2010 Fourth International Conference on Network and System Security

Digging into Anonymous Traffic: a deep analysis of the Tor anonymizing network

Abdelberi Chaabane, Pere Manils, Mohamed Ali Kaafar
INRIA Rhône-Alpes
Grenoble, France
{chaabane, manils, kaafar}@inrialpes.fr

***Abstract*—Users' anonymity and privacy are among the major concerns of today's Internet. Anonymizing networks are then poised to become an important service to support anonymous-driven Internet communications and consequently enhance users' privacy protection. Indeed, Tor an example of anonymizing networks based on onion routing concept attracts more and more volunteers, and is now popular among dozens of thousands of Internet users. Surprisingly, very few researches shed light on such an anonymizing network. Beyond providing global statistics on the typical usage of Tor in the wild, we show that Tor is actually being mis-used, as most of the observed traffic belongs to P2P applications. In particular, we quantify the BitTorrent traffic and show that the load of the latter on the Tor network is underestimated because of encrypted BitTorrent traffic (that can go unnoticed). Furthermore, this paper provides a deep analysis of both the HTTP and BitTorrent protocols giving a complete overview of their usage. We do not only report such usage in terms of traffic size and number of connections but also depict how users behave on top of Tor. We also show that Tor usage is now diverted from the onion routing concept and that Tor exit nodes are frequently used as 1-hop SOCKS proxies, through a so-called tunneling technique. We provide an efficient method allowing an exit node to detect such an abnormal usage. Finally, we report our experience in effectively crawling bridge nodes, supposedly revealed sparingly in Tor.**

I. Introduction

Anonymizing networks such as Tor [1] and I2P [2] find increasing interest by users that are aware about their anonymity and/or privacy. Historically, the main goal of these networks was to avoid "political" censorship from a few countries and to allow freedom of speech on the Internet. However, many Internet access restrictions policies deployed either by law enforcement or due to ISP self traffic regulations, seem to generalize such a seek for Internet anonymous communications. This push more people throughout the world to support Tor efforts by setting onion routers and exit nodes. Surprisingly, only few works (i.e. [4]) have explored for what Tor is being actually used and misused, and how the Tor network looks like in the wild. This might be due to technical barriers to comply with ethical and legal aspects of logging clear traffic, but also to a common belief in the research community that anonymizing networks are used for the sake of freedom of speech and that it should be unexplored so as to not reveal sensitive information. We believe that understanding the artifacts of such anonymizing network is a mandatory step to not only insure the users' security but to reveal some intrusive usage that would prevent the network and its users from operating normally.

In this paper, we provide a deep analysis of the Tor network in the wild, by setting several exit nodes and distributing them worldwide (Section III). Taking special cautionary measures to comply with the legal and ethical aspects of users' privacy, we performed an analysis of the application usage of the Tor network through a deep packet inspection (as opposite to a simple port-based classification), and show that most of the traffic exchanged through Tor is an undesirable BitTorrent traffic (Section IV). We also observed an important fraction of "unknown" traffic. We present the technique we used to reveal that the vast majority of this traffic is actually an encrypted BitTorrent traffic. Our analysis shows then that the BitTorrent traffic on top of Tor accounts for much more traffic size that what it is commonly believed. We also studied the HTTP and BitTorrent usage over Tor and compared Tor users behaviors to typical Internet users (Section V and VI). In addition, we study the Tor network architecture as it is being actually used, and show that many Tor users do not comply with the protocol, and rather prefer creating tunnels making Tor acting as a simple (1-hop) SOCKS proxy (Section VII). We also show that it is easy to circumvent the bridges collection limits (Section VIII-B).

II. Background

In the following, we provide a brief overview of the Tor anonymizing network. We also summarize the BitTorrent protocol as it is being studied in this paper as one of the major protocols on top of Tor.

A. Tor Overview

Tor is a circuit-based low-latency anonymous communication service [5]. Its main design goals, as stated in the original paper, are to prevent attackers from linking communication partners, or from linking multiple communications to or from a single user. Tor relies on a distributed overlay network and onion routing to anonymize TCP-based applications like web browsing, secure shell, or peer-to-peer communications.

When a client wants to communicate with a server via Tor, he selects n nodes of the Tor system (where n is typically 3) and builds a *circuit* using those selected nodes. Messages are then encrypted n times using the following *onion encryption* scheme: messages are first encrypted with the key shared with the last node (called the *exit node* of the circuit) and

978-0-7695-4159-4/10 $26.00 © 2010 IEEE
DOI 10.1109/NSS.2010.47

subsequently with the shared keys of the intermediate nodes from $node_{n-1}$ to $node_1$. As a result of this onion routing, each intermediate node only knows its predecessor and successor, but no other nodes of the circuit. In addition, the onion encryption ensures that only the last node is able to recover the original message.

A Tor client typically uses multiple simultaneous circuits. As a result, all the streams of a user are multiplexed over these circuits. For example, a BitTorrent user can use one of the circuits for his connections to the tracker and other circuits for his connections to the peers.

Finally, some ISP may block access to Tor network by filtering the IP addresses of Tor nodes. To circumvent this censorship, the Tor project has created the so-called *bridges*. These are new types of Tor routers that are not listed in the main Tor directory, and hence cannot be blocked. Tor restricts access to this list and gives a small subset (3 bridges IP addresses) per unique requester IP for a fixed period of time.

B. *BitTorrent Overview*

A torrent is a set of peers sharing the same content. In this section, we briefly describe the protocol flow when Alice joins a torrent (Figure 1).

To join a torrent, Alice sends an *announce* message to the tracker that maintains the list of all peers in that torrent (step 1 in Figure 1). The announce is an HTTP GET message containing the identifier of the requested torrent. Such identifier is known as the *infohash* of the torrent and is unique.

Once the tracker receives the announce message for a specific torrent identified by the infohash, it selects a random subset of peers in that torrent and returns the endpoints (the IP and port of a peer) of those peers (step 2). Then, Alice establishes a TCP connection and sends a handshake message to each peer (steps 3 & 4).

Finally, popular BitTorrent clients, e.g., μTorrent and Vuze, allow to configure SOCKS proxies and give the option to use the proxy for connections to the tracker, to the peers, or both. Therefore, a BitTorrent client can use Tor, configuring the Tor interface as a SOCKS proxy, for communication to the tracker or the peers independently. Alice can then decide to connect to the tracker via Tor, but to have a direct connection to peers in order not to have performance penalty.

III. Data collection methodology

We instrumented and monitored 6 Tor exit nodes with the default exit policy and 100KB of announced bandwidth. We monitored the traffic for a total period of 23 days on controlled servers that were distributed world wide: two in U.S., two in Europe (France, Germany), and two in Asia (Japan, Taiwan). Each server provides around 20GB of data each day. Almost half of the traffic corresponds to the encrypted Tor traffic, exchanged between the Tor onion routers. To avoid results that might be time correlated, we performed our analysis on two different periods of time. The first dataset (*DataSet1*) was obtained by monitoring our exit nodes from the 3^{rd} of December 2009 for a period of one week, and accounts for 600 GB of data. The second dataset (*DataSet2*) reports data collected from the 18^{th} of January to the 3^{th} of February 2010, and consists in 1.6 TB of data.

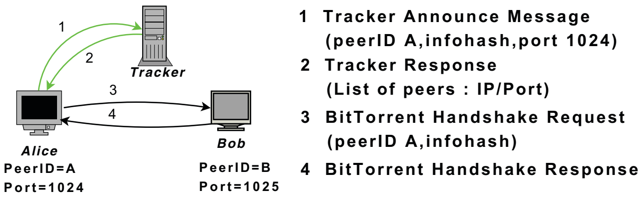
1
2
Tracker
3
4
Alice
PeerID=A
Port=1024
Bob
PeerID=B
Port=1025
1 Tracker Announce Message
(peerID A,infohash,port 1024)
2 Tracker Response
(List of peers : IP/Port)
3 BitTorrent Handshake Request
(peerID A,infohash)
4 BitTorrent Handshake Response

Fig. 1. BitTorrent Protocol Diagram

It is worth noticing that these datasets' size represents the amount of analyzed and not stored data, as discussed later. The results of the two periods show very similar properties, which demonstrate that there is no time correlation. We distinguish between two logging policies:

a) Exit traffic logging: In order to comply with the legal and ethical aspects of privacy, we performed our analysis on the fly. Because in many of our experiments we handle sensitive user data, special cautionary measures were taken in order to present only aggregated statistics as suggested by Loesing et al. in [3]. From the exit traffic logging perspective, only aggregated data was stored, and in particular, we do not keep track of those IP addresses involved into the Tor protocol once we extract useful statistics (locations, associated circuits and applications, etc.).

b) Entrance logging: We have set up a Tor entry point to depict the geographical usage of Tor, recording each IP address establishing a connection to the Tor node. To distinguish between final users and other Tor-special entities (onion routers (OR) and bridges) we have crawled the Tor network.

IV. Application usage

In this section, we concentrate on characterizing what applications are typically used on top of Tor, and to what extent this may impact the Tor network. A previous analysis from McCoy et al. [4] already identified different applications by analyzing the traffic that goes through a controlled exit node. In this section, besides considering the Tor usage from a wider perspective, we will focus on the differences that may have happened after this first analysis was performed. Tor has gained in popularity through the years, and its related traffic has certainly evolved. This is confirmed by our findings below. Moreover, we tackle the problem of application identification through deep packet inspection, and not through a simple port-based classification. This provides more accurate classification of the traffic that is exchanged through the Tor network.

A. *Deep Packet Insepection*

Deep Packet Inspection (DPI) is mainly used for the purposes of traffic shaping based on application detection or intrusion detection. It consists in digging inside packets, using both header and content (payload) to collect useful information, so as to recognize the application that corresponds to the inspected packet.

TABLE I
APPLICATION USAGE (DATASET 1)

Protocol	Packets (Millions)	Size	Flows (Thousand)
HTTP	185.7 (34.31%)	136 GB (36.44%)	4735 (68.57%)
BitTorrent (clear)	136.8 (25.27%)	93 GB (24.92%)	320.5 (4.64%)
SSL	28.5 (5.26%)	20 GB (5.37%)	126 (1.83%)
Others P2P/ file sharing	5.7 (1.07%)	4.4 GB (1.17%)	15 (0.22%)
Insecure (ftp, telnet, email, etc.)	1.3 (0.26%)	1.2 GB (0.32%)	6 (0.09%)
Instant Messaging	6.5 (1.22%)	972 MB (0.26%)	119 (1.72%)
Well-known (other recognized protocols)	18.2 (3.37%)	22.6 GB (6.04%)	1173 (16.99%)
"Unknown"	158 (29.21%)	95 GB (25.47%)	410 (5.94%)
Total	541.5	373.6 GB	6905

An important challenge that faces the exploitation of DPI for application classification is the ethical and legal aspects when knowing both the IP address and related payload, which induces at least privacy compromising issues. Because of these, many researches are reluctant to use DPI. However, the accuracy of simple TCP/IP header inspection techniques, and in particular port-based classification of applications, is hard to infer especially with new techniques employed by P2P applications to avoid ISP traffic throttling. We argue that when considered carefully, DPI is then among the most accurate and useful techniques to characterize the traffic. In order to comply with ethical and legal aspects while monitoring and extracting payload of the packets we captured on top of our Tor exit nodes, we obfuscated all the IP addresses. In particular, each information we retrieve is extracted from packets from which we removed their original destination IP address. Prior to this, we have extracted useful information from the IP address on the fly, before logging the packet as a pcap file. This simple anonymization of the captured packets, along with the anonymization offered by the onion routing concept (Tor) allows to not compromise users privacy while preserving useful information. To characterize protocol usage on top of Tor, we used a self-modified version of OpenDPI [10] (an open source deep inspection packets tool). The analysis of the DataSet1 is reported in Table I.

B. Discarded Flows and Preliminary Results

In our analysis, we take into considerations flows that have at least succeeded one data packet transfer. i.e. we have discarded all flows that generated less than four packets (the three TCP handshake packets and one data packet). In fact, we have observed that a huge amount of connections fail to reach their destination (a single SYN packet is transmitted) or just timeouts after the TCP handshake succeeded. These unsuccessful connections attempts represent 40% of the connections established through our exit nodes. We believe this a BitTorrent symptomatic usage, as SYN packets are often generated by BitTorrent clients that try to connect to other peers that are no longer available, and timeouts are typically the result of busy peers that stopped managing connections after accepting them. Nevertheless, we decide to discard such kind of useless traffic, as it does not represent any application, but only aborted (most likely BitTorrent) connections.

Let us now analyze the results as presented in Table I. As expected, the HTTP protocol constitutes a significant proportion of the traffic in terms of connections. In particular, it is consuming on average slightly more than 35% of the allocated bandwidth on the considered exit nodes. On the other hand, BitTorrent (in clear, as opposed to encrypted BitTorrent that we will identify next) represents nearly the same amount of traffic size with less than 5% of all the established connections. This clearly confirms the very important, yet undesirable load BitTorrent is injecting into the Tor network. We will however show through our analysis (Section IV-C), that this load is even more important than what one can conclude from preliminary results as illustrated in Table I. The usage of other P2P applications is very small, and we observe that BitTorrent is overwhelming the network usage both in terms of packets and traffic size. In contrast to what McCoy et al. reported in [4], the usage of BitTorrent seems to have evolved and its utilization on top of Tor is clearly now within the same order of magnitude than the HTTP usage. We also report the very low usage of "insecure" protocols (non encrypted). In [4], protocols such as FTP, telnet and Email represented a total of less than 0.1%, which is confirmed in our measurements. However, we notice the evolution of the utilization of HTTPS and other secured protocols (SSL row) that represent more than 5% of the traffic size, while [4] noticed only 1.55%. Users might have gained experience through the usage of Tor, avoiding insecure protocols on top of Tor (easy to be eavesdropped by a malicious exit node).

C. What Is the Unknown Traffic?

As reported in Table I, a significant part of the traffic is still unclassified. It represents more than 25% of the entire volume whereas it participates with less than 6% of flows. This means that a small number of connections are responsible of a high data transmission. This behavior suggests that such a traffic likely belongs to any of the P2P protocol. To verify this, we analyzed the distribution of destination ports for those unclassified connections. We observed that destination ports were uniformly distributed, which can led us to believe that such a traffic is a BitTorrent traffic. In fact, to avoid port based detection, BitTorrent clients choose a random port at installation time. This results in uniformly distributed ports. Although these proofs suggest BitTorrent to be responsible of this traffic, our DPI engine does not recognize it. This is most likely because this traffic is encrypted and thus unrecognizable. A step further is then to compute the entropy of a sample data. The computed high entropy value confirmed that this

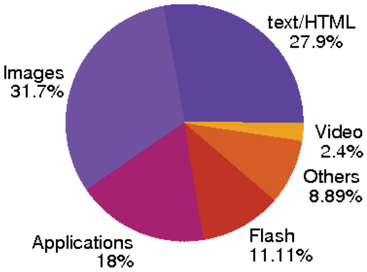

Fig. 2. HTTP content type distribution

data is either encrypted or compressed. Finally, we design an approach that validates our claims (and thus that such a traffic is a BitTorrent traffic), and in addition determines the number of encrypted BitTorrent connections.

Hijacking Trackers' Responses Hijacking, taking place at the exit node, consists in rewriting the list of peers returned by the tracker to Alice (see Figure 1) so that the first peer in the list corresponds to an instrumented BitTorrent client belonging to us. Receiving the subsequent Alice's connection, we can then determine, during BitTorrent handshake establishment, whether Alice is using encryption or not. Furthermore, if Alice uses Tor only to connect to the tracker, our controlled peer will see the Alice's public IP address. If Alice uses Tor also to connect to peers, as the IPs of Tor exit nodes are public, we can easily determine whether we have compromised Alice's public IP. Hijacking is possible because the communication between peers and trackers is neither encrypted nor authenticated. This is a typical man-in-the-middle attack.

Using this technique we can get two valuable information. First, we compute the ratio of encrypted handshake and thus the amout of encrypted BitTorrent traffic. Second, we calculate the number of clients that use Tor only to connect to tracker, and those who also use Tor for content distribution.

This technique shows that **52.78%** of the BitTorrent handshakes are encrypted and thus not recognized by our DPI engine. This confirms our assumption that the unknown traffic is most likely an encrypted BitTorrent.

Conclusion Our findings suggest that BitTorrent is becoming the first contributor in terms of traffic size inside Tor. In essence, more than half of the traffic carried over Tor is BitTorrent. This harmful traffic is responsible of the network overload and the high increase of the latency. It must be noted however, that such an evolution goes along with observations performed outside the Tor environment. In fact, some DPI and traffic management firms such Ipoque [13] and Cachelogic [14] showed that P2P traffic became the dominant application in today's Internet. In 2008, Ipoque found that P2P in Europe accounted for more than 50% of the traffic and web contributing in only a quarter of the traffic. Even though this can be explained by the download of large files within these P2P protocols, the evolution of the number of BitTorrent connexions we observed in Tor stipulates that BitTorrent is being more and more used. This can be mainly explained by the climate of cold war between P2P users and anti-piracy groups.

V. HTTP USAGE

We now focus on the HTTP protocol, being the prime protocol Tor has been designed for. We aim to provide a deep analysis on how this protocol is used on top of Tor. In the following, we characterize the behavior of Tor users while accessing the web and answer the following question: is the behavior of Tor users different from typical [1] users, according to normal (non anonymous) web surfing models [8]? We also concentrate on the way Tor "high" latency may discourage users from browsing interactive contents. Finally, we classify which kind of contents Tor users may be specifically interested in.

A. Content Type Distribution

The HTTP protocol carries a wide spectrum of data going from simple text to rich media such as images and video. Furthermore, a large variety of applications are embedded into browsers to enrich the end user environment. Analyzing this data allows us to have a more comprehensive view of how the web is used on top of Tor. To do so, we do not only extract the `content-type` header in a HTTP response but also use a complementary test based on the LibMagic library [11]. We extract the first 10 bytes of each HTTP response and parse it using the LibMagic library to determine the content type. We believe that considering 10 bytes is a good trade-off between detection effectiveness and privacy. Our findings are shown in Figure 2.

We notice that the most significant content is, as expected, images and text/html. Surprisingly, applications (e.g. rar and zip) content represent a significant proportion of the observed traffic. In addition, we noticed that 6% of the entire traffic is originating from Direct Download Link (DDL). This can be explained by the fact that some users may have switched from P2P networks known to be heavily monitored to DDL-based content, much more harder to control. This behaviour switching have already been noticed in residential broadband Internet where Mainer et al. [8] showed that 16% of the HTTP traffic in that case involves Direct Download providers and that such traffic originates almost 90% of application exchanged bytes. On the other hand, Flash and video usage representing 13.5% of the observed content, shows that the latency induced by the Tor relaying is not an actual brake for browsing Web 2.0. This result shows also that bulk traffic over HTTP is higher than what has been observed previously in [4]. We can explain that by the migration of the web in general from static content mainly composed of texts and images to multimedia-rich contents.

B. Web Categories Distribution

Even though Tor has been originally designed to fight censorship, the actual usage of Tor has never been revealed. In this section, our objective is to infer Tor users' behavior when surfing the Web. First, we extract nearly 4 millions domain names from the HTTP headers, that we classify using the

[1] Referring to users that do not use Tor

TABLE II
MOST VISITED WEB-SITES ACCORDING TO THEIR CATEGORIES

Rank	Category	Percentage
1	Search Engines/Portals	14.45%
2	Pornography	11.50%
3	Computers/Internet	11.45%
4	Social Networking	9.52%
11	Blogs/Web Communications	2.26%
13	StreamingMedia/MP3	1.82%
14	Software Downloads	1.66%
36	Hacking	0.3%
40	Political	0.18%
42	Illegal/Questionable	0.15%
52	IllegalDrugs	0.06%

Trend Micro online URL query service [12]. This classification provides an overview of the main topics of interest of web users on top of Tor.

We report different categories in Table II with their respective rank and percentage of visited web sites that fall into the corresponding category. We observed that more than 65% of all visited web sites are grouped into only 10 categories. A significant part of users is mainly interested in few categories while accessing the web through the Tor network. As expected, search engine access ranks first. As typical users would do, when accessing web pages, Tor users perform a search query to click on the correct URL link, following then "normal" surfing behavior. Pornography ranked second, with more than 10% of all visited websites belonging to this category. Most users consider such content as a must-anonymized traffic, and use Tor to do so when accessing porn web sites. Less expectable is the Social Networking category that ranks 4^{th}. This can be explained by either the usage of online social networks (OSN) to spread and access sensitive political or personal information, and so the use of OSN as a freedom of speech catalyzer. Recent examples demonstrate this. Indeed, Iranians protesters used OSNs to organize their protests actions and events. Political opposition use also OSNs to show evidence of persecutions and to reveal their claims to the world. However, the small number of Tor users in these politically-sensitive countries (as we will show in Section VIII-A) argues in favor of the development of OSNs on top of Tor, because of corporate censorship conducted by enterprises to prevent their employees from accessing such web categories [9]. This may push many users to use Tor as a way to circumvent such filtering policies. Finally, we stress the small proportion of sites categorized as containing illegal contents, showing that Tor is also being used as a way to be anonymous while undertaking illegal actions on the web.

VI. BITTORRENT USAGE

BitTorrent users find in Tor a way to distribute content anonymously, going unnoticed from anti-piracy groups, government and ISPs. With more than 50% of the overall traffic (see Section IV), BitTorrent is the most important exchanged traffic within Tor. In the following, we depict BitTorrent users' behavior by focusing on how they are using BitTorrent and which type of contents they are exchanging.

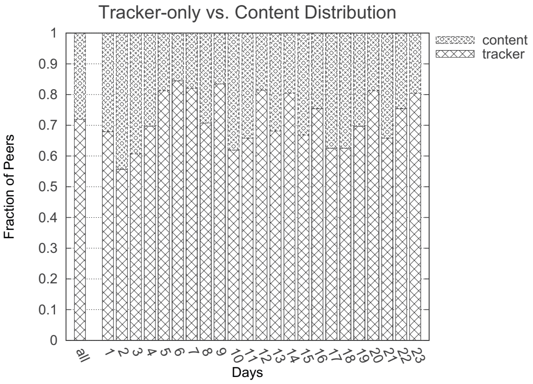

Fig. 3. Proportion of peers who use Tor for content distribution (content) or only to connect to the tracker (tracker). *all* is the average over all days.

A. Content Distribution vs Tracker Access

Tor can be used by a BitTorrent user to (1) hide from the tracker, (2) hide from other peers, i.e., content distribution, or (3) hide from both the tracker and other peers. In this section, we characterize the usage of BitTorrent users on top of Tor.

Usage (1) is the one advocated by the Tor project in its conditions of utilization. As BitTorrent content distribution overloads the Tor network, the Tor project considers usages (2) and (3) as undesirable. However, it is tempting for users willing to trade performances for anonymity to use Tor for content distribution thus violating Tor's conditions of utilization. Quantifying the fraction of users distributing content over Tor is important for two reasons. First, it tells the reason why BitTorrent users are on top of Tor. Second, it says how many BitTorrent user are responsible of overloading the Tor network.

To quantify the fraction of BitTorrent users using Tor for content distribution, we rely on the hijacking technique described in Section IV-C. This technique forces a peer to unwillingly connect to a controlled machine, impersonated by an adversary. As mentioned in Section IV-C, an adversary can easily determine the usage of a hijacked peer. In particular, a peer with usage (1) will connect to the attacker from a public IP whereas a peer with usage (2) or (3) will connect to the attacker from the IP address of an exit node. We remind that the IPs of the exit nodes are public so it is easy to determine whether a peer only hides from the tracker or also from the peers. We rely on the peer IDs (embedded in each BitTorrent communication packet) to count the number of unique peers that connect to us every day.

One limitation of our methodology is that we cannot distinguish between usage (2) and (3). However, we argue that usage (2) should be marginal as it implies that a user goes into the trouble of distributing content over Tor whereas her public IP address is published into the tracker.

We show the distribution of the peers with usage (1) (tracker-only) and usage (3) (content) in Figure 3. Most BitTorrent users (73%) only hide from the tracker and do not distribute content over Tor therefore they respect Tor's conditions of utilization. This trend is relatively constant in time for a period of 23 days. As these peers who only hide

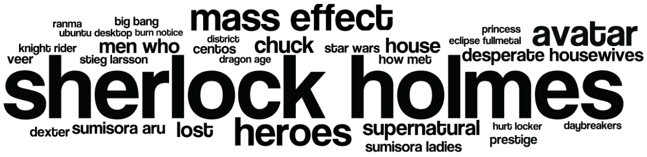

Fig. 4. Word cloud of the 30 most popular torrents downloaded through Tor.

from the tracker just send a few announce messages on Tor, this result implies that only few other peers (27% of BitTorrent users) are responsible of most of the BitTorrent traffic on top of Tor.

B. Downloaded Files

In order to know which files BitTorrent users are downloading through Tor, we collected the `infohashes` present in announce and handshake messages. Recall from section II-B that an infohash represents a unique identifier of a torrent.

We collected a total number of 662.184 infohashes, where 201.779 were embedded in the announce messages and 460.405 appeared in BitTorrent handshakes. This collection was possible because neither the tracker's requests nor the connections between peers carrying those infohashes were encrypted. After removing duplicates, we ended up with 79.865 unique infohashes.

Given the resulting list of infohashes we then tried to *resolve* each infohash to the torrent name by seeking several torrent discovery web sites. Among the 79.865 infohashes, 28.494 (35.7%) were not present. This can be explained by the fact that this content is exchanged between members of private communities. These communities (also known as "BitTorrent darknets") have been already studied by Zhang et al. in [15], showing that the infohash-based intersection of active torrents in private and public sites is extremely small. This shows that users in BitTorrent darknets are aware of their illegal acts (most of the content in darknets is copyright protected [15]) and this issue may push them to use Tor as a way to hide their identity.

For the infohashes we were able to resolve, we kept the torrent name. Considering the frequency of the infohash we depicted the word cloud (Figure 4) of the first 30 most popular torrents. We found films and games that, at time of our experiments, have just been released. As example: *Sherlock Holmes* film was released on 26th of December. But we also found more conventional content like TV shows, included among others *Heroes*, *Dr. House*, and *Desperate Housewives*. All these content is copyright protected, which clearly proves that a huge portion of the BitTorrent users on top of Tor are participating in the distribution of copyright materials.

VII. Misbehaving Clients

We define *misbehaving clients* as users that use the Tor network in a way that does not comply with the onion routing concept. Next, we present our experience while running our 6 Tor exit nodes, revealing our observations of many users bypassing the high latency induced by the three hop-based relay of Tor, by exploiting the Tor exit nodes as 1-hop SOCKS proxies. This still allows them to be anonymous (even if it is a lower level of anonymity) while decreasing significantly delays to access destinations. In Section VII-A, we will then concentrate on this abnormal usage of Tor, and propose in Section VII-B an effective method to identify such a deviant usage of Tor.

A. Tor Tunnels: When Tor Becomes a Simple SOCKS Proxy

Tor follows the original onion routing protocol design [7]. Path to the destinations, as seen by the clients, are composed by 3 nodes: the entry point, the middle node and the exit node. It has been already discussed that onion routing with less than 3 relays (hops) may compromise anonymity [16]. Even though risks to be de-anonymized are higher when using Tor as simple proxy-based network, several users opt for such an option as a simple way to hide their IP addresses in the Internet with small latencies. More importantly, a single-hop based Tor allows for free and highly available SOCKS proxy and a very juicy feature: the traffic between the client and the proxy is encrypted. A SOCKS proxy offering such free desirable properties attracts many users that may concede a strong level of anonymity in favor of lower latency or just will to bypass corporate firewalls and content filtering systems. A direct connection is then established between the clients and a Tor exit node, emulating the behavior of a middle node, and creating what is referred to as a *Tor tunnel*.

Even though the Tor project does not support the use of exit nodes as single-hop proxies, there exist several tools that allow users to establish Tor tunnels through a Tor exit node. In the following, we present a method to identify such behavior and quantify how many users used our Tor exit nodes as 1-hop proxy during our experimentations.

B. Methodology

The following method allows a Tor exit node to detect, with a high provability, connections that are exploiting the exit node as a Tor tunnel.

Once a Tor client builds a circuit, it sends specific Tor control messages (called `RELAY_BEGIN` cells) to instruct the last hop in the circuit (the exit node) to establish a TCP connection to the destination $host/port$ specified in the cell. Typically, the client *randomly* chooses 3 Tor nodes (also called *onion routers*) to build a 3-hop circuit. Hence, when the client sends the `RELAY_BEGIN` cells, the chosen exit node receives the cells from a connection whose source's IP address belongs to the middle node. In the Tor tunneling case, the client builds a 1-hop circuit, thus establishing a direct connection to an exit node, and it starts sending the `RELAY_BEGIN` cells. In other words, the problem of identifying Tor tunnels can be summarized in identifying connections carrying `RELAY_BEGIN` cells that do not originate from a Tor onion router.

Recall that the list of onion routers is public and the `RELAY_BEGIN` cells are sent by Tor clients through the chosen circuit [2] and can only be decrypted by the corresponing

[2] In Tor, control messages are also transmitted in an onion routing-like communication

TABLE III
TOR CONNECTIONS CONTAINING RELAY_BEGIN CELLS.

OR connections	Unique IP addresses	Once OR	Once non OR	Always OR	**Always non OR**
299977	6393	6234	504	5889	**159**

exit node in the circuit.

Although other Tor stakeholders whose IP addresses are not public (e.g. bridges, hidden services) can establish circuits/connections to a public node, say n, when n is not playing the role of an exit node, it cannot recognize the RELAY_BEGIN cells or even be the destination of the connections (i.e.: a bridge *cannot* establish a direct connection to an exit node).

As a consequence, when n acts as an exit node, if it receives a RELAY_BEGIN cell inside a connection having as source IP address one of the public onion router IPs, n can conclude that the cell is *most likely* generated by a normal Tor client (a user building a 3-hop circuits). On the other hand, when the exit node receives the RELAY_BEGIN cells from a host that does not appear in the public onion routers' list, then the host is *most likely* using the Tor node n as a 1-hop proxy. One can observe that this method may lead to some false negatives when the host is a public onion router and it is also establishing 1-hop circuits from the same node.

C. Detection Results

Table III shows the detection results we obtained when monitoring the 6 controlled exit nodes. A total of almost 300 K Tor connections have been received, originated from 6393 unique hosts. In order to validate whether the host establishing the connection is an onion router or not, we used one of the Tor directories archives [3] that contains snapshots of the Tor network state, including the IP address of the onion routers participating in the network.

For each incoming TCP connection established at time t, we checked against the Tor archive if the source IP address was an onion router at that time t. Because a single host may establish many connections to our exit nodes and the Tor archive may lack the data for some times t, we found that the same IP address may sometimes appear in the Tor archive and disappears some other times for different connections in the 23-days period.

In the columns Once OR (resp. Once non OR), we show the number of IP addresses that appear (resp. do not appear) in the Tor archive at least once. Hosts not being seen in the Tor archive (504) are potentially 1-hop users but this value might include a few false negatives caused by the archive's incompleteness. The last two columns show the number of hosts that always (resp. never) appeared in the Tor archive (Always OR and resp. Always non OR). For those hosts that always appear in the archive (5889), we can conclude with high confidence that they were playing the role of middle onion routers connecting to our exit nodes in 3-hop circuits (despite some scarce false negatives of users running a Tor node and maybe using Tor as 1-hop proxy at the same time). On the other hand, the IP addresses that never appeared as being onion routers (159) and were establishing connections to our exit nodes and sending RELAY_BEGIN cells, can be considered with very high confidence as having abused Tor to use it as a 1-hop proxy.

TABLE IV
GEOPOLITICAL DISTRIBUTION OF TOR CLIENT (TOP 7)

Countries	Percentage	Cumulative
Germany	14.7%	14.7%
United States	12.8%	27.5%
Poland	11.08%	38.58%
Romania	7.7%	46.28%
Russian Federation	7.3%	53.58%
China	5.8%	59.38%
France	4.3%	63.68%
Others	36.32%	100 %

VIII. GEOPOLITICAL VIEW

In the following sections we analyse the geographic distribution of Tor clients and bridges.

A. Tor Clients Distribution

Recall from Section III that we have also set up and monitored the traffic at the level of a Tor entry point. We aim in this section to draw an updated view of Tor clients. For a period of one day, we logged the traffic that transited through our entry point and hence collected 7575 unique clients originating from more than 100 countries.

We observed that more than 70% of the clients were originating from only 10 countries. Germany and U.S represent more than the quarter of the clients. Such a high ratio may be explained by Internet demographics aspects (especially the high Internet penetration in these countries) from one hand, but also by the increase and strengthening of anti-piracy and copyright laws during the past few years. The concentration of Tor clients among this small subset of countries and in particular, the absence of politically-sensitive countries among the top countries of the observed clients coupled with the announcements of the Tor project that bridges are still in their infancy and not yet often used by clients [6] may be a good indicator of the common usage of Tor. This observation is confirmed in Table IV, where we observe the Top 7 of the country distribution of Tor clients. Few eastern Europe nations (Poland, Romania and Russia) represent nearly 20% of the Tor clients and Chinese clients correspond to 5,8% of overall clients. It is worth noticing that these statistics are different from what McCoy et al. reported in [4] two years ago, where China ranked second. The Tor clients distribution seems then to evolve. The introduction of bridges as a way to avoid connecting to entry points may explain the discrepancy between our findings and those presented in [4].

B. Bridges Distribution

Obtaining a complete list of Tor relays is an easy task. One have just to query the Tor directory. Knowing this, some ISP (or Internet agencies controlled by governments) may block access to the Tor network by filtering connections to

[3] E.g., http://archive.torproject.org/tor-directory-authority-archive/

TABLE V
GEOPOLITICAL DISTRIBUTION OF BRIDGES (TOP5)

Countries	Percentage	Cumulative
Germany	31.56%	31.56%
China	16.33%	47.89%
United States	11.08%	58.97%
Italy	6.9%	65.87%
France	6.69%	72.56%
Others	27.44 %	100%

all known Tor relays. This prevents end users from reaching the Tor network. To circumvent this censorship, bridges were introduced by the Tor project. They are a new kind of Tor routers that are not advertised in the main Tor directory and thus cannot be blocked by ISP. To avoid crawling, Tor restricts access to this list and gives a small subset (3 bridges IP addresses) per unique requester IP address for a period of time. To have an overview of Tor bridges and their distribution, one has then to deal with this restriction. However, during our experiments we noticed that the bridges distribution restriction policy the Tor project sets up suffers from at least two flaws: first it is based on the uniqueness of the IP address of the requester, and second the answers are not protected with a captcha-like mechanism to prevent automatic crawling.

The crawler we designed is then simply based on the usage of Tor nodes themselves to collect as many bridges IP addresses as possible. The crawler connects to Tor and sends requests to the servers managing the bridges so as they are misled. Surprisingly the web server managing the bridges identities does not recognize frequent requests from Tor exit nodes, and continues proposing new bridges IP addresses, for each request originating from the Tor network. Since there is no captcha-like mechanism to prevent automatic crawling, we believe this is a serious risk against the hiding of bridges identities, as any adversary can use the Tor network itself to constantly collect bridges IP addresses, so as to block communications towards them. Using our simple technique, we collected 3393 unique IP addresses of Tor bridges. Our experiments last for 7 days and 3 hours, during which we most likely collected multiple IP addresses belonging to unique bridge nodes, representing the churn of these bridge nodes (several IP addresses are assigned from a dynamic pool of addresses, and so each time a bridge is set off and then joins back the Tor network, we potentially collect a new assigned address).

Table V shows the geopolitical distribution of these Tor bridges. The cumulative column shows that 3 countries represent nearly 60% of all the collected bridges. We remind that any user can set up his Tor client to act as a bridge and that Tor project encourages users to do so. The high number of bridges in Germany can be explained by the high number of German Tor users and therefore the potential bridges providers. On the other hand, the number of Chinese bridges is less expected. As reported in the Tor's project blog [18], there are evidences that the Chinese government is blocking the access to public Tor nodes. Since bridges are designed to connect to public onion routers, the bridges we identified in China can be considered as doubtful as they would not be able to reach the Tor network.

IX. CONCLUSIONS

This paper provides a detailed analysis of the anonymized traffic traveling through the Tor network using a deep packet inspection approach. We have demonstrated the importance of BitTorrent traffic over Tor. Using a hijacking technique, we do show that a vast majority of the previously considered "unkown" traffic corresponds to encrypted BitTorrent communications. This implies that BitTorrent is the major protocol used on top of Tor in terms of exchanged traffic, consuming more than half of the bandwidth of our exit nodes. Our results support then the idea that P2P traffic is not disappearing but simply hiding through encrypted channels. We have also analyzed how some users abuse the Tor exit nodes to make them act as 1-hop proxies, through a so-called Tor tunneling technique. We then provide a technique to detect such behavior and quantify such abuse of the deployed Tor exit nodes. Finally, we do show that the bridges distribution process as deployed by the Tor project is vulnerable to a simple crawling technique that exploits the exit nodes themselves to collect as many bridge identities as possible. As a conclusion, we hope that our results will contribute in better understanding of the Tor anonymizing network, so as to enhance several features for better deployment.

ACKNOWLEDGMENT

The authors thank Stevens Le Blond for fruitful discussions about BitTorrent on top of Tor. We also thank Roger Dingeldine, Andrew Lewman and the anonymous reviewers whose comments helped improve this paper.

REFERENCES

[1] *The Tor Project*. http://www.torproject.org
[2] *I2P Anonymous Network*. http://www.i2p2.de
[3] K. Loesing, S. Murdoch and R. Dingledine. *A Case Study on Measuring Statistical Data in the Tor Anonymity Network*. In Proc. of Financial Cryptography and Data Security, Tenerife, Spain, 2010.
[4] D. McCoy, K. Bauer, D. Grunwald, T. Kohno and D. Sicker. *Shining Light in Dark Places: Understanding the Tor Network*. In Proc. of Privacy Enhancing Technologies Symposium (PETS), Leuven, Belgium, 2008.
[5] Roger Dingledine, Nick Mathewson and Paul Syverson. *Tor: The Second-Generation Onion Router*. In Proc. of USENIX Security Symposium, pp 303-320, San Diego, CA, USA, 2004.
[6] Roger Dingledine. *Tor and censorship: lessons learned.* http://media.ccc.de/browse/congress/2009/26c3-3554-de-tor_and_censorship_lessons_learned.html
[7] D. Chaum. *Untraceable electronic mail, return addresses, and digital pseudo-nyms*. Communications of the ACM, 4(2), 1981.
[8] G. Maier, A. Feldmann, V. Paxson, M. Allman. *On dominant characteristics of residential broadband internet traffic*. In Proc. of the ACM SIGCOMM IMC, Chicago, IL, USA, 2009.
[9] *Web Filter for Enterprise*. http://www.stbernard.com
[10] *Open Source Deep Packet Inspection Engine*. http://www.opendpi.org
[11] *Library for classifying files according to magic number tests.* http://sourceforge.net/projects/libmagic/
[12] *Trend Micro Online URL Query - Feedback System* http://reclassify.url.trendmicro.com/
[13] *Bandwidth Management with DPI*. http://www.ipoque.com/
[14] *Cachelogic*. http://www.cachelogic.com
[15] Chao Zhang, Prithula Dhungel, Di Wu, Zhengye Liu and Keith W. Ross. *BitTorrent Darknets*. Infocom, San Diego, CA, USA, 2010.
[16] P. Syverson, Tor mailing list. *The Case for Banning Reduced Hop Count Implementations*. http://archives.seul.org/or/talk/Jan-2010/msg00077.html
[17] P. Manils, A. Chaabane, S. Le Blond, M.A. Kaafar, C. Castelluccia, A. Legout, W. Dabbous. *Compromising Tor Anonymity Exploiting P2P Information Leakage*. http://hal.archives-ouvertes.fr/inria-00471556/en/
[18] Tor project blog. *China blocking Tor: Round Two*. https://blog.torproject.org/blog/china-blocking-tor-round-two