Darknet Security: A Categorization of Attacks to the Tor Network

Enrico Cambiaso[1], Ivan Vaccari[1], Luca Patti, and Maurizio Aiello[1]

Consiglio Nazionale delle Ricerche (CNR-IEIIT), Genoa, Italy
{name.surname}@ieiit.cnr.it

Abstract. In the darknet security topic, it is important to analyze the threats that characterize the network. This paper deeply investigates the literature of attacks against the Tor network, presenting the most relevant threats in this context. In order to provide an important tool for the research community, we propose an exhaustive taxonomy based on the target of the attack. Such taxonomy represents a characterization scheme to identify cyber-attacks related to darknet environments and better understand their functioning. The proposed work should therefore be considered an important step forward in the darknet security field.

Keywords: darknet · cyber-security · tor · onion network · taxonomy.

1 Introduction

In the communication era, the global Internet network represents a fundamental resource for everyday live. Security aspects on the Internet assume today a very important role [1]: being a crucial element for users activities, governments, and critical infrastructure systems, the Internet network has to be kept a safe place for its users and inter-communicating systems, ensuring secure communications and guaranteeing users rights. In the privacy context, it is important to ensure hiding capabilities for both the content exchanged between two entities and the identity of the entities themselves [2].

Anonymity network systems were primarily designed to preserve communications privacy to censored Internet users. Anonymity is achieved by embedding user data inside of different layers of encryption and by forwarding the traffic through a set of relay/routing nodes or proxies [3]. Onion networks [2, 4] represent today one of the available solutions adopted in this context. Such networks are based on onion routing approaches, involving encryption procedures making routing nodes unable to read exchanged data between two (client and server) entities. There exist several different anonymizing networks [5], such as Freenet [6], I2P [7], [8], MorphMix [9], Hornet [10] or Tarzan [11]. Nevertheless, nowadays, the most adopted onion network is Tor [5].

Representing the second version of the original Onion Routing protocol [2, 4], the Tor network (Tor in the following) is today considered one of the most popular network protocol for anonymous communications. Developed starting from an internal project of the United States Naval Research Lab, hence inherited by the Tor non-profit organization in late 2003, Tor was created to improve

CEUR-WS.org/Vol-2315/paper10.pdf

privacy and security of Internet users. Tor rapidly acquired adoption on the Internet: while on January 2010, about 1,000 Tor public relays was distributed around the world, this number quickly raised to nearly 8,000 on January 2015, and is nowadays stabilized to around 7,000 nodes. In virtue of its effectiveness, the anonymity levels provided by Tor are often uncomfortable to law enforcement or governments prone to Internet censorship activities. This statement is confirmed by a July 2014 competition organized by the Russian government, giving a 110,000 USD price to any Russian citizen breaking the Tor network for tracking purposes [1].

Due to the adoption of the Tor network and the nature of exchanged contents, it is important to deeply explore the network, its functioning and the associated weaknesses, in the darknet security context. In this paper, we analyze cyber-attacks on the Tor network, by proposing an exhausting taxonomy of available attacks, by analyzing the target of the attack. Although other works propose a survey of Tor attacks [12], the proposed work reports a broader set of threats, also proposing a categorization of them. Particularly, we report the functioning of the Tor network in Section 2, hence categorizing and describing available attacks in Section 3. Section 4 reports instead considerations on the analyzed attacks, while, finally, Section 5 reports the conclusions of the work.

2 The Tor Protocol

Tor can be adopted in order to hide the identity of the client while surfing on the surface web (including websites reachable through a common browser) [13], or while accessing hidden services on the Tor network [14]. Considering the first scenario, accordingly to Figure 1, each communication involves several public relay nodes: (i) the client, (ii) the server, (iii) a Tor entry node (or guard node), (iv) a Tor exit node, and (v) a set of Tor middle nodes greater or equal to one. Since a single middle node is usually adopted [3] (as depicted in Figure 1), we will consider such scenario in the following.

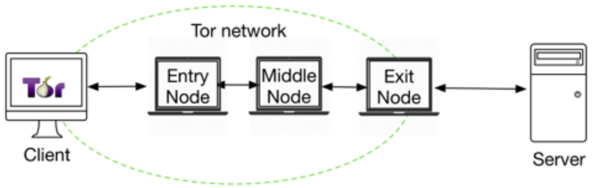

Fig. 1. Tor communication sample

[1] ”Putin Sets $110,000 Bounty for Cracking Tor as Anonymous InternetUsage in Russia Surges” (Accessed on Nov 6th, 2018): http:www.bloomberg.comnews2014-07-29putin-sets-110-000-bounty-for-cracking-tor-as-anonymous-internet-usage-in-russia-surges.html

Tor also supports additional guard nodes known as bridge nodes [15]: unlike public relay nodes, in this case the identity/IP address of the node is not public. Also, the Tor network can be extended by any host wishing to become part of the network. Although this characteristic makes Tor an extremely scalable network, it also makes it vulnerable by design to man-in-the-middle attacks, especially in case a malicious user controls the exit node of the network and communications with the server are not encrypted [16].

The set of Tor nodes involved in the communication is chosen by the client itself and it represents a Tor circuit, build by the client at the begin of the connection by communicating with the network nodes belonging to the circuit and exchanging a separate encryption key with each node. Messages are encrypted by the client through an onion encryption scheme [17]. Since each node involved in the communication is the only one able to decrypt the message it receives, each node is only aware of the identity of its predecessor and successor on the circuit. In addition, while the entry/guard node is the only one directly communicating with the client, only the exit node is able to read the original message (to be delivered to the server on the surface Internet network), that can be encrypted if encryption algorithms are adopted (such as SSL).

3 Attacks to the Tor ecosystem

During the years, several studies have been conducted on onion networks [2, 18–20], with particular focus on the Tor network, that is the most adopted one. Considering onion networks security, attacks may target three different entities of the network:

- Client: in this case, the aim of the attacker is to target a client of the Tor network;
- Server: in this case, the Tor hidden server is targeted by the attacker;
- Network: in this case, the Tor network itself is targeted by the attacker.

Concerning such different entities, we now describe available attacks, by focusing on the Tor onion network. We also analyze attacks targeting generic/mixed entities.

3.1 Attacks to the client

Several studies have been conducted during the years in order to hack the Tor network and de-anonymize its users, associating their IP address to an outgoing packet [21, 22]. These studies often led to concrete attacks exploiting specific vulnerabilities, removed by Tor developers through appropriate updates of the browser software. In this part of the paper we report attacks aimed at creating a damage to the Tor client.

Plug-in based attacks These attacks are executed to target the user through the Internet browser he adopts to navigate on the network. Such threats make use of external software plugged into the browser (plug-in), such as Flash, Java, and ActiveX Controls [23, **?**]. These applications run as separate software, executed with users permissions on the operating system. Some of these technologies, like Java or Adobe Flash, for instance, are executed in apposite virtual machines or frameworks bypassing proxy configuration settings adopted by the Tor browser, hence directly communicating on the Internet network, without making use of the Tor ecosystem. Browser attacks may be implemented by following different approaches [24]: (i) by operating on the public Internet server contacted by the client, through a malicious server system embedding, for instance, Adobe Flash contents on the web page; (ii) by adopting an evil exit node, intercepting users communications on not encrypted channels (e.g. clear HTTP connections) and embedding malicious plug-in related contents. Due to the possible exposure of the clients identity deriving from the usage of browser plug-ins, as suggested by anonymizing browsers themselves, such technologies, often disabled by default, should be avoided in order to communicate on the onion network anonymously and safely.

Torben attack The Torben attack [25] is executed to identify a Tor client, by manipulating web pages to force the user to access content from untrusted sources and by exploiting low-latency characteristics of anonymization networks to infer indicators of web pages that are transmitted, hence retrieving information on the web pages the client visits through Tor.

P2P Information leakage This kind of attack is perpetrated in order to de-anonymize Tor clients by exploiting their connections to peer-to-peer systems. Indeed, considering for instance the BitTorrent protocol [26], it is possible for a malicious user to retrieve the IP address of a client connecting over Tor to communicate with the torrent tracker. A torrent tracker is a network service the client has to communicate to retrieve information about the list of peers able to share the requested resource [16]. Peers information are provided as couples of IP address and listening port.

The attacker exploits in this case the fact that, although the list of trackers may be retrieved anonymously via Tor, P2P connections are often accomplished unsafely, by directly communicating with the peer. Therefore, it is possible for the attacker to exploit the man-in-the-middle addiction of the Tor network to alter the content of the list returned by the torrent tracker, by including into the list the IP address of a malicious torrent peer. Since communication with such peer would not be established through Tor, it is possible for the attacker to retrieve the IP address of the client originating the request to the tracker [26].

Induced Tor guard selection The Tor entry node is the only node directly communicating with the client. Nevertheless, since Tor packets payload is encrypted, it is not possible for the entry node to retrieve the clean content of exchanged messages without knowing the decryption keys of the circuit nodes. Therefore, although a single malicious guard node may not compromise the communication, it may be required to the attacker to own the entry node of a Tor circuit [4].

In order to induce a Tor client to adopt a specific malicious entry node, it is possible to drop communications of the client to public entry nodes, except the attackers ones [27]. This operation can be accomplished, for instance, by altering traffic capabilities of the victim, blocking connections to legitimate entry nodes at the network level through appropriate policies, defined, e.g., by network administrators or local Internet Service Providers.

Raptor Routing attacks on privacy in Tor (RAPTOR) [28] is a suite of attacks that can be launched by the Autonomous System (AS) [22] to deanonymize clients. One of the attacks is based on traffic analysis of asymmetric communications that characterize the network. Another attack exploits the natural churn in Internet routing and BGP paths to accomplish traffic analysis. Finally, the last attack is based on Internet routing manipulation through BGP hijacking activities, accomplished to discover users' Tor guard nodes.

Unpopular ports exploitation This attack exploits the fact Tor exit nodes often limit the range of ports they can connect to on the public surface Internet [24]. The attack attempts to retrieve clients identity by making use of a set of malicious entry nodes and a set of malicious exit nodes. Exit nodes controlled by the attacker support communications on unpopular ports. It is also required to the attacker to control the service host contacted by the client [3]. The attacker aim is to induce the client to create a Tor circuit through an entry node and an exit node under the control of the malicious user. Such configuration would allow the attacker to retrieve the identity of the Tor client, for instance through traffic correlation techniques [29].

In order to perpetrate the attack, the malicious user injects a script into the web page requested by the client, thus inducing the browser to open a connection to an Internet service listening on unpopular port. Such connection is established through the Tor network. This behaviour will induce the client to create a Tor circuit allowing communication on the specified unpopular port. Since the attacker controls a set of exit nodes supporting such communication, the probability to control the exit node of the circuit increases.

Low-resource routing attacks In order to perpetrate such attack, the adversary has to enroll or compromise some high-bandwidth, high-uptime Tor routers [5]. By assuming such compromising, the attacker can decrease the resource requirements of the malicious node, by using low-bandwidth connections, hence exploiting the possibility of a node to report incorrect bandwidth values. Since this advertisement is not verified by trusted directory servers [23], the relay node appears to have a high-bandwidth and its chance to be chosen for a circuit is particularly high[2]. In case both the entry and the exit nodes in a circuit are

[2] Currently, this approached is no longer possible to adopt, since directory servers control the effective bandwidth declared.

compromised, all information received may be logged and processed, for instance through traffic correlation approaches, to reveal the IP address of the client.

3.2 Attacks to the server

In this kind of threats, the purpose of the attacker is to target the hidden service, in order to reveal its identity or to weaken it. Indeed, as previously mentioned, the Tor network can be adopted in order to access services both on the public surface Internet and Tor (hidden services). In the latter case, the identity of the service is unknown to the client [5]. Concerning attacks whose purpose is to reveal the hidden service IP address, the following assumptions may be required [12]: (i) the attacker has to impersonate a malicious client and a guard node; (ii) the hidden service is forced to choose a compromised guard node as entry node. Different attacks to hidden services are available.

Cell counting and padding During such attacks [12, 30], the hidden service is forced to establish a connection to a malicious rendezvous point. The attacker sends a specifically crafted Tor cell/packet to the introduction point of the hidden service, specifying the chosen rendezvous point [31]. Hence, the introduction point forwards the message to the hidden service, that is induced to build a Tor circuit to reach the (malicious) rendezvous point. When the rendezvous point receives the message (containing some sort of cookie/token generated from the client), it is designed to send a specific number (50) of padding cells to the hidden service, by using the same circuit. Such padding cells, supported by the protocol and discarded by the hidden service, simplifies the generation of a signatures on the traffic [24]. At this point, the rendezvous point terminates/closes the circuit. The entry node, supposed to be controlled by the attacker, monitors the traffic of the circuits that pass through it. If it receives a cell including the circuit closure, it will verify that such reception occurs after the reception of the cell containing the confirmation cookies, and that the number of past cells is 3 cells up through the circuit and 53 down through the circuit. If these conditions are met, the attacker can deduce that the guard node he owns was chosen from the hidden service, hence, it is possible for the attacker to retrieve the IP address of the hidden service.

Tor cells manipulation By manipulating Tor cells/packets it is possible to locate a targeted hidden service [12]. Particularly, when the client sends a cell to a hidden service to initiate the communication, the request is ”proxed” by the rendezvous point, that is assumed to be controlled by the attacker. Such condition provides to the malicious user the ability to detect the request and apply minor changes to the message/cell data (even a single bit may be changed, hence making the cell not compliant to the protocol), thus forwarding the message to the hidden service and simultaneously sending a timestamp of the modified cell to a central server under the control of the attacker. The cell may not be recognized as an intact cell from the hidden service, that would send back a destroy message to the client. This message, directed from hidden service to the client, is designed to pass from hidden services entry node (controlled by the attacker) first, that may send to the central server some cells information like the command specified on the cell (`CELL_DESTROY`), the cell timestamp, the circuit ID and the source IP address. At this point, the cell is designed to reach the rendezvous point, which may report the central server the timestamp of the cell before forwarding it to the client. Finally, from the central server, through time correlation may find the IP address of the hidden service.

Caronte attack Caronte [32] is a tool to automatically identify location leaks in hidden services. Such information includes, i.e., sensitive data in the content served by the hidden service or the configuration of the server, potentially able to disclose the IP address of the hidden service. These location leaks are usually introduced by the administrator of the hidden service and, in virtue of this, they do not refer to some sort of vulnerability of Tor.

Off-path MitM This attack is based on the execution of a man-in-the-middle (MitM) attack against a Tor hidden service [33]. In particular, by assuming the private key adopted by the hidden service to communicate on the network is owned by the attacker, it is possible to accompish a MitM attack. The important aspect in this case is that it is not required the attacker to be located in the communication path between the client and the server.

3.3 Attacks to the network

In this case, the target of the attack is the Tor network itself. By targeting the entire network, it is important to consider that in this case, multiple nodes may be affected by the malicious activities. Hence, in this case, the attack effects may be propagated to the entire network, instead, for instance, to affect a single node.

Bridge discovery In this case, the aim is to retrieve information on Tor bridge nodes. Such information are not publicly available [15]. Two different bridge discovery approaches are considered [34]: from one side, it is possible to enumerate Tor bridges through bulk emails and HTTPS servers over Tor. From the other side, it is possible to adopt a malicious Tor middle router/node to exploit the weighted bandwidth routing algorithms of Tor for bridge discovery purposes.

Denial of service Denial of service (DoS) attacks are executed to make a network component or service not available on the network, or to reduce its availability. A DoS attack against the Tor network is CellFlood [17]. This attack exploits the fact that adopting a private key to perform 1024-bit operations is, on modern servers, about 20 time slower than performing the same operations with the public key. Therefore, in order to process a Tor cell is 4 times longer/heavier, compared to create it. This approach may lead a malicious client to flood a targeted node with specifically created cells, in order to seize all the computing resources of the target, hence leading to a denial of service.

Sniper The Sniper attack [32] exploits the flow control algorithm of Tor, by executing a DoS attack against a target Tor relay, killing the Tor process on the machine. This is reached by forcing a node to buffer large amounts of data (utilizing valid protocol messages) until it is overloaded and forced offline. The adversary can attacks a huge number of nodes to degrade network capabilities and increase the chance for a client to choose an attacker's node. In the paper two attacks are described: (i) the attacker stops reading from the TCP connection containing the attack circuit, which causes the TCP window on the victim's outgoing connection to close and the victim to buffer up to 1000 cells; (ii) the attacker causes cells to be continuously sent to the victim (exceeding the 1000 cell limit and consuming the victim's memory resources), either by ignoring the package window at packaging end of the circuit, or by continuously sending `SENDME` messages[3] from the delivery end to the packaging, end even though no cells have been read by the delivery end.

3.4 Generic attacks

Since attacks may not target a single Tor entity (client, server, network), in the following we report a set of attacks designed to target multiple entities.

Traffic analysis attacks This kind of attacks is based on network traffic analysis [4]. For this type of attack, packets are inserted server-side, trying to observe these packets from client-side through a statistical correlation. The goal is to derive the circuit established by the client and associate the client with the observed packets from the exit node. It is assumed that the attacker is able to observe the traffic that enters and leaves the Tor network through the nodes, at various points. The proposed attack, tries to force the client to make a connection to a malicious server, such that it is able to inject a specific repetitive traffic in the TCP connection. The attacker in possession of a great amount of entry node, will observe traffic between various entry node and client, and then will try to detect that specific traffic entered by the malicious server. Once the traffic is recognized, by statistical correlation, it is possible associate traffic with the client, so obtaining Tor circuit used. In general, it has been proved that it is possible to counter traffic analysis methods by employing mixing strategies [35].

Timing attacks This attack represents a variant of traffic analysis attack previously mentioned. Indeed, timing attacks [24, 36] try to obtain a relationship between the client and the server, by observing exchanged packets to accomplish temporal correlation. The attacker must in this case own both the entry and the exit node of the victim's circuit. In this case, it is possible to associate packets to a defined client/server, through a temporal analysis, even though the content of the packet is unknown or encrypted. Traffic may be actively temporarily interrupted at predefined intervals, in order to facilitate correlation. In order

[3] A `SENDME` message specifies the exit node to increase its congestion window, hence to continue to pull data from the external source and forward it into the Tor circuit.

to protect the nodes from this types of attacks, Tor nowadays embed packets buffering, delaying and shuffling approaches.

It is also possible to combine the same approach to accomplish traffic analysis [37]. In this case, by executing timing attacks on the traffic related to the victim and adopting traffic analysis to accomplish bandwidth estimation, the attack is able to infer the network identity of an anonymous client, hidden service, and anonymizing proxies.

Shaping attacks This attack represents a variation of the timing attack previously described. While in case of timing attacks, traffic may be interrupted for specific periods, in this case, the attack actively alters the traffic shape to facilitate correlation. By analyzing and comparing the shape, it is possible to identify variations from the expectations [29] to compare different traffic flows and correlate the traffic with higher confidence.

4 Considerations on available attacks

Mentioned attacks are important threats to the Tor ecosystem and they can be adopted by malicious users in order to retrieve information or to perpetrate malicious activities. According to our categorization, Table 1 reports the targeted entity of each attack.

Threat	**Client**	**Server**	**Network**
Torben	•		
P2P info leakage	•		
Induced Tor guard selection	•		
Raptor	•		
Unpopular ports exploitation	•		
Low-resources routing	•		
Cell counting and padding		•	
Tor cells manipulation		•	
Caronte		•	
Off-path MitM		•	
Bridge discovery			•
Denial of service			•
Sniper			•
Traffic analysis	•	•	
Timing	•	•	
Shaping	•	•	

Table 1. Overview of Attacks against the Tor Network

By exploiting network or protocol vulnerabilities, it is possible to target the Tor network through different approaches and by targeting different entities of the network.

Instead, in order to identify and protect Tor, different research activities focus on the adoption of machine learning approaches to identify running attacks to the Tor network. In particular, neural networks may be adopted to implement an Intrusion Detection System able to identify running threats [38, ?]. In addition, machine learning algorithms and techniques may be employed to identify whether a host is generating Tor related traffic, in order to detect possible malware exploiting the underlying network [39].

5 Conclusions and further work

In this paper, we have investigated the darknet security topic, related to attacks that are related to a darknet environments. By focusing on the Tor onion network [5], we have deeply investigated the literature of cyber-attacks exploiting such system. In order to provide a more easy to understand overview of the threats against darknet environments, we have proposed an easy-to-understand categorization of attacks against darknet environments, by also categorizing the investigated threats. The proposed categorization should be considered an important step in the darknet security context, since it provides an important tool to classify threats, hence, to better understand them and to propose efficient protection systems.

Further work on the topic may be directed on the execution of the mentioned threats on controlled environments. Also, additional work may be focused on the investigation of detection and mitigation approaches able to counter the analyzed threats, by proposing an appropriate taxonomy of protection systems.

6 Acknowledgement

This work has been supported by the following research projects: (i) Advanced Networked Agents for Security and Trust Assessment in CPS/IoT Architectures (ANASTACIA) project has received funding from the European Unions Horizon 2020 Research and Innovation Pro- gramme under Grant agreement no. 731558. (ii) My Health-My Data (MHMD) project has received funding from the European Unions Horizon 2020 Research and Innovation Programme under Grant agreement no. 732907. (iii) Integrated Framework for Predictive and Collaborative Security of Financial Infrastructures (FINSEC) project has received funding from the European Unions Horizon 2020 Research and Innovation Programme under Grant agreement no. 786727.

References

1. A. Zeng and W. Liu, “Enhancing network robustness against malicious attacks,” *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 2012.
2. D. Goldschlag, M. Reed, and P. Syverson, “Onion routing for anonymous and private internet connections,” *Network*, 1999.

3. M. A. Sulaiman and S. Zhioua, “Attacking tor through unpopular ports,” in *Proceedings - International Conference on Distributed Computing Systems*, 2013.
4. S. J. Murdoch and G. Danezis, “Low-cost traffic analysis of Tor,” in *Proceedings - IEEE Symposium on Security and Privacy*, 2005.
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 - WPES '07*, 2007.
6. I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley, “Protecting free expression online with freenet,” *IEEE Internet Computing*, 2002.
7. B. Zantout and R. Haraty, “I2P data communication system,” in *The Tenth International Conference on Networks*, 2011.
8. G. N. Tchabe and Y. Xu, “Anonymous Communications: A survey on I2P,” tech. rep., 2014.
9. M. Rennhard and B. Plattner, “Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection,” in *Proceedings of the Workshop on Privacy in the Electronic Society WPES 2002*, 2002.
10. C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig, “HORNET,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*, 2015.
11. M. J. Freedman and R. Morris, “Tarzan : A Peer-to-Peer Anonymizing Network Layer,” *Proceedings of the 9th ACM Conference on Computer and Communications Security - CCS '02*, 2002.
12. S. Nepal, S. Dahal, and S. Shin, “Deanonymizing schemes of hidden services in tor network: A survey,” in *International Conference on Information Networking*, 2015.
13. B. He, M. Patel, Z. Zhang, and K. C.-C. Chang, “Accessing the deep web,” *Communications of the ACM*, 2007.
14. I. Sanchez-Rola, D. Balzarotti, and I. Santos, “The Onions Have Eyes: A Comprehensive Structure and Privacy Analysis of Tor Hidden Services,” in *Proceedings of the 26th International Conference on World Wide Web - WWW '17*, 2017.
15. S. Matic, C. Troncoso, and J. Caballero, “Dissecting Tor Bridges: a Security Evaluation of Their Private and Public Infrastructures,” in *Network and Distributed System Security Symposium (NDSS)*, 2017.
16. P. Manils, C. Abdelberri, S. L. Blond, M. A. Kaafar, C. Castelluccia, A. Legout, and W. Dabbous, “Compromising Tor anonymity exploiting P2P information leakage,” *arXiv preprint arXiv:1004.1461*, 2010.
17. M. V. Barbera, V. P. Kemerlis, V. Pappas, and A. D. Keromytis, “CellFlood: Attacking tor onion routers on the cheap,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013.
18. S. Mauw, J. Verschuren, and E. de Vink, “A formalization of anonymity and onion routing,” *Proceedings of ESORICS 2004*, 2004.
19. M. G. Reed, P. F. Syverson, and D. M. Goldschlag, “Anonymous connections and onion routing,” *IEEE Journal on Selected Areas in Communications*, 1998.
20. K. Bauerd, M. Sherr, D. McCoy, and D. Grunwald, “ExperimenTor: A Testbed for Safe and Realistic Tor Experimentation,” in *Cyber Security Experimentation and Test*, 2011.
21. S. L. Blond, P. Manils, A. Chaabane, M. A. Kaafar, A. Legout, C. Castellucia, and W. Dabbous, “De-anonymizing BitTorrent Users on Tor,” *World*, 2010.

22. G. Danezis and C. Troncoso, “You cannot hide for long: de-anonymization of real-world dynamic behaviour,” *Proceedings of the 12th ACM Workshop on privacy in the electronic society*, 2013.
23. N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, “Cookieless monster: Exploring the ecosystem of web-based device fingerprinting,” in *Proceedings - IEEE Symposium on Security and Privacy*, 2013.
24. T. G. Abbott, K. J. Lai, M. R. Lieberman, and E. C. Price, “Browser-based attacks on Tor,” in *International Workshop on Privacy Enhancing Technologies*, pp. 184–199, Springer, 2007.
25. D. Arp, F. Yamaguchi, and K. Rieck, “Torben: A Practical Side-Channel Attack for Deanonymizing Tor Communication,” in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security - ASIA CCS '15*, 2015.
26. R. L. Xia and J. K. Muppala, “A survey of BitTorrent performance,” *IEEE Communications Surveys and Tutorials*, 2010.
27. Q. Li, P. Liu, and Z. Qin, “A stealthy attack against tor guard selection,” *International Journal of Security and its Applications*, 2015.
28. L. Vanbever, O. Li, J. Rexford, and P. Mittal, “Anonymity on QuickSand : Using BGP to Compromise Tor,” in *HotNets*, 2014.
29. M. Aiello, E. Cambiaso, S. Scaglione, and G. Papaleo, “A similarity based approach for application DoS attacks detection,” in *Proceedings - International Symposium on Computers and Communications*, 2013.
30. Z. Ling, J. Luo, K. Wu, and X. Fu, “Protocol-level hidden server discovery,” in *Proceedings - IEEE INFOCOM*, 2013.
31. P. Eckersley, “How unique is your web browser?,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.
32. S. Matic, P. Kotzias, and J. Caballero, “CARONTE: Detecting Location Leaks for Deanonymizing Tor Hidden Services,” in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.
33. A. Sanatinia and G. Noubir, “Off-path man-in-the-middle attack on tor hidden services,” *New England Security Day, NESD*, 2017.
34. Z. Ling, J. Luo, W. Yu, M. Yang, and X. Fu, “Tor Bridge Discovery: Extensive Analysis and Large-scale Empirical Evaluation,” *IEEE Transactions on Parallel and Distributed Systems*, 2015.
35. K. S. Kohls and C. Pöpper, “POSTER: Traffic Analysis Attacks in Anonymity Networks,” in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 917–919, ACM, 2017.
36. B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, “Timing attacks in low-latency mix systems,” in *FC*, 2004.
37. S. Chakravarty, M. V. Barbera, G. Portokalidis, M. Polychronakis, and A. D. Keromytis, “On the effectiveness of traffic analysis against anonymity networks using flow records,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.
38. T. Ishitaki, D. Elmazi, Y. Liu, T. Oda, L. Barolli, and K. Uchida, “Application of Neural Networks for Intrusion Detection in Tor Networks,” in *Proceedings - IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2015*, 2015.
39. A. Cuzzocrea, F. Martinelli, F. Mercaldo, and G. Vercelli, “Tor traffic analysis and detection via machine learning techniques,” in *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, 2018.