Compromising Tor Anonymity Exploiting P2P Information Leakage

Pere Manils, Abdelberi Chaabane, Stevens Le Blond,
Mohamed Ali Kaafar, Claude Castelluccia, Arnaud Legout, Walid Dabbous

ABSTRACT

Privacy of users in P2P networks goes far beyond their current usage and is a fundamental requirement to the adoption of P2P protocols for legal usage. In a climate of cold war between these users and anti-piracy groups, more and more users are moving to anonymizing networks in an attempt to hide their identity. However, when not designed to protect users information, a P2P protocol would leak information that may compromise the identity of its users. In this paper, we first present three attacks targeting BitTorrent users on top of Tor that reveal their real IP addresses. In a second step, we analyze the Tor usage by BitTorrent users and compare it to its usage outside of Tor. Finally, we depict the risks induced by this de-anonymization and show that users' privacy violation goes beyond BitTorrent traffic and contaminates other protocols such as HTTP.

Keywords

Anonymizing Networks, Privacy, Tor, BitTorrent

1. INTRODUCTION

The Tor network was designed to provide freedom of speech by guaranteeing anonymous communications. Whereas the cryptographic foundations of Tor, based on onion-routing [3, 9, 22, 24], are known to be robust, identity leaks at the application level can be exploited by adversaries to reveal Tor users identity. Indeed, Tor does not cipher data streams end-to-end, but from the source to a Tor exit node. Then, streams from the Tor exit node to the destinations are in plain text (if the application layer does not encrypt the data). Therefore, it is possible to analyze the data stream seeking for identity leaks at the application level. Tor does not consider protocol normalization, that is, the removal of any identity leak at the application level, as one of its design goals. Whereas this assumption is fair, Tor focuses on anonymizing the network layer, it makes the task of users that want to anonymize their communications much harder. As an illustration, the Web communications on Tor are the subject of many documented attacks. For instance, attacks can leverage from misbehaving browsers to third party plugins or web components (JavaScript, Flash, CCS, cookies, etc.) present in the victim's browser to reveal browser's history, location information, and other sensitive data [7, 2, 4, 17].

In order to prevent or at least reduce these attacks, the Tor project recommends the use of web proxy solutions like Privoxy or Polipo [19, 5, 21]. The Tor project is even maintaining a Firefox plugin (Torbutton [20]) that, by disabling potentially vulnerable browser components, aims to countermeasure most of the well-known techniques an adversary can exploit to compromise identity information of Tor users. Thus a big effort has been invested and is heading on improvement and protection of the HTTP protocol on top of Tor, but surprisingly, such an effort is limited to this protocol.

P2P applications and more specifically BitTorrent, an application that is being daily used by millions of users [12], have been so far neglected and excluded from anonymizing studies. The crux of the problem is that BitTorrent easily allows any adversary to retrieve users' IP addresses from the tracker for torrents they are participating to. Indeed, by design BitTorrent exposes the IP address of peers connected to torrents. This implies important anonymity and privacy issues, as it is possible to know who is downloading what. To go around this issue, many BitTorrent users that care about their anonymity use Tor, although the Tor project explicitly not recommend the use of BitTorrent on top of the Tor network, because of the major risk of overloading the network.

BitTorrent is a complex protocol with many potential identity leaks, as user privacy is not among its design goals. However, this serious issue is overlooked by BitTorrent users who believe that they can hide their identify when using Tor.

Today's reality is that BitTorrent is one of the most used protocols on top of Tor (with HTTP/HTTPS) in terms of traffic size and number of connections as reported by [16] and observed during our own experiments. Surprisingly, no studies have been conducted on the way BitTorrent may leak the identity of users when the application is running over an anonymizing network. Although, it might be argued that BitTorrent is mostly used for piracy (distribution of illegal content), we believe that privacy issue is a major impediment for the commercial and legal use of BitTorrent. Moreover, identity leaks at the level of a stream might also contaminate other streams, thus compromising non-BitTorrent traffic.

Our study attempts to answer the following three questions:

- How is Tor being used by BitTorrent clients?
- Does the anonymity and privacy's vulnerability of BitTorrent makes Tor less anonymous, leaking information about other Tor usage?
- To what extent can we use de-anonymization to track users and break their privacy through Tor?

The answers to these questions have implications in numerous Tor security applications. In essence, we show in this paper that there is a gap between users' willingness to use BitTorrent anonymously and their expectation to hide their Internet activities through Tor.

As a first contribution (Section 3), we show using three techniques, how a malicious exit node may de-anoymize BitTorrent traffic. Two of our proposed techniques are completely passive, relying on information leakage of the application itself (in our particular case, it is the BitTorrent client leaking information). The third technique is active and exploits the lack of authentication in the BitTorrent protocol.

As a second contribution (Section 3.5), we demonstrate using a so-called *domino effect* that the identity leak contaminates all streams from the same Tor circuit, and, even from other Tor circuits. In particular, we show that BitTorrent users' privacy may be infringed, and more importantly, that these privacy issues may go far beyond P2P traffic.

Exploiting our attacks, we provide as a third contribution (Section 4) the first in-depth study of BitTorrent's usages on top of Tor. In particular, we quantify how BitTorrent users interact with Tor, and detail their behaviors compared to regular BitTorrent users.

Finally, we show how our attacks can be used to perform profiling of BitTorrent users (Section 5). Focusing on the HTTP protocol (being the most used and most protected by Tor) we show that a significant quantity of information is leaking, proving that we can quickly move from a P2P anonymity weakness on top of Tor to privacy issues.

As a conclusion,with hope that our work will contribute to the ongoing debate on the balance between anonymity and privacy preserving and performance-efficient applications (e.g. [1]), we show that the fixes of the anonymity issues we identified may involve support of different cryptographic operations between BitTorrent entities, particularly when used on top of Tor.

2. BACKGROUND

In the following, we provide a brief overview of the Tor anonymizing network. We also introduce different aspects of the BitTorrent protocol, being largely exploited in our attacks.

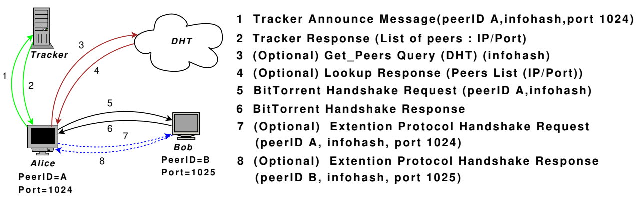

Figure 1: BitTorrent Protocol Diagram

2.1 Tor Overview

Tor is a circuit-based low-latency anonymous communication service [6]. 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 on onion routing [3] to anonymize TCP-based applications like web browsing, secure shell, or P2P communications.

When a client wants to communicate to a server via Tor, he selects n nodes of the Tor system (where n is typically 3) and builds a *circuit* using those 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 *exit node*) on the circuit, and 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.

Onion routing originally built a separate circuit for each TCP stream. However, this required multiple public key operation for every request, and also presented a threat to anonymity from building so many circuits [6]. Tor, instead, multiplexes multiple TCP streams from the same source on a single circuit to improve efficiency and anonymity. In order to avoid delays, circuits are created preemptively in the background. Also to limit linkability, new TCP streams are not multiplexed in circuits containing already older-than-10-minutes streams.

A Tor client typically uses multiple simultaneous circuits. As a result, all the streams of a user are multiplexed over these circuits. Thus, connections to the tracker and connections to the peers can be assigned to different circuits.

2.2 BitTorrent Information Leakage

A torrent is a set of peers sharing the same content. In this section, we briefly describe the information that can leak from a peer Alice when she joins a torrent as in 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 message is an HTTP GET message that contains three important identifiers that we used in our attacks: i) The infohash that is a 160 bits unique identifier of a torrent. ii) The TCP port number selected randomly at the installation of the client on which the peer is listening on. iii) The peer ID of the client, that is the concatenation of an identification of the client version and a random string. This peer ID can be generated at client installation, each time the client is restarted, or each time Alice joins a torrent. iv) Optionally, the IP address of the interface from which Alice sent the message. 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).

Alice can also use a DHT [15, 14] that runs on top of UDP, to find peers in a torrent. In order to retrieve the list of peers, Alice performs a *find_node* query containing the infohash. The result of this query is the ID of the DHT node that maintains the tracker for the queried infohash. Then, Alice performs a *get_peers* query to the DHT node in order to retrieve the endpoints for a random subset of the peers already in the torrent (steps 3 & 4). As with a tracker, Alice can retrieve all the endpoints[1] of a torrent with the DHT. Then, Alice establishes a TCP connection and sends a handshake message to each peer (steps 5 & 6). This handshake message contains the infohash of the torrent, and the peer ID. The port number the peer is listening on is present in the handshake when the extended messages option [18] is enabled in the BitTorrent client (steps 7 & 8). This is the case by default with μTorrent, the most popular BitTorrent client. The extended handshake, *sometimes*, contains the IP address of Alice. We will come back to this issue in Section 3.2.

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.

3. DE-ANONYMIZING BITTORRENT USERS

In the following, we describe the experimental methodology and techniques used to de-anonymize BitTorrent users on top of Tor, and we present the results of their evaluation in the wild.

3.1 Methodology

To de-anonymize the IP address of BitTorrent users in the wild, we instrumented and monitored 6 Tor exit nodes for a period of 23 days. From January 15 to February 7th, we monitored the Tor traffic on controlled exit nodes that were distributed around the globe: two in Asia, two in Europe, and two in the U.S. As anyone can volunteer to host a Tor exit node, performing the attacks described in this paper is within any adversary's grasp.

In order to comply with the legal and ethical aspects of privacy, we performed our analysis on the fly. In addition, special cautionary measures were taken in order to present only aggregated statistics as suggested by Loesing et al. in [13].

[1]By performing numerous queries to the DHT.

3.2 Simple Inspection of BitTorrent control messages

In this section, we show that an attacker can de-anonymize the IP of a BitTorrent user simply by looking at the IP field contained in the BitTorrent control messages introduced in Section 2.2, i.e., announce and extended handshake.

Tracker Announces. As we have mentioned in Section 2.2, the announce message is sent to the tracker to request a list of peers in a torrent. Depending on the client, that message may contain the IP address of the user.

We captured 200k announce messages on our exit nodes. Among the 35% of those messages that contained a non-empty IP parameter, 4% were invalid IP addresses, 38% contained a private IP address, and the remaining 58% contained a public IP address. We ended up with $3,698$ unique public IP addresses.

Surprisingly, most of the public IP addresses we found were IPv6. We also observed that the same versions of BitTorrent clients were alternating between two behaviors, embedding in some cases public IP addresses and in others private ones. The top 3 BitTorrent clients that were constantly embedding public IP addresses (normalized by their presence in our traces) were μTorrent, BitSpirit, and libTorrent.

Extension Protocol Handshakes. After a regular BitTorrent handshake, a client supporting the Extension Protocol may send an additional handshake as described in Section 2.2. That extended handshake message may also contain the user's IP address.

We captured 45k extended handshakes on our exits nodes. In 84% of the handshakes an IPv4 address was present. Of those messages containing an IP address, 33% contained a public IP address that was not the IP of a Tor exit node. In total, we collected $1,131$ unique public IP addresses.

In 67% of the handshakes containing an IPv4 address, the IP belonged to an exit node. Although we have not tested the behavior of those clients, we suspect that they use a service to determine their IP address as seen from the Internet. As they will contact that service through Tor, the service will report the IP address of an exit node.

Conclusions. The inspection of BitTorrent control messages is the simplest of the three attacks that we identify in this paper. To conduct this attack, an attacker only needs to monitor the announce and extended handshake messages on a Tor exit node. However, we have not checked the authenticity of the public IP address contained in those messages, therefore we do not include them in our statistics.

3.3 Hijacking Trackers' Responses

Unlike the previous attack, hijacking the tracker responses guarantees that the de-anonymized IP belongs to the BitTorrent user. Assume Bob is the attacker in Figure 1. Hijacking consists in rewriting the list of peers returned by the tracker to Alice so that the first endpoint in the list belongs to Bob. If Alice uses Tor only to connect to the tracker, but not to connect to peers, then Bob will see Alice's public IP address. As the IPs of Tor exit nodes are public, Bob can easily determine whether he has 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.

Another advantage of hijacking over a mere inspection of the extended handshakes is that it works even when Alice encrypts her communication with other peers. Indeed, clients such as μTorrent support encrypted communication among peers so that a third party, e.g., ISP, cannot identify that the communication belongs to the BitTorrent protocol. In that case, an eavesdropper will not be able to read the IP field of the extended handshake but Bob will see Alice's public IP address because she will establish a TCP connection to Bob. Also note that Bob can answer to Alice's handshake and let Alice send a piece to him to make sure she is distributing the content.

Hijacking the announce responses on a single exit node for 23 days, we were able to collect $3,054$ unique IP addresses, out of which 814 (27%) belonged to exit nodes and $2,240$ (73%) were public. We remind that the hijacking attack works when Alice uses Tor only for tracker communication,

Conclusions. Hijacking the tracker responses allows an attacker to de-anonymize a user who only connects to the tracker using Tor. In addition to the code to instrument and monitor the exit node, this attack requires approximately 200 lines of code to rewrite the list of endpoints, which makes it relatively easy to launch. As we will see in Section 4, more than 70% of BitTorrent users use Tor only to connect to the tracker, making hijacking quite efficient to de-anonymize users.

3.4 Exploiting the DHT

The exploitation of the DHT allows to de-anonymize a user, even if she uses Tor to connect to other peers. Tor does not support UDP communications that are used by the DHT. As a BitTorrent client will fail to connect to the DHT using its Tor interface, it connects to the DHT using the public network interface and publishes its public IP and listening port into the DHT. Therefore, even though Alice connects to Bob through Tor, Bob can lookup Alice's public IP address in the DHT. We have validated this behavior with μTorrent, the most popular BitTorrent client [25].

Assume Bob wants to de-anonymize Alice's IP address in the Figure 1 and that Alice sends an announce or extended handshake message through an exit node that Bob controls.

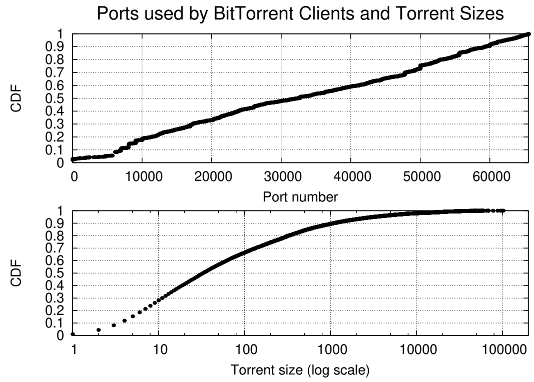

Figure 2: Distribution of the listening ports used by the BitTorrent clients that we have de-anonymized on 6 monitored exit nodes during 23 days (top). Distribution of the torrent sizes in which peers using those exit nodes were participating in during the same period (bottom). *Because the port numbers are uniformly distributed and most torrents are small, the port number is a good peer identifier.*

Bob knows Alice's listening port number and the infohash she's downloading. Bob can then perform a *find_node* request to find the tracker of that infohash and iteratively send *get_peers* messages to it to collect all the endpoints. If one of the endpoint has the same port as Alice's listening port, then Bob has most likely de-anonymized Alice's public IP.

We make the assumption that Alice's listening port number is a good identifier within a torrent. That implies that listening port numbers are uniformly distributed on $[0; 65535]$. As most clients select the listening port randomly at the installation of the client, they should be uniformly distributed. We confirm that assumption in Figure 2 (top). However, we exclude ports 80, 443, 6881, 16884, 35691, and 51413 that are more popular than others. This choice is conservative because we accept to de-anonymize less users to reduce the number of false positives.

For Alice's listening port number to be a good identifier within a torrent, that torrent should also be small in terms of size. We also confirm that assumption in Figure 2 (bottom) where 90% of the torrents have less than $1,000$ peers. By exploiting the DHT, we de-anonymized $6,151$ unique public IPs.

Conclusions. Exploiting the DHT overcomes the weaknesses of the previous attacks. The de-anonymized IP have a very high probability to belong to Alice, and it even compromises the IP of Alice if she uses Tor to connect to other peers. However, as this attack implies to collect all the endpoints for a given torrent, an attacker should develop a dedicated crawler or heavily modify an existing client with DHT support.

In the other hand, not all clients support the DHT. However, the most popular BitTorrent client, μTorrent, supports it by default. In addition, the current trend for large BitTor-

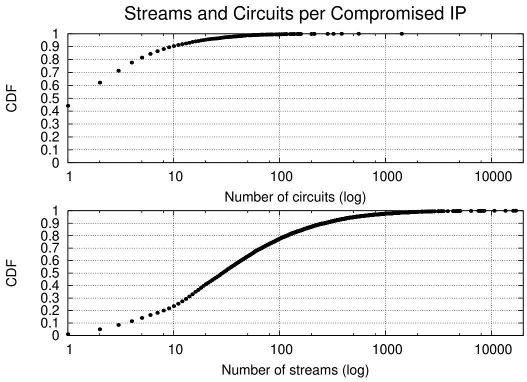

Figure 3: Distribution of de-anonymized circuits (top) and streams (bottom) per de-anonymized IP. *Once we de-anonymize an IP, we are able to de-anonymize multiple circuits and streams using the domino effect.*

rent trackers to promote magnet links [10] instead of torrent files, i.e., DHT instead of trackers, pushes more and more clients to support the DHT.

3.5 The Domino Effect

In the previous sections we have shown different attacks that allow to de-anonymize a BitTorrent stream on top of Tor. Furthermore, as described in Section 2.1, Tor multiplexes different TCP streams of the same source over a single circuit. Consequently, the source of the BitTorrent stream revealed by one of the attacks will also be the source for all the other streams that are multiplexed in the same circuit as the one used by the BitTorrent stream. We call this issue the *intra-circuit domino effect*, as identifying the source of a single stream in a circuit reveals the source for all other streams multiplexed on this circuit.

Moreover, a BitTorrent stream usually contains the peer ID identifying the BitTorrent user. Once a BitTorrent stream is de-anonymized, we associate the peer ID to the compromised IP address. Then by simply comparing the peer ID of other identified BitTorrent streams (belonging to other circuits), we can link the origin of new circuits with previously compromised IP addresses, increasing the set of circuits (and thus streams) linked to the same user.

However, the latter approach is not sufficient when the BitTorrent traffic is encrypted. The following complementary approach would allow to reveal the IP address of the initial source of other encrypted circuits, as long as this source has been identified using one of our techniques while accessing the tracker (recall that the traffic to the tracker is not encrypted). Indeed, first we keep the list of couples (IP,port) returned by the tracker to the compromised peer. If, after a short period of time (say a few minutes) for a new circuit we identify a stream whose destination is one of those (IP,port), we deem then the source of this stream is the one previously identified by one of our techniques. We call the way we link compromised IP addresses to streams belonging to different circuits, the *inter-circuit domino effect*.

Whereas this inter-circuit effect allows to identify the source of a circuit for which our techniques cannot be performed, it might lead to false positives as different peers may share the same list returned by the tracker. The choice of a very short period of time and the low probability that two different peers choose the same exit node to contact the same peer allows then to limit the number of false positives.

Figure 3 shows the CDF of de-anonymized circuits and streams per IP address. We de-anonymized a few circuits for most IPs, i.e., less than 10 circuits for 90% of the IPs, but a significant number of streams. For 75% of the cases, more than 10 streams were de-anonymized. Finally, we observed that the intra-circuit domino effect de-anonymized approximately 80% of the streams, while the remaining 20% were de-anonymized by the inter-circuit domino effect.

4. BitTorrent USAGE ON TOP OF TOR

Characterizing a real deployed anonymizing network is important. In particular, McCoy et al. [16] characterized the usage of Tor two years ago and tried to analyze how Tor is used and mis-used. While these results revealed useful statistics about Tor usage in general, they did not focus on P2P protocols. Additionally, they did not de-anonymize users and hence, they could not link particular usages to locations.

In [16], authors have already shown the importance of the BitTorrent protocol on top of Tor in terms of traffic size and number of connections. BitTorrent ranked the second position among all the identified protocols, representing 40% of all the observed traffic at an exit node.

Our techniques have revealed many IP addresses of BitTorrent users on top of Tor, providing us with a set of *unique* BitTorrent users. More importantly, once de-anonymized, the IPs may be linked to single users based on the connections they established on top of Tor. We first exploited these IP addresses to draw a better view of the individual usage of BitTorrent on top of Tor for each single user. Then, we extended this information to the whole set of IPs. This allows us to compare our results with the regular usage of BitTorrent outside of Tor, using the traces collected in [12].

4.1 Typical Usage of Tor by BitTorrent Users

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

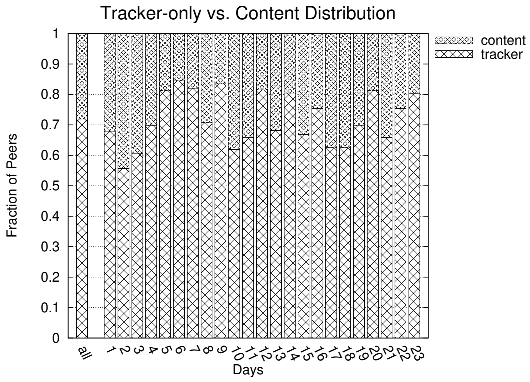

Figure 4: For each day, this histogram shows the proportion of peers who use Tor for content distribution (content) or only to connect to the tracker (tracker). ***all*** **is the average over all days.** *72% of peers use Tor only to connect to the tracker.*

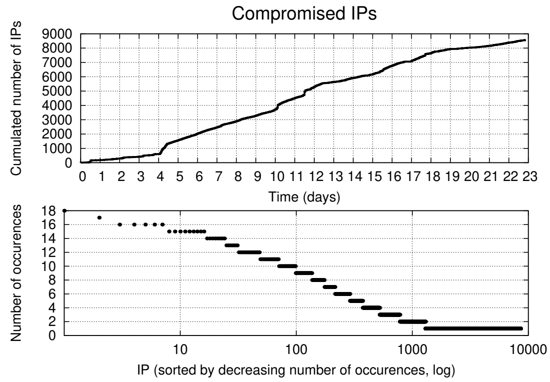

Figure 5: Number of unique IPs that we de-anonimized in time (top) and number of occurrences of a de-anonymized IP with at least one day of interval (bottom). *We compromised IPs uniformly in time. Whereas we de-anonymized most IPs once, a few IPs were de-anonymized many times.*

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 attack. That attack forces a peer to unwillingly connect to an attacker. As mentioned in Section 3.3, an attacker 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 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 4. Most BitTorrent users (72%) 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 the peers who only hide from the tracker just send a few announce messages on Tor every 10 minutes, this result implies that only few peers are responsible of most of the BitTorrent traffic on Tor.

4.2 Returning Users

We show the cumulative number of de-anonymized unique IP addresses in time in Figure 5 (top). Apart from a few bursts on days 4, 10, and 11, we compromise IP addresses uniformly in time with a rate of approximately 372 IPs per day. As we instrument only 6 exit nodes, the cumulated number of de-anonymized IP addresses does not converge yet after 23 days.

However, we de-anonymize many IPs multiple times with at least one day of interval in Figure 5 (bottom). We have chosen at least one day of interval before incrementing the number of occurrences of an IP to have a fair basis of comparison among IPs. As the measurement lasts for 23 days, the maximum number of occurrences is 23. We notice that the IP with the largest number of occurrences appears 18 times, meaning that we de-anonymize that IP almost every day. This behavior suggests that some IP addresses play a peculiar role in BitTorrent, e.g., heavy downloaders, use Tor.

4.3 Usage per Location

We now correlate the location of the de-anonymized IP addresses of BitTorrent users on top of Tor to the location of 10 million IPs of *regular* BitTorrent users. Those 10 million IPs were collected on August 22nd, 2009 and are the merge of 12 global snapshots of ThePirateBay, the largest BitTorrent tracker of the Internet, taken with an interval of two hours. Although the practicality of collecting the IP of most BitTorrent users of the Internet is a serious privacy threat, the complete description of this collect is beyond the scope of this paper. We refer to [12] for more information.

Usage per Country. Figure 6 depicts the distribution of the de-anonymized IP addresses per country for both BitTorrent users on top of Tor and regular users. We observe that BitTorrent users on top of Tor are concentrated in fewer countries than regular BitTorrent users. Indeed, almost 70% of them come from less than 10 countries, whereas in a non-Tor environment, this same percentage is collected from around 20 countries. This is consistent with the fact that Tor is much more popular in few countries as reported by [16]. This is illustrated in Table 1 (left), that shows the popular-

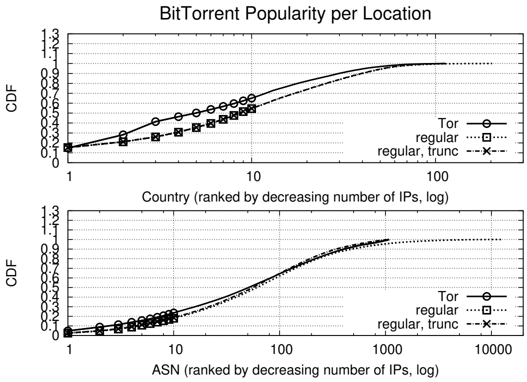

Figure 6: Distribution of IPs per country (top) and AS (bottom) for the BitTorrent users on top of Tor (solid line) and the regular BitTorrent users (hashed line). For both subfigures, we also show the CDF of users per country (resp. AS) in regular BitTorrent with the number of countries (resp. ASes) truncated to be the same as in Tor (dash-dot line).

ity of BitTorrent on top of Tor per country. It is interesting to note the "Over" column where we divide the fraction of users on top of Tor in a given country by the fraction of regular BitTorrent users in that country. This clearly shows that comparatively to its regular usage, BitTorrent on top of Tor is extremely popular in UK, Japan, and India. Besides Tor's popularity in different countries, the usage of BitTorrent on top of Tor might also be impacted by the severity of copyright-laws in countries where BitTorrent is used.

Usage per AS. Figure 6 (bottom) represents the distribution of IP addresses per Autonomous System (AS) for both BitTorrent users on top of Tor, and regular users. We do not show the "Over" fraction for ChinaNet as we have observed that Chinese users do not use ThePirateBay, the tracker that we have used to capture the location of regular BitTorrent users. Again, BitTorrent users on top of Tor are concentrated in fewer ASes than regular BitTorrent users. This is consistent with the concentration into few countries we have noticed previously. The table 1 (right), representing the popularity of BitTorrent on top of Tor per AS, shows that comparatively to its regular usage, BitTorrent on top of Tor is extremely popular in NTT, HanseNet, and Deutsche Telekom.

5. FROM NON ANONYMITY TO PRIVACY ISSUES

Tor enables a user to anonymously browse the Internet, i.e., without revealing his IP address(es) to destinations. As such, Tor provides IP anonymity. It also prevents any entity from linking the source IP address and the destination IP address. It provides what we refer to as IP un-linkability.

In this section, we show that these properties are difficult to fulfill in practice. In particular, we show that BitTorrent users take tremendous privacy risks while using Tor.

Table 1: Popularity of BitTorrent on top of Tor per country (left) and AS (right). The over-representation (Over) for a given country (resp. AS) is the fraction of BitTorrent IPs on top of Tor divided by the fraction of IPs on regular BitTorrent for that country (resp. AS).

Rank	#	%	Over	CC
1	1,255	14	0.9	US
2	1,147	13	5.4	JP
3	1,125	13	2.8	DE
4	426	5	1.2	FR
5	321	3	1.3	PL
6	301	3	0.9	IT
7	264	3	0.7	CA
8	240	2	5.7	IN
9	232	2	0.9	TW
10	229	2	4.6	UK

Rank	#	Over	CC	AS
1	415	4.4	DE	Deutsche Telekom
2	338	5.5	JP	NTT
4	213	1.9	MY	TMNet
3	210	1	IT	Telecom Italia
5	156	0.9	US	AT&T
6	148	1	FR	Orange
7	141	4	DE	Hansenet
8	134	-	CN	ChinaNet
9	132	1	PL	TPNet
10	121	1.4	FR	Free

First, we present a coarse-grained analysis by focusing on de-anonymized users behavior during their web browsing sessions. Then, we show that an attacker can obtain, using the domino effect we described in Section 3.5, private data and link them to the de-anonymized user.

5.1 Coarse-grained Analysis

Figure 7 is an illustration of the type of profiling that can be performed once the source IP addresses of Tor users are retrieved. It displays the ratio of most popular categories of sites accessed by Tor users according to their country of origin. These categories were selected using the classification made by [8].

First, we notice that most of BitTorrent users share a common "typical" behavior (independently from their origins): they are heavy downloaders. In fact, on average 50% of their web usage is categorized as `Peer-to-Peer` or `File Sharing`, with slight variation noticed among different countries. This high frequency in accessing P2P web sites can be explained by the fact that since they are BitTorrent users, they are often browsing torrents' search sites like `torrentz.com` or `mininova.com`. The `File Sharing` category is ranked as second, which shows that

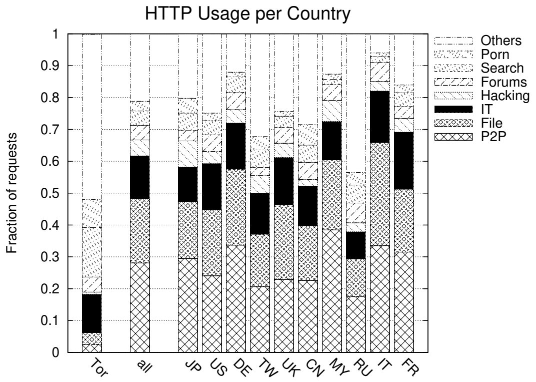

Figure 7: Histogram of the HTTP usage per country. ***Tor*** **represents the overall distribution of requests per category for all Tor users.** ***all*** **represents the overall distribution of requests per category for de-anonymized users. The country codes represent the same information but for each of the top10 countries in number of requests.**

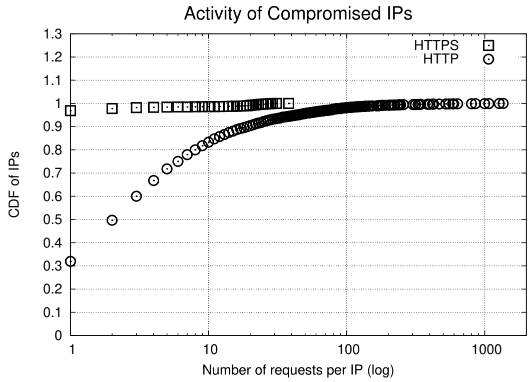

Figure 8: CDF of compromised IP addresses establishing HTTP/HTTPS connections.

most BitTorrent users *also* use HTTP portals for file sharing and multimedia download.

Second, as depicted in the bar labeled `Tor` in Figure 7, typical[2] Tor users' behaviors is significantly different from BitTorrent users' typical web browsing. The latter have little interest in social networks or blog web sites (representing 13% of the web sites common Tor users visited). On the other hand, BitTorrent users seem more interested in forums and hacking sites. In the light of these observations, we can guardedly conclude that most of BitTorrent users on top of Tor show higher IT skills than the average Tor users. Finally, it should be noted that Search, IT and Porn categories seem to be a constant in Tor typical usage.

5.2 Fine-grained Analysis: Users' Profiling

In this section we show evidence of how anonymity issues may lead to privacy risks, while using BitTorrent on top of Tor. Indeed, once de-anonymized, BitTorrent streams can be linked to other applications' streams. Recall from Section 3.5 that by exploiting the domino effect, we can link not only streams belonging to the same circuit to the de-anonymized IP address, but also other streams that may be associated with the same BitTorrent user. This gives an adversary with valuable tools to (i) have a full list of torrent files that a targeted user is downloading, and more importantly to (ii) monitor, among others, the user browsing activities by sniffing the HTTP connections she is establishing through the controlled exit node.

As a consequence, the adversary can extract the user's visited web sites, retrieve searched keywords and even collect user's cookies that transit through the controlled exit node. If anonymized, this information may be not that important for any adversary. However, exploiting our attacks, and thus de-anonymizing the actual user's IP address originating the connections, such information become very sensitive, as several data mining techniques (e.g. [11, 23]) could be used to profile the targeted user.

An example of such serious risks BitTorrent users are taking, when both using BitTorrent and surfing the Internet, is illustrated by the high number of compromised IPs that were establishing HTTP connections in addition to BitTorrent. Figure 8 shows that roughly 70% of users we compromised their IPs have used at least once both protocols on top of Tor at the same time. It also shows that several users are frequently browsing the Internet while using BitTorrent, with more than 100 established HTTP connections we linked to the compromised IP.

The risks these users are taking are even more important when the Internet activity is originated from countries known to have "restrictive" Internet access. As an illustrative example, and for the purpose of proving the sensitive information that can be collected using our de-anonymizing techniques, we have identified and profiled several IPs from China and Myanmar frequently accessing web sites and blogs that belong to political opposition groups. This evidence shows how dangerous our attacks could be, especially if used by third parties to profile users and track "deviant" Internet access, infringing then private users' data.

6. DISCUSSION AND CONCLUSIONS

We have presented three techniques targeting the anonymity of BitTorrent users on top of Tor. In practice, we have demonstrated how an adversary may, with low resources, break users anonymity and shown evidence of serious privacy risks this might induce. We also described, through a so-called domino effect, how identity leak may contaminate different protocols on top of Tor.

In addition, we have quantified and characterized the BitTorrent usage on top of Tor. Exploiting our de-anonymizing attacks, this paper shows the disconnection between users'

[2] Referring to Tor users without any constraint on the protocols they are running.

willingness to use BitTorrent anonymously and their expectation to preserve their identity through Tor. In essence, even if BitTorrent users expect Tor to provide anonymity and IP un-linkability, we show that this is not actually the case. In other words, BitTorrent users are in general not more protected on top of Tor than elsewhere. Our findings may then discourage BitTorrent users from using the Tor network, freeing it from an useless (and undesirable), yet important load.

Even though a solution consisting in end-to-end encryption and authentication in BitTorrent might countermeasure our attacks, we believe this would be a costly solution for trackers to implement, and would induce higher latencies into BitTorrent connections. These non desirable properties such solution exhibits would certainly make heavy downloaders and content providers reluctant to adopt it.

In summary, two factors help our attacks to succeed. First, because BitTorrent is used on top of Tor, it becomes more vulnerable to traffic monitoring and even to communications' hijacking. Second, the lack of cryptographic material's support among BitTorrent entities creates a gap between security and users' expectations when using Tor. We believe this security versus performance balance (also well illustrated by Google persisting in not using HTTPS for a vast majority of its services despite serious risks [1]) should be carefully considered not only by BitTorrent content's providers, but also by users that are willing to anonymously use both BitTorrent and other protocols on top of Tor.

7. REFERENCES

[1] Six pages letter to google's ceo, eric schmidt. `http://www.wired.com/images_blogs/threatlevel/2009/06/google-letter-final2.pdf`, accessed Apr, 2010.

[2] Marco Bonetti. Breaking tor sessions with html5. DeepSec `http://sid77.slackware.it/tor/BreakingTor.pdf`, 2009 (accessed Apr, 2010).

[3] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudo-nyms. In *In Communications of the ACM*, volume 4(2), 1981.

[4] Andrew Christensen. Practical onion hacking: finding the real address of tor clients. FortConsults advisory `http://www.fortconsult.net/images/pdf/Practical_Onion_Hacking.pdf`, 2009 (accessed Apr, 2010).

[5] Chris Davis and Juliusz Chroboczek. Polipo. `http://www.pps.jussieu.fr/~jch/software/polipo/`, accessed Apr, 2010.

[6] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *In USENIX Security Symposium*, 2004.

[7] Gregory Fleischer. Attacking tor at the application layer. DEFCON `http://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-gregory_fleischer-attacking_tor.pdf`, 2009 (accessed Apr, 2010).

[8] FORTINET. Fortiguard web filtering service. `http://www.fortiguard.com/webfiltering/webfiltering.html`.

[9] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding routing information. In *In Information Hiding First International Workshop,*, pages 137–150, 1996.

[10] Greg Hazel and Arvid Norberg. Extension for peers to send metadata files (bep 9). `http://www.bittorrent.org/beps/bep_0009.html`.

[11] Raymond Kosala and Hendrik Blockeel. Web mining research: A survey. *SIGKDD Explorations*, 2:1–15, 2000.

[12] Stevens Le Blond, Arnaud Legout, Fabrice Le Fessant, Walid Dabbous, and Mohamed Ali Kaafar. Spying the World from your Laptop – Identifying and Profiling Content Providers and Big Downloaders in BitTorrent. In *3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'10)*, San Jose, CA United States, 2010. Usenix.

[13] Karsten Loesing, Steven Murdoch, and Roger Dingledine. A case study on measuring statistical data in the tor anonymity network. In *In Financial Cryptography and Data Security '10*, 2010.

[14] Andrew Loewenstern. Bittorrent extension protocol (bep 5). `http://www.bittorrent.org/beps/bep_0005.html`.

[15] Petar Maymounkov and David Mazires. Kademlia: A peer-to-peer information system based on the xor metric. In *In IPTPS*, pages 53–65, 2002.

[16] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the tor network. In *In Privacy Enhancing Technologies Symposium 2008 (PETS)*, 2008.

[17] Metasploit. Decloaking engine. `http://decloak.net`, 2009 (accessed Apr, 2010).

[18] Arvid Norberg, Ludvig Strigeus, and Greg Hazel. Bittorrent extension protocol (bep 10). `http://www.bittorrent.org/beps/bep_0009.html`.

[19] Mike Perry and Scott Squires. privoxy, free software foundation. `http://www.privoxy.org/`, accessed Apr, 2010.

[20] Mike Perry and Scott Squires. Tor button. `https://www.torproject.org/torbutton/`, accessed Apr, 2010.

[21] The Tor project FAQ. Why do we need polipo or privoxy with tor? `https://wiki.torproject.`

org/noreply/TheOnionRouter/TorFAQ#WhydoweneedPolipoorPrivoxywithTor.3FWhichisbetter.3F, accessed Apr, 2010.

[22] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. In *IEEE Journal on Selected Areas in Communications*, volume 16(4), pages 482–494, 1998.

[23] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 675–684, New York, NY, USA, 2004. ACM.

[24] Paul Syverson, Michael Reed, and David Goldschlag. Onion routing access configurations. In *DARPA Information Survivability Conference and Exposition (DISCEX)*, 2000.

[25] Chao Zhang, Student Member, Prithula Dhungel, Student Member, Di Wu, and Keith W. Ross. Unraveling the bittorrent ecosystem. Polytechnic Institute of NYU, 2009.