what time, for how long, and what kind of information they are exchanging. Encrypting the payloads of the packets does hide the contents, but it does not hide the headers' information which includes the sender's and receiver's identities. Because Tor utilizes the encryption techniques to create tunnels and incorporates indirect server communication, Tor users can even bypass the Internet Service Provider's blocked websites.

In order to disperse the observers' capabilities to track users, Tor routes the traffic through different relays (called Tor nodes) before reaching to the ultimate destination. No single node at any point can bind a sender to a receiver. Figure 1 shows how Tor works. First, the user gets the list of the Tor relays from the directory server. Next, the user chooses three random relays among the list through which the traffic will go. Finally, new routes are to be taken upon new connections. No observers nor malicious Tor nodes does know the user's complete route nor they know who is doing what.

2.2 Investigation model

Figure 2 shows our investigation model. A Tor Browser Bundle's user connects to the Internet and does several activities, such as viewing images and watching videos. This paper tries to find any memory data remnants after conducting such activities. These remnants (if exist) might threaten the user's privacy if captured by an adversary.



Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.

Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

Step 3: If at a later time, the user visits another site, Alice's tor client selects a second random path. Again, green links are encrypted, red links are in the clear.

Figure 1: How Tor works (*the figure is taken from www.torproject.org*).

3 Experimental setup

We design our experiments to answer the following question: **Does Tor Browser Bundle leave data in memory which can be utilized by adversaries in breaching users' privacy?**.

To answer the above question, we designed several experiments. Figure 3 shows our setup for the experiments. We use a Windows 7 virtual machine (VM) with 512MB of RAM. The host machine runs Windows 7.

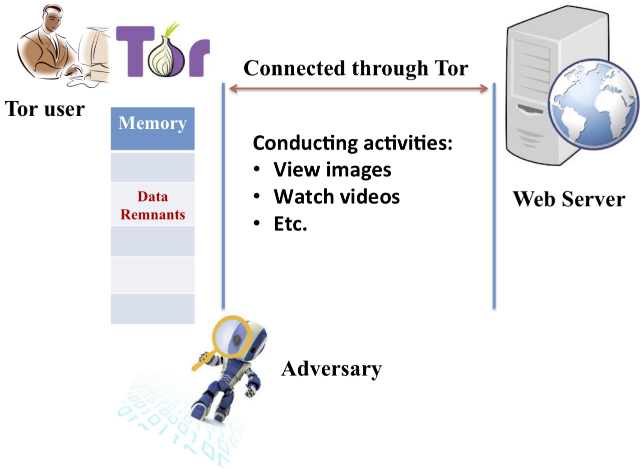


Figure 2: Investigation model.

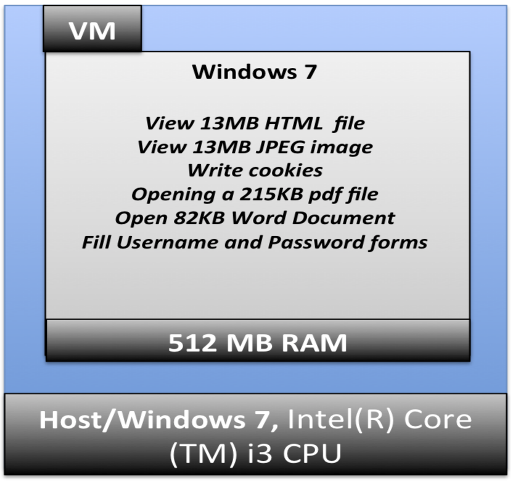


Figure 3: Experimental setup.

3.1 Experiments

The following activities are tested against possible artifacts:

- Viewing a 13MB HTML file: in this experiment we open a web page with 985 different sentences. We will be looking for these sentences in the memory.
- Viewing a 13MB JPEG image: here we view an online image and examine the memory to check if the image stick there.
- Writing cookies: we open a website that writes 100 cookies in the client's side. We want to check the cookies' persistency in the memory.
- Opening a 215KB pdf file: a pdf file as an email attachment is viewed with the email's online viewing facility. The file contains 985 sentences which we will be looking for in the memory.
- Opening a 82KB Word Document: a word document as an email attachment is viewed with the email's online viewing facility. The file contains 985 sentences which we will be looking for in the memory.
- Filling in username and password in the Hotmail web site: a username and password are filled and submitted. We want to check them in the memory.

3.2 Experiments procedure

Here is the procedure we followed in all the experiments:

- Run Tor Browser Bundle.
- Conduct an activity (from the above mentioned ones) on a separate browser Tab.
- Dump the memory while the Tab is active/open.
- Dump the memory after closing the Tab.
- Dump the memory after closing the whole Tor browser.
- Dump the memory after 15 minutes of closing the browser.
- Restart the machine after each activity.

According to the procedure above, we will get 4 memory dumps for each activity. We will search for the artifacts in these memory dumps to check if we can find any. Python scripts have been created to search for such artifacts.

4 Results

In this section, we present our results for the experiments discussed in Section 3. The tables below have the following abbreviations: TbOp stands for Tab Open, TbCl stands for Tab Closed, TrCl stands for Tor Closed, and 15Min Ltr stands for 15 Minutes Later.

Tables 1 through 6 show the results for each experiment separately. All the results are consistent with the following conclusion: all valuable artifacts can only be recovered while Tor and its browser are open. Once the browser's tabs and Tor are closed, all data remnants are cleared by Tor. This conclusion states that **Tor not only enhance your**

privacy from the network eavesdroppers, but also from machines' intruders.

Table 1: Viewing the HTML with 985 sentences experiment.

	TbOp	TbCl	TrCl	15Min Ltr
Sentences	985	977	263	0
Website URL	10	9	0	0
Tor URL	20	19	6	0

Table 2: Viewing the JPEG image experiment.

	TbOp	TbCl	TrCl	15Min Ltr
Image	all found	0	0	0
Website URL	22	16	0	0
Tor URL	29	32	4	2

Table 3: Writing cookies experiment.

	TbOp	TbCl	TrCl	15Min Ltr
Cookies	100	17	0	0

Table 4: Opening a pdf file with 985 sentences experiment.

	TbOp	TbCl	TrCl	15Min Ltr
PDF contents	982	498	0	0
Website URL	764	2	1	0
Tor URL	1	1	1	0
File name	54	58	47	0

Table 5: Opening a word document with 985 sentences experiment.

	TbOp	TbCl	TrCl	15Min Ltr
File contents	533	99d	0	0
Website URL	1025	965	0	0
Tor URL	38	38	4	4
File name	11	91	0	0

Table 6: Filling forms experiment.

	TbOp	TbCl	TrCl	15Min Ltr
Username	6	4	0	0
Password	0	0	0	0
Website URL	30	33	0	0

5 Discussion and future work

This paper examines only one aspect of Tor, which is the memory data remnants of the Tor Browser Bundle. Examining other aspects of Tor (such as the effect of the browser's plug-ins) that might expose users' privacy is a future direction. Also, testing other privacy-preserving tools other than Tor is a future work.

Even though the RAM memory is volatile (*i.e.*, information could be vanished after restarting or shutting down the machine), its contents is very precious. All kinds of data do go through the memory before and after processing, and sometime, they never go to a permanent storage.

6 Related work

Several works have been conducted to measure Tor's security, privacy, and performance [21, 5, 13, 15, 23, 17].

Searching memory for data remnants have been studied from both the security and forensics perspectives [12, 24, 20, 26, 22, 7, 11, 10, 8, 6, 19, 25, 3, 4, 2].

Web browsers have paid attention to the privacy of their users through providing the private browsing mode. Mozilla Firefox's private browsing can be reached from the "New Private Window" command. Internet Explorer has the "InPrivate Browsing" mode. Google Chrome has the "New incognito window". Safari includes the "Private Browsing" mode. Several works have studied the private browsing in these browsers [14, 18, 1, 9, 16].

7 Conclusion

The privacy of the Internet users is threatened by many third parties and thus becomes an essential requirement. Eavesdroppers and traffic analyzers are there to monitor users' activities for variety of reasons. Tor comes in to enable its users to practice their activities while preserving their privacy. Tor uses many relays to create virtual encrypted tunnels through which users cannot be linked to their activities and thus stay anonymous. This paper examines one aspect of Tor: the memory artifacts of its browser bundle. We conducted several experiments to check what artifacts the Tor browser might leave in memory. This paper shows that Tor destroys all in-memory valuable information and thus Tor users can enjoy Tor securing their privacy.

References

[1] G. Aggarwal, E. Bursztein, C. Jackson, and D. Boneh. An analysis of private browsing modes in modern browsers. In *USENIX Security Symposium*, pages 79–94, 2010.

[2] M. Al-Saleh and Z. Al-Sharif. Ram forensics against cyber crimes involving files. In *The Second International Conference on Cyber Security, Cyber Peacefare and Digital Forensic (CyberSec2013)*, pages 189–197. The Society of Digital Information and Wireless Communication, 2013.

[3] M. I. Al-Saleh and Z. A. Al-Sharif. Utilizing data lifetime of tcp buffers in digital forensics: Empirical study. *Digital Investigation*, 9(2):119 – 124, 2012.

[4] M. I. Al-Saleh and Y. A. Forihat. Skype forensics in android devices. *International Journal of Computer Applications*, 78, 2013.

[5] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-resource routing attacks against tor. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 11–20. ACM, 2007.

[6] P. Broadwell, M. Harren, and N. Sastry. Scrash: a system for generating secure crash information. In *Proceedings of the 12th conference on USENIX Security Symposium - Volume 12*, SSYM'03, pages 19–19, Berkeley, CA, USA, 2003. USENIX Association.

[7] J. Chow, B. Pfaff, T. Garfinkel, K. Christopher, and M. Rosenblum. Understanding data lifetime via whole system simulation. In *Proc. 13th USENIX Security Symposium*, August 2004.

[8] J. Chow, B. Pfaff, T. Garfinkel, and M. Rosenblum. Shredding your garbage: reducing data lifetime through secure deallocation. In *Proceedings of the 14th conference on USENIX Security Symposium - Volume 14*, SSYM'05, pages 22–22, Berkeley, CA, USA, 2005. USENIX Association.

[9] D. Dayalamurthy. Forensic memory dump analysis and recovery of the artefacts of using tor bundle browser–the need. 2013.

[10] D. Engler, D. Y. Chen, S. Hallem, A. Chou, and B. Chelf. Bugs as deviant behavior: a general approach to inferring errors in systems code. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*, SOSP '01, pages 57–72, New York, NY, USA, 2001. ACM.

[11] T. Garfinkel, B. Pfaff, J. Chow, and M. Rosenblum. Data lifetime is a systems problem. In *Proceedings of the 11th workshop on ACM SIGOPS European workshop*, EW 11, New York, NY, USA, 2004. ACM.

[12] H. Inoue, F. Adelstein, and R. A. Joyce. Visualization in testing a volatile memory forensic tool. *Digital Investigation*, 8(Supplement):S42–S51, 2011.

[13] K. Loesing, W. Sandmann, C. Wilms, and G. Wirtz. Performance measurements and statistics of tor hidden services. In *Applications and the Internet, 2008. SAINT 2008. International Symposium on*, pages 1–7. IEEE, 2008.

[14] A. Mahendrakar, J. Irving, and S. Patel. Forensic analysis of private browsing mode in popular browsers.

[15] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of tor. In *Security and Privacy, 2005 IEEE Symposium on*, pages 183–195. IEEE, 2005.

[16] D. Ohana and N. Shashidhar. Do private and portable web browsers leave incriminating evidence?: a forensic analysis of residual artifacts from private and portable web browsing sessions. *EURASIP Journal on Information Security*, 2013(1):1–13, 2013.

[17] L. Overlier and P. Syverson. Locating hidden servers. In *Security and Privacy, 2006 IEEE Symposium on*, pages 15–pp. IEEE, 2006.

[18] H. Said, N. Al Mutawa, I. Al Awadhi, and M. Guimaraes. Forensic analysis of private

browsing artifacts. In *Innovations in Information Technology (IIT), 2011 International Conference on*, pages 197–202. IEEE, 2011.

[19] J. Sammons. *The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics.* Elsevier, 2012.

[20] A. Schuster. The impact of microsoft windows pool allocation strategies on memory forensics. *Digital Investigation*, 5, Supplement(0):S58 – S64, 2008. The Proceedings of the Eighth Annual DFRWS Conference.

[21] B. Shebaro, F. Perez-Gonzalez, and J. R. Crandall. Leaving timing-channel fingerprints in hidden service log files. *digital investigation*, 7:S104–S113, 2010.

[22] M. Simon and J. Slay. Recovery of skype application activity data from physical memory. In *ARES*, pages 283–288, 2010.

[23] R. Snader and N. Borisov. A tune-up for tor: Improving security and performance in the tor network. In *NDSS*, volume 8, page 127, 2008.

[24] J. Solomon, E. Huebner, D. Bem, and M. Sze?žynska. User data persistence in physical memory. *Digital Investigation*, 4(2):68 – 72, 2007.

[25] R. M. Stevens and E. Casey. Extracting windows command line details from physical memory. *Digital Investigation*, 7, Supplement(0):S57 – S63, 2010. ¡ce:title¿The Proceedings of the Tenth Annual {DFRWS} Conference¡/ce:title¿.

[26] A. Walters and N. L. Petroni. Volatools : Integrating volatile memory forensics into the digital investigation process. *Digital Investigation*, pages 1–18, 2007.