Computer Networks 59 (2014) 77–90

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

TorrentGuard: Stopping scam and malware distribution in the BitTorrent ecosystem

Rubén Cuevas [a,*], Michal Kryczka [a,b], Roberto González [a], Angel Cuevas [a], Arturo Azcorra [a,b]

[a] *Universidad Carlos III de Madrid, Spain*
[b] *Institute IMDEA Networks, Madrid, Spain*

ARTICLE INFO

Article history:
Received 29 June 2012
Received in revised form 13 August 2013
Accepted 6 December 2013
Available online 14 December 2013

Keywords:
Peer-to-peer networking
BitTorrent
Malware distribution

ABSTRACT

In this paper we conduct a large scale measurement study in order to analyse the fake content publishing phenomenon in the BitTorrent ecosystem. Our results reveal that fake content represents an important portion (35%) of those files shared in BitTorrent and just a few tens of users are responsible for 90% of this content. Furthermore, more than 99% of the analysed fake files are linked to either malware or scam websites. This creates a serious threat for the BitTorrent ecosystem. To address this issue, we present a new tool named TorrentGuard for the early detection of fake content. Based on our evaluation this tool may prevent end users from downloading more than 35 millions fake files per year. This could help to reduce the number of computer infections and scams suffered by BitTorrent users. TorrentGuard is already available and it can be accessed through both a webpage or a Vuze plugin.

1. Introduction

BitTorrent is one of the most popular applications in the current Internet. It is daily utilised by millions of users and is responsible for a major portion of the Internet traffic [29]. This success motivated the research community to investigate different aspects of BitTorrent covering performance [22,28], economics [12,15,34] and incentives [19,30] issues. However, to the best of the author knowledge, the research community has put less attention to BitTorrent security aspects. Some previous works have analysed the vulnerabilities of the BitTorrent protocol to free-riders [24,25,32] whereas some others address the lack of privacy offered by BitTorrent [10]. More recently, in a previous work [14] we unveiled that the BitTorrent ecosystem is suffering from a continuous *poisoning index* attack resulting in 30% of published torrents being associated to fake content. Furthermore, we showed that this fake content produces 25% of the download events. These initial results highlight a serious issue that, to the best of the authors knowledge, has still not been addressed by the research community.

In this paper we thoroughly analyse the *fake publishing* phenomenon in BitTorrent in order to understand its real impact on the system performance as well as the potential risks of fake content for BitTorrent users. Furthermore, we propose a practical solution to mitigate this problem. We base our study on data collected from torrents published in The Pirate Bay portal during a period of 14 days from 30-04-2011 to 13-05-2011. 35% of almost 30 K analysed torrents are associated to fake content. This depicts a 5% increment in the presence of fake content within the BitTorrent ecosystem in a period of one year between our two measurement studies. This justifies (even more) the necessity of the research conducted in this paper.

In order to fight the fake publishing phenomenon, the first step is to properly characterise the fake publishers and their behaviour. The current implemented solutions by BitTorrent portals identify fake publishers through the

[*] Corresponding author. Address: Avda. Universidad, 30. Leganés. 28911, Spain. Tel.: +34 916246232.
E-mail address: rcuevas@it.uc3m.es (R. Cuevas).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.bjp.2013.12.007

user account that they use to upload fake torrents to the portal. We show in the paper that this technique is inefficient since a fake publisher can generate as many user accounts as needed in those portals. Instead, the parameter that uniquely identifies a fake publisher is the IP address that it uses to perform its activity. Surprisingly, our data reveals that just 20 fake publishers (whose IP we identify) are responsible for injecting 90% of fake content in the BitTorrent ecosystem. Moreover, most of these IP addresses belong to Hosting Providers where fake publishers rent dedicated high-resource servers to perform their activity.

The fake publishing activity requires the attention of the publisher to create the needed user accounts and check when they have been removed. Furthermore, this activity requires dedicated resources (e.g. rented servers). This investment in time and resources can be only justified by a strong motivation behind the distribution of fake content. We have downloaded and manually inspected a large number of fake content published during our measurement period and found 3 different profiles among fake publishers: (i) a first group of fake publishers aims to spread malware using the popular BitTorrent system; (ii) a second set of users tries to attract BitTorrent users to scam websites in order to get economical benefit from the victims by using different scam techniques; (iii) the last group is formed by antipiracy agencies that upload fake versions of those content that they want to protect.

Our data shows that more than 99% of the published fake content is associated with the two first profiles. This supposes a very serious threat for the BitTorrent ecosystem since the activity of these publishers may lead to thousands of undesirable episodes of scammed users and computer infections. These findings suggest that new solutions need to be proposed in order to eliminate or at least reduce the number of fake content available in the BitTorrent ecosystem. Towards this end, we have designed and implemented TorrentGuard. This is a novel detection tool that allows to identify the IP address of the fake publisher, thus being able to report as fake each content published from this IP address at the moment of its publication. Based on the performed evaluation, TorrentGuard would be able to avoid more than 35 millions fake content downloads every year. This means, preventing hundreds of thousands of users to suffer from computer infections or scam incidents every year. We would like to acknowledge that the current implementation of TorrentGuard is an effective defense against the publishing techniques employed by fake publishers nowadays. Although, it is unlikely to be a defense that can fully eliminate the problem of fake files if fake publishers start using more sophisticated techniques, it is an important step forward to difficult the injection of fake content in the BitTorrent ecosystem. TorrentGuard can be currently used through a publicly available website and an official Vuze plugin.

The rest of the paper is structured as follows. Section 2 presents the background information. In Section 3 we describe our measurement methodology and present our dataset. Next, Section 4 characterises fake publishers, while Section 5 classifies them depending on the goal they pursuit with their activity. Section 6 shortly characterises the downloaders of the fake content. In Section 7 we describe and evaluate our solution to improve the detection of fake content. We also discuss possible countermeasures to TorrentGuard and their efficiency. Section 8 describes relevant works to this paper. Finally, Section 9 concludes the paper.

2. Background

In this Section we briefly describe the main aspects of the BitTorrent ecosystem making special emphasis on the procedure of publishing content on The Pirate Bay (and by extension on other BitTorrent portals) and specifically, on how fake publishers do it. This is summarised in Fig. 1. For a full description of the BitTorrent ecosystem we refer the reader to [21,35].

2.1. Main elements of BitTorrent ecosystem

- *BitTorrent portals:* these are webpages which index .torrent files, classify them into different categories and provide basic information for each file. These portals serve as rendez-vous points between content publishers and BitTorrent downloaders. The publishers upload their .torrent files to BitTorrent portals and the clients download them.
- *.torrent file:* this is a meta-information file including relevant information for the BitTorrent protocol such as: (i) the content infohash, this is a unique identifier of the content in the BitTorrent ecosystem; (ii) the IP address of the BitTorrent Tracker managing the content distribution process; (iii) the size of the content and the number of pieces forming the file.
- *magnet link:* this is an URI-like link that includes the infohash of a specific content and optionally the address of a tracker [1]. A user can launch a download process retrieving the magnet link instead of the .torrent file from a BitTorrent portal. Then, with the magnet link the user can obtain the .torrent file from other peers in the swarm.[1] The magnet links have recently become significantly important because from March 1st 2012 The Pirate Bay exclusively index magnet links [2].
- *BitTorrent Trackers:* these are servers which manage the BitTorrent download process of a given content. The set of peers downloading a given file is named *swarm*. The tracker maintains a list with the IP addresses and the download progress of all the peers forming the swarm associated to a specific content. Furthermore, when a new peer joins the swarm, it contacts the tracker in order to obtain a list of IP addresses of other peers participating in the swarm. By doing so, the new incomer is able to retrieve pieces of the content from these peers.
- *BitTorrent downloaders (peers):* these are clients forming the swarm that download and/or upload pieces of the content. We distinguish two types of peers. A *seeder* is a peer that possess a complete copy of the content, thus only uploads pieces whereas a *leecher* does not have the complete file so that it uploads and downloads pieces.

[1] Also the magnet link can be used as index to retrieve the associated .torrent file from the different DHTs implemented by BitTorrent clients [13].

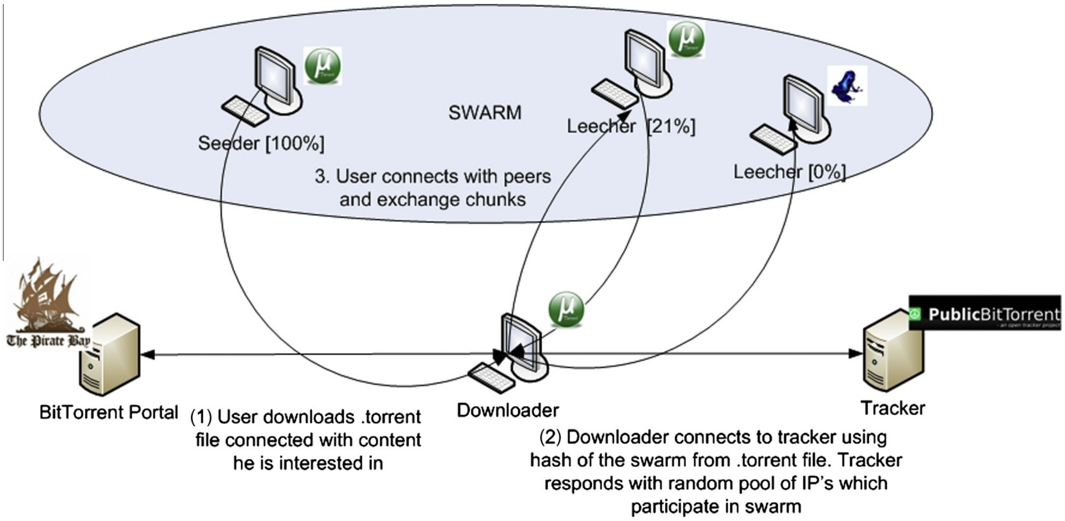
SWARM
Seeder [100%]
Leecher [21%]
Leecher [0%]
3. User connects with peers and exchange chunks
The Pirate Bay
PublicBitTorrent
BitTorrent Portal
(1) User downloads .torrent file connected with content he is interested in
Downloader
Tracker
(2) Downloader connects to tracker using hash of the swarm from .torrent file. Tracker responds with random pool of IP's which participate in swarm

Fig. 1. BitTorrent ecosystem basic functionality.

- *BitTorrent publishers:* these are the clients that make available the first copy of the content in the BitTorrent ecosystem.

2.2. Publishing a content in BitTorrent

When a publisher wants to publish a content in the BitTorrent ecosystem, it firstly creates a .torrent file. After creating the .torrent file, the publisher uploads it to one or more BitTorrent portals. For this purpose, it uses a user account (with a specific username) created in these portals. Furthermore, the publisher distributes the first copy of the content by acting as the initial seeder in the associated swarm. Hence, *the content publisher can be identified by the IP address of the initial seeder distributing the content and by the username utilised to upload content to a BitTorrent Portal.*

In this paper we specifically address the fake content publishing phenomenon in BitTorrent. A fake publisher is a user that exploits the BitTorrent ecosystem to publish fake content, this is, content that is different than what is expected from its name. A fake publisher makes available the fake content from a single IP address (or a limited number of IP addresses) that corresponds to the initial seeder of all its published content. Furthermore, a fake publisher typically creates a user account in a BitTorrent portal from which it uploads .torrent files associated with its fake content. Some portals, such as The Pirate Bay, remove this user account after some client reports that it is being used to publish fake content. Then, the fake publisher reacts by creating a new account to publish new .torrent files and this loop keeps repeating. Hence, contrary to the case of regular publishers (that can be identified by its associated username in the BitTorrent portal), fake publishers can exclusively be identified by its IP address. Finally, it must be noted that, to the best of the authors knowledge, the previously described technique based on users' reports is the only one used nowadays for detecting and deleting fake content.

2.3. Downloading a content in BitTorrent

When a user wishes to download a content, it first downloads the .torrent file associated to the content from a BitTorrent portal such as The Pirate Bay. Then, the user retrieves the IP address of the Tracker managing the swarm from the .torrent file and connects to it. The Tracker provides the user with a list (50–200) of IP addresses participating in the swarm along with the number of seeders and leechers forming the swarm. Finally, the user starts downloading the pieces of the content from the obtained IP addresses.

2.4. BitTorrent portals, the case of The Pirate Bay

We use The Pirate Bay as the reference BitTorrent Portal for our study. Previous works [35] have demonstrated that The Pirate Bay is a key element and the most important portal in the BitTorrent ecosystem. In particular, Alexa[2] indicates that The Pirate Bay receives double number of visits than the second most popular BitTorrent portal. Therefore, it seems the most suitable venue for a (fake) publisher to attract a large number of downloaders. Furthermore, in The Pirate Bay a publisher needs to create a user account in order to upload .torrent files to The Pirate Bay whereas other portals such as IsoHunt [3] use crawling techniques to obtain the offered content from third portals such as The Pirate Bay. In addition, The Pirate Bay offers the following relevant services to our study: (i) an RSS feed system in which each new published content is announced along with the username that uploaded the .torrent file to the portal; (ii) each registered user in The Pirate Bay portal has an individual webpage in which its published torrents are listed and (iii) The Pirate Bay removes the accounts, webpages and .torrent files of those users whose content is detected as fake. Typically, this happens after a client, who downloaded the content, reports its falseness to The Pirate Bay

[2] www.alexa.com.

administrators, then the administrators check if the content is actually fake and if so they remove the account, webpage and .torrent files of the misbehaving user.

The previous discussion supports that The Pirate Bay is the most interesting portal to be considered in order to understand the content publishing phenomenon in BitTorrent.

3. Measurement methodology

This section describes our measurement methodology to identify and characterise the main properties of fake publishers (i.e., users publishing fake content). For this purpose we crawl The Pirate Bay [9,35].

The main objective of our measurement study is to identify fake publishers. Towards this end, our measurement tool has three independent modules. The first one is responsible for finding the IP address and username of the publisher associated with each announced content in The Pirate Bay. For this purpose, the module is subscribed to the RSS feed of The Pirate Bay in order to learn each torrent just after its birth. After getting a new .torrent file the tool obtains the username that uploaded the .torrent file to The Pirate Bay. Furthermore, it uses the infohash within the .torrent file[3] to connect to the associated Tracker to obtain the IP addresses of the peers forming the swarm in its very initial stage. Then, it is very likely that we can find the IP address of the content publisher (initial seeder). Specifically, we face three different situations: (i) The tracker only reports the IP address of the initial seeder. This is likely to happen since we connect to the swarms just after the torrent birth. (ii) The tracker announces the presence of one seeder and few leechers in the swarm. Then, by connecting to all these peers and obtaining their bitfields (vector that shows the number of pieces that a peer possesses) we are able to identify which one is the initial seeder, and thus the content publisher. (iii) In some cases, the Tracker announces the presence of quite a few seeders in the swarm thus we cannot identify the initial seeder. This happens because the swarm has been formed before the torrent is announced in the RSS feed of The Pirate Bay portal. Therefore, using the described methodology we are able to characterise the content publisher by both its username and IP address in many cases.

The second module of our tool is responsible for identifying those publishers that are in fact fake ones. For this purpose our tool connects periodically (every 5 min) to The Pirate Bay webpage of each known publisher. If at some point The Pirate Bay webpage of an user has been removed we consider that the IP address associated with the removed account belongs to a fake publisher. Furthermore, we also collect the time that The Pirate Bay requires to detect and eliminate each fake publisher account.

Finally, our tool has a third module that counts the number of peers that connect to the swarm of each fake content in order to download it. Specifically, our tool systematically queries the Tracker managing the download of each fake content to obtain those IP addresses participating in the swarm. In order to accelerate this process we perform this task from four independent machines.

3.1. Dataset description

We have applied the described methodology between 30-04-2011 and 13-05-2011, in addition to 5 days of warm-up phase dedicated to identify the initial fake publishers' IP addresses. During the measurement period we have collected 29,330 torrents, from which 10,206 (35%) were identified as fake ones. Furthermore, we have collected the IP addresses of those peers participating in swarms associated with fake content until two instants: (i) the moment the content is removed from The Pirate Bay and (ii) the end of our measurement study.

4. Fake publishers characterization

Our results reveal that more than 1/3 of the content published in The Pirate Bay is fake. This shows an increasing trend in the number of fake content with respect to our previous study done one year earlier when the fake content represented a 30%. Therefore, it is critical to eliminate or at least reduce this huge number of fake content in the BitTorrent ecosystem. The first step towards this end is to identify who is responsible for publishing this fake content and characterising its behaviour. In this section we address this issue using the collected data. More specifically, we aim to answer questions such as: *How many fake publishers (i.e., IP addresses) are uploading fake content to the BitTorrent ecosystem?*, *From where (i.e., which ISP) they perform their activity?* or *How frequently they upload fake content?*

4.1. Number and contribution of fake publishers

Unexpectedly, we observe that only 71 IP addresses are responsible for those 4779 fake content for which we identified the initial seeder. This indicates that almost 70 fake content are published from each of these IPs in average. However, it is interesting to investigate the skewness in the contribution of each one of these fake publishers. Towards this end, Fig. 2 depicts the percentage of fake content published by the top x% of these fake publishers. The graph shows a skewed distribution where 10 IPs (14%) are responsible for publishing almost 75% of all the fake contents. Moreover, this number increases to 90% if we consider the top 20 IP addresses (28%). Therefore, we can conclude that a reduced number of just 20 fake publishers are responsible for poisoning the BitTorrent ecosystem. In the rest of the paper we focus on thoroughly studying this group of 20 fake publishers that we refer to as *Top Fake Publishers.*

4.2. Location of fake publishers

We have mapped the IP address of each one of the Top Fake Publishers to its correspondent ISP using the MaxMind database [26]. Surprisingly, 17 out of the Top 20 fake publishers operate from Hosting Providers. These are

[3] Note that we have implemented a new functionality that allows our tool to get the infohash also from magnet links.

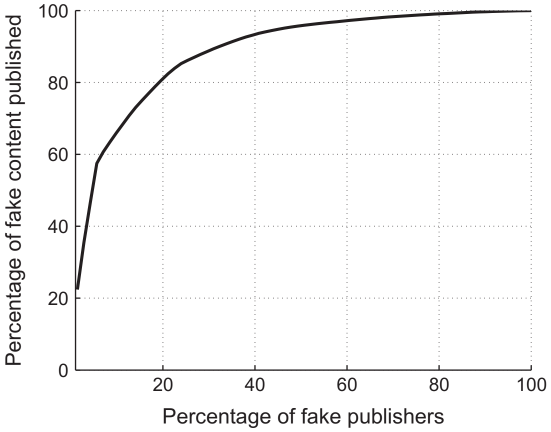

Fig. 2. Percentage of fake content published by the top x% fake publishers.

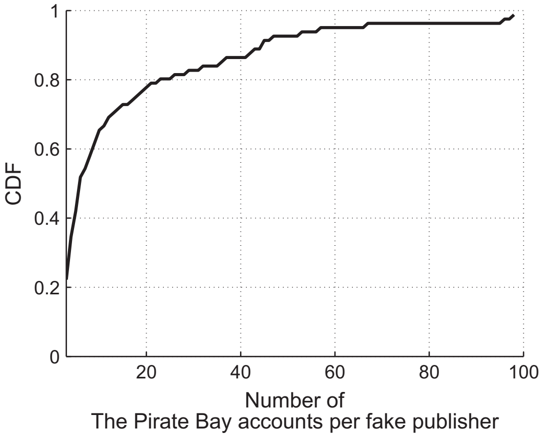

Fig. 3. CDF of the number of The Pirate Bay accounts per fake publisher.

companies dedicated to rent high-resources (cpu, memory and bandwidth) provisioned servers. Moreover, 70% of fake content is seeded from just two Hosting Providers named *OVH Systems* and *Obtrix* located at France and New Zealand, respectively. Note that fake publishers need resources in order to sustain the distribution of a large number of fake files [14] and anonymity due to the *illegal* activity performed. The use of rented servers in Hosting Providers provides both requirements.

The use of dedicated servers in Hosting Providers reveals that most fake publishers perform their activity from a stable IP since those servers typically have a static IP address configured. This makes them easily identifiable. In this sense, the use of anonymity services such as TOR [4] or proxy services [5,6] seems to be useful for fake publishers in order to make difficult their identification. However, we have not found that fake publishers identified in our dataset use such services. This suggests that the severe performance degradation associated to these anonymity services prevent fake publishers from using them. We further discuss these aspects in Section 7.5.

4.3. Pirate Bay accounts utilisation

The Pirate Bay solicits to solve a CAPTCHA [11] in order to create an account to avoid the automatic generation of accounts. Hence, fake publishers are obeyed to create their accounts manually, outsource this activity or use sophisticated methods to automatically solve captchas. Fig. 3 shows the CDF of the number of The Pirate Bay accounts used by each one of the 71 identified fake publishers. A fake publisher use (in median) 6 accounts in a period of 14 days. However, 5% fake publishers inject content using more than 58 different accounts in the same period. This represents an average number of 4 accounts per day. This result suggests that fake publishers need to dedicate time to track the availability of their accounts in order to generate new ones if needed.

Interestingly, we also observe a second strategy that although marginal is worth to report. In these cases, fake publishers hijack accounts with a legit publishing history. This provides a trusted reputation among the downloaders. Therefore, this could extend the time that a fake publisher could be injecting fake torrents before being reported. However, due to the required technical skills for applying this technique, this case represents less than 1% of all fake accounts.

4.4. Publishing strategies

Fake users follow two different strategies to upload fake contents to The Pirate Bay portal. On the one hand, we found users that publish a large number of fake content in a row (typically around 10) in just few seconds after creating a user account. Once the account is deleted, they repeat the same process from a new account. Around 70% Top Fake Publishers use this technique. On the other hand, 30% Top Fake Publishers upload just one or two fake contents with a username. This is a more conservative technique that extends the time that those fake accounts are active before being eliminated when compared to the previous case. Specifically, the accounts of those publishers using the first strategy are detected and then deleted in 92 min (in average) whereas the accounts of those using the second strategy are deleted in 253 min, thus their content is available 2.75 times more time. Unexpectedly, although the second strategy offers longer accounts' lifetime, it attracts only 47 downloaders per torrent (in average) in front of the 84 attracted by fake publishers using the first strategy. This happens because the fake publishers using the first strategy typically use popular names associated to their content whereas publishers using the second more conservative strategy do not use such popular names.

4.5. Strategies to attract downloaders

The main goal of fake publishers in BitTorrent is to produce as many downloads of their content as possible. Therefore, they need to offer torrents that sound very attractive for the downloaders. Towards this end, we have observed that fake publishers use three different strategies: (i) they assign to their files very popular names such as the title of the last released Hollywood movies; (ii) creating the false impression that the content has been published by a well-known and trusted user. For this purpose, the fake publisher names its content in the same way as a trusted user. For instance, eztv one of the most popular publisher in The Pirate Bay adds the signature [eztv] at the end of the title of its published files. Then, some fake publishers also add this signature to the title of their fake content; (iii) presenting attractive performance statistics (i.e., a high number of seeders and leechers) for the fake torrent. In this way, the fake torrent is perceived as a very popular torrent by the downloaders, that assume they will obtain a high download rate in case of selecting that torrent. To generate these fake statistics the publisher connects to the Tracker many times using a single IP but different ports. The tracker considers each of these IP + port pairs as a single peer and reports a high number of seeders and leechers. The Pirate Bay retrieves and presents these statistics from the Tracker.

In summary, the fake content publishing activity is performed from Hosting Providers facilities by just few dozens of users. Furthermore, fake publishers are aware of how the BitTorrent ecosystem works, thus they use sophisticated strategies in order to improve the success of their activity.

5. Fake publishers profiles

After characterising the behaviour of Fake Publishers, we still need to answer an important question: *What incentives a user has to publish fake content?* To answer this question we have downloaded up to 10 files published by each fake publisher in our dataset and manually inspected them. Our analysis reveals the presence of three different profiles: malware propagators, scammers and antipiracy agencies. Next, we describe in detail each one of these profiles.

5.1. Malware propagators

These users exploit the popularity of BitTorrent in order to rapidly propagate malware among thousands of users. On the one hand, the content published by some fake publishers is the malware itself. In this case, the content including the malware pretends to be typically a patch for a popular game, a key generator, etc. On the other hand, a second set of users use a more sophisticated technique. They publish a movie with a catchy title. The content has the standard size of a DivX movie (i.e., between 700 MB and 1 GB), and even sometimes includes a second small file with a real sample of the movie. Hence, the file has the appearance of a (non-fake) legitimate content. However, when a user downloads the content and tries to play the movie, it is requested to play it using Windows Media Player (WMP) in case a different player is run instead. When the movie is finally played with the WMP a pop-up window appears requesting to install new codecs along with an url link from where these codecs can be downloaded. Of course, the file including those pretended codecs is reported as a malware by anti-virus software.

5.2. Scammers

In this case, the fake publisher uses a similar technique to the sophisticated one described above. However, when the user plays the movie with WMP, it is automatically redirected to a website in the Internet. A second variant used by scammers is to provide a file protected with a password (typically a .rar file), and offer the user a link to a website in which the password can be obtained. Once the user gets into one of these websites, a credit card payment is requested in order to obtain some privilege to watch the downloaded movie (e.g., the password of the .rar file). In some other situations the user is informed that in order to check he is not a bot, a survey must be filled previously to watch the movie. This survey results to be a contest in which the client is obeyed to subscribe for a paid premium SMS service. These websites are often reported as scam on different forums.

We have performed a more detailed analysis of these websites. On the one hand, when a user wants to abandon the webpage several pop-up windows appear trying to change the user's mind and making the action of leaving the webpage at least bothersome. On the other hand, when a user enters some of these webpages, a pop-up window advertising a Facebook group of the webpage shows up. This pop-up does not react to the browser close button, rather, just by clicking on the "I like it" Facebook button the window closes. This method aims to increase the trust of the webpage so that users interpret it is a legit website. More importantly, this finding suggests that these scammers do not limit their activity to BitTorrent but they also try to capture victims from other popular applications such as online social networks.

5.3. Antipiracy agencies

The two previous profiles have harmful purposes. Antipiracy agencies instead, publish fake versions of the copyrighted content that they want to protect. For instance, sometimes this content includes antipiracy ads. The action performed by antipiracy agencies is limited in the number of contents (it is done exclusively under request from a company) and time (in the weeks before and after the content, e.g., a movie, is released).

In summary, we distinguish three different profiles among fake publishers. On the one hand, 65% of the Top Fake Publishers in our dataset are malware propagators and are responsible for around a 30% of the published fake content. On the other hand, a 35% of the Top Publishers are scammers and they published a 70% of the fake content during our measurement period. Finally, antipiracy agencies are responsible for a very small fraction of the fake content published due to the specificity of their actions. Therefore, most of the fake content published (by malware propagators and scammers) is potentially harmful, specially for not technically skilled downloaders. This represents a serious risk for the BitTorrent ecosystem, and by extension for the whole Internet. Then, solutions to eliminate or, at least, mitigate this problem must be proposed. We address this issue in Section 7.

6. Characterizing the downloaders of fake content

In this section we look at the studied phenomenon from the victims' side [14]. We first estimate the percentage of BitTorrent users belonging to each country. For this purpose, we leverage our dataset from [14] that includes a sample of more than 27 M BitTorrent users collected between April and May 2010. Then, we estimate the percentage of BitTorrent users for a given country X as the number of BitTorrent users located in that country divided by 27 M. Similarly, we estimate the percentage of victims within each country. To this end, we use the dataset presented in this paper that includes 1.35 M identified victims. Note that if the probability for a victim to belong to a specific country were equal for any country, then, the distribution of victims across countries would be similar to the distribution of BitTorrent users across those countries.

It is worth noting that the size of our sample of BitTorrent users and victims is enough large to derive meaningful results. In particular, the error introduced by the size of our sample of BitTorrent users (victims) to compute the percentage of them belonging to a given country can be calculated as the error produced by an hypothesis test for a proportion [33]. Assuming an infinite population of BitTorrent users (victims) this test indicates that the size of our sample of BitTorrent users (victims) leads to an error $\leqslant$0.03% (0.11%) with 99% confidence.

Table 1 shows the percentage of victim downloaders of fake content, the percentage of BitTorrent users and the ratio between these two percentages for the 10 countries with the largest number of victims. As indicated above, if victims were homogeneously distributed across countries, the computed ratio would be close to 1 for every country. However, this is not the case. On the one hand, we observe that some countries such as US, China and Brazil show a ratio >1. For instance, Brazil has a ratio equal to 1.59. This means that Brazil has 59% more victims than expected from an homogeneous distribution. On the other hand, countries such as UK, India or Spain show a value <1. For instance Spain has a ration equal to 0.47. This means, Spain only has 47% of the victims it should have from an homogeneous distribution.

Next, we study the number of fake content downloads performed by a single user. This helps to understand whether there are users that are highly vulnerable to the described threats. Fig. 4 shows the CDF of the number of fake content downloaded by each victim. We can see that 70% of the victims downloaded just 1 fake content. However, it is worth to note the presence of hundreds of users who downloaded multiple fake torrents during the measurement period.

Table 1
Demographics of BitTorrent users vs fake content downloaders per country (the third column represent the ratio column 1/column 2).

Country	Percentage of BitTorrent users downloading fake content (%)	Percentage of BitTorrent users (%)	The ratio
United States	12.40	10.42	1.19
China	6.27	4.20	1.49
Great Britain	4.60	6.26	0.73
Brazil	4.26	2.68	1.59
Italy	3.88	4.13	0.94
India	3.78	5.71	0.66
Canada	3.29	3.85	0.85
Spain	2.79	5.95	0.47
Austria	2.73	2.83	0.96
Poland	2.66	2.86	0.93

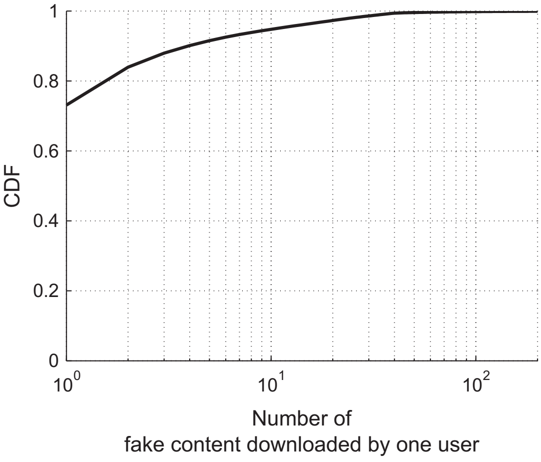

Fig. 4. CDF of the number of fake content downloaded by one user.

In a nutshell, the obtained results suggest that users from some specific countries (those having a ratio less than 1) are more skilled to identify fake content so being more protected against possible infections and/or scam episodes. More importantly, we have revealed that hundreds of users in our dataset download more than 5 fake content in a period of two weeks. These seems to be non-skilled users that are seriously exposed to scammers and malware propagators. These highly vulnerable users are the ones that will potentially obtain a higher benefit from the system described in the next section.

7. TorrentGuard

In the previous sections we have demonstrated that a large number of fake content (35%) is currently being published in the BitTorrent ecosystem, and what is worse, most of these fake contents are potentially harmful for those users that download them. We have also seen that the techniques used to remove these contents are inefficient and require heavy human intervention to: first, detect and report the falseness of a given content, and second, remove it from the BitTorrent portals (this is done by the portal administrator). Furthermore, the scope of the user reports is limited to a single BitTorrent Portal, thus the content is removed exclusively from this portal instead of the whole BitTorrent ecosystem.

In this Section we present our tool, named TorrentGuard, that aims to automatise and accelerate the process of detecting fake publishers. For this purpose, TorrentGuard identifies a fake publisher by its IP address instead of its username as it is done by BitTorrent portals such The Pirate Bay nowadays. By doing so, a fake content can be identified just after its birth since we can identify that

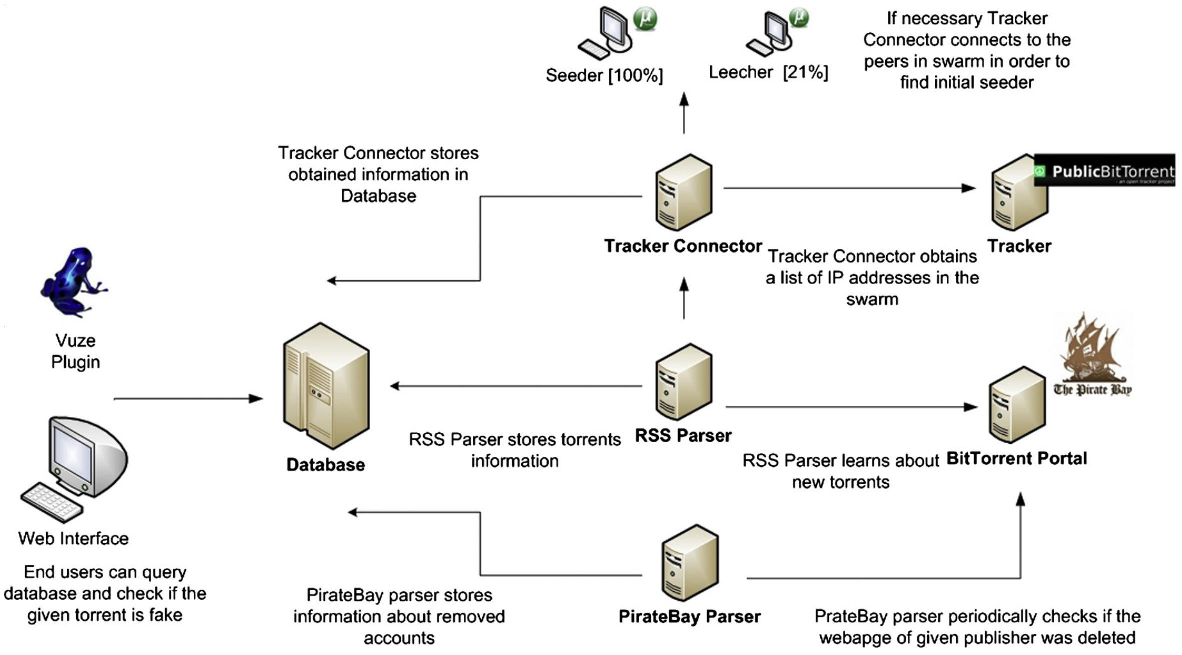
Seeder [100%]
Leecher [21%]
If necessary Tracker Connector connects to the peers in swarm in order to find initial seeder
Tracker Connector stores obtained information in Database
Tracker Connector
PublicBitTorrent
Tracker
Tracker Connector obtains a list of IP addresses in the swarm
Vuze Plugin
Database
RSS Parser stores torrents information
RSS Parser
The Pirate Bay
RSS Parser learns about new torrents
BitTorrent Portal
Web Interface
End users can query database and check if the given torrent is fake
PirateBay parser stores information about removed accounts
PirateBay Parser
PrateBay parser periodically checks if the webapge of given publisher was deleted

Fig. 5. The schema of TorrentGuard.

the IP address of the initial seeder belongs to a fake publisher. This allows to accelerate the detection process.

Furthermore, contrary to current techniques used by BitTorrent portals, TorrentGuard removes the fake content from the whole BitTorrent ecosystem because it reports the content infohash. Since the infohash uniquely identifies a content in the BitTorrent ecosystem, a user of TorrentGuard can identify the content as fake independently of the portal from which the .torrent file was retrieved or even if it was obtained from the BitTorrent DHT service.

We acknowledge that TorrentGuard may not be a final and definitive solution for the problem of fake content publishing in BitTorrent because in the future fake publishers can adopt publishing strategies that may not be detected by the current mechanisms implemented by TorrentGuard. However, it is a clear step forward to help in the identification of fake publishers and difficulty their activity.

In the rest of the section we present the details of the TorrentGuard implementation as well as the performance results obtained over a testing period of 14 days.

7.1. TorrentGuard implementation

Fig. 5 depicts a complete schema of TorrentGuard. It is composed by the following modules:

- *RSS Parser*: this module continuously monitors the RSS feed of The Pirate Bay portal. For each new published torrent the RSS Parser gathers the content infohash, from either the .torrent file or the magnet link,[4] and also the publisher's username. Furthermore, the RSS Parser sends requests to the Tracker Connector.
- *Tracker Connector*: this module is responsible for connecting to the tracker for every torrent obtained by the RSS Parser. The main objective of the Tracker Connector is to obtain the IP address of the initial seeder. In those swarms where the list of IP addresses returned by the tracker contains more peers than just one seeder, this module connects to all the peers and retrieves their bitfield in order to identify which one is the initial seeder. If the IP address of the initial seeder matches with one of those included in the blacklist of fake IP addresses, this torrent is marked as fake.
- *The PirateBay parser*: this module periodically connects to The Pirate Bay webpage associated to the different discovered publishers. Eventually, when a publisher's webpage (i.e., account) is removed from The Pirate Bay, The Pirate Bay Parser marks this username as fake.
- *Database*: It stores all the relevant information for the detection and evaluation of TorrentGuard. For each inspected torrent it stores detailed information such as the publisher's username and the initial seeder IP address (in case this is possible to obtain). More importantly, it includes two blacklists. The first one includes the infohashes of all the discovered fake torrents whereas the second one includes the IP addresses of fake publishers found so far.
- *Website interface* and *Vuze plugin*: The TorrentGuard functionality is publicly available throughout two different interfaces: a website[5] and a official Vuze plugin.[6] These interfaces provide access to the blacklist of fake

[4] From 1st of March 2012, our tool uses exclusively magnet links for this purpose, as The Pirate Bay stopped serving .torrent files from that date.

[5] This application is available at http://torrentguard.netcom.it.uc3m.es/.

[6] http://www.vuze.com/plugins/details/TorrentGuard.

torrents allowing a user to verify if a torrent file is associated to a fake content before starting the download process.

Next, we describe the functionality of the integrated TorrentGuard tool detailing the interaction between the different modules as well as the configuration parameters. It uses The Pirate Bay portal in order to identify new fake publishers and the IP addresses from where they operate. Towards this end, the RSS Parser continuously monitors the RSS feed of The Pirate Bay portal to learn about new torrents and identify for each torrent the publisher's username. Furthermore, it sends a query to the Tracker Connector that retrieves the IP address of the initial seeder (if it is possible). Both, the publisher's username and IP address (i.e., IP address of the initial seeder) are stored in the database. In parallel, The Pirate Bay Parser periodically connects to the webpage of the different discovered publishers within The Pirate Bay. If we find that a publisher's account is removed, this user and all its torrents are marked as fake. In addition, we annotate this publisher's IP address as a *potential fake IP address*. If three different accounts associated to a given publisher's IP address are removed from The Pirate Bay, we consider that IP as a *fake IP address*. From this moment on, any content published from that IP address is identified just after its birth and reported as fake. The number of removed accounts needed to mark an IP address as *fake* is a configurable parameter in TorrentGuard. We decided to set up this parameter equal to three because as we will show in Section 7.2 this produces a negligible ratio of false positive and false negatives. Decreasing this value makes TorrentGuard more aggressive and may increase the number of false positives. Instead increasing it makes TorrentGuard more conservative what may increase the number of false negatives.

Therefore, in the worst case, i.e., for new fake publishers, TorrentGuard employs the same time as The Pirate Bay to identify fake content. However, once the fake publisher's IP address has been identified, TorrentGuard is able to report fake content immediately after its publication. This provides a significant improvement compared to standard detection mechanisms. In other words, with TorrentGuard it is not necessary to manually report each fake user account as the existing solutions require.

Moreover, the current existing solutions are limited to the portal where they operate. For instance, in the case of The Pirate Bay, once a content is identified as fake it is removed from the portal but not from the BitTorrent ecosystem. Instead, TorrentGuard is a cross-portal solution that is able to identify the infohash of the fake content preventing its download independently of the source from where the user obtained the .torrent file: any BitTorrent portal or the DHT service.

In short, TorrentGuard is a novel tool that: (i) reduces fake content detection time since it uses IP-based detection instead of username-based detection and (ii) allows to identify a fake content in the whole BitTorrent ecosystem instead of just a single portal because it identifies the fake content using the infohash (a unique identifier of the content in the whole BitTorrent ecosystem) rather than the torrent-id in a specific portal.

7.2. TorrentGuard performance

We have evaluated the performance of TorrentGuard and compared it with the fake content detection mechanism used by The Pirate Bay during a testing period of 14 days. First, we count how many fake content published in The Pirate Bay are identified by the TorrentGuard just after its birth. Furthermore, we measure how long The Pirate Bay takes to identify these fake content. The obtained results show that TorrentGuard is able to early detect around 50% of the fake content uploaded to The Pirate Bay. Moreover, Fig. 6 presents the CDF of the time difference between the detection instant by TorrentGuard and by The Pirate Bay for these content. We observe, that TorrentGuard reduces the detection time 60 min in median. Moreover, the reduction in detection time is higher than 2 h for 20% of the fake contents, and for some cases it goes up to several days.

Although previous results already demonstrate the significant improvement provided by our tool compared to the state of the art solution, the final objective of TorrentGuard is reducing the number of download events associated with fake content, thus preventing BitTorrent users facing malware and scam. Then, if TorrentGuard was widely used, it would have prevented almost 390 K fake content downloads just during the 14 days of the evaluation period compared to The Pirate Bay. By extending this value to a complete year, we can state that TorrentGuard would be able to eliminate more than 10 millions fake content downloads per year compared to the solution used by The Pirate Bay. However, as stated before The Pirate Bay solution is specific for this portal but it is not applicable to the whole BitTorrent ecosystem. Specifically, in our dataset we identify around 950 K fake content downloads occurring after The Pirate Bay identifies these content as fake. Our proposed solution would be able to avoid also these downloads. Overall, TorrentGuard could avoid more than 1.35 millions fake content downloads in a period of two weeks. This means more than 35 millions in the course of a year. Finally, it is worth to mention that this impressive

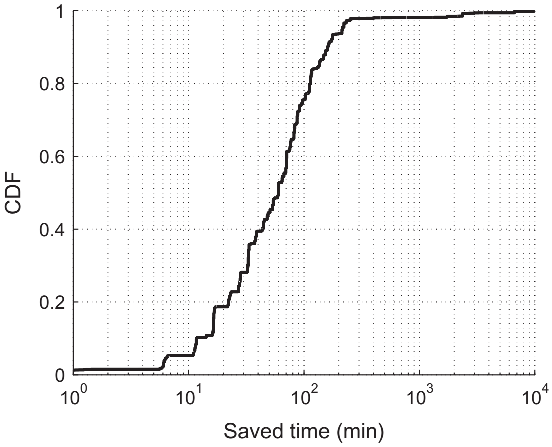

Fig. 6. CDF of the saved time in fake content detection when using TorrentGuard in front of The Pirate Bay.

number is only a lower bound since in our evaluation we only consider download events associated to few of the most important BitTorrent Trackers[7] but we do not consider download events coming from minor BitTorrent Trackers or the BitTorrent-associated DHT systems.

In a nutshell, our initial evaluation suggests that TorrentGuard could avoid up to tens of millions fake downloads per year. More importantly, this supposes (depending on the success of the fake publishers' strategies) up to hundreds of thousands computer infections and scam episodes. Hence, our evaluation shows very promising results to incentivize the BitTorrent community to use the TorrentGuard.

7.3. TorrentGuard efficiency

The efficiency of a detection system is typically characterised by the rate of false negative and false positive occurrences. In the specific case of TorrentGuard false negatives are represented by those fake torrents escaping our detection tool whereas false positives refer to those content classified as fake which actually are not fake ones.

Exhaustively measuring the false negative rate is not scalable in the case of TorrentGuard since it would require to download and manually inspect every single content classified as legit (i.e., non-fake) by TorrentGuard. This means up to dozens of thousands of content every month. Instead, we have performed an affordable evaluation by downloading few dozens of torrents classified as legit by TorrentGuard and manually inspected them. We did not find any fake torrent among them. We can state, however, that our tool discovers all fake contents which are also detected by The Pirate Bay.

In order to evaluate the false positives rate of TorrentGuard we use those Pirate Bay usernames whose account has not been deleted from The Pirate Bay but their contents have been classified by TorrentGuard as fake. The first intuition is that TorrentGuard may be mistaken for some of these usernames. In particular, we have downloaded content from each one of these referred Pirate Bay accounts and manually checked if it was fake or not. We have performed our experiment considering two different values of the number of reports needed to mark an IP address as fake, $R = 1$ and 3. On the one hand, for the case of $R = 1$, we found 1 false positive among more than 50 manually analyzed contents leading to a false positives rate <2%. On the other hand, for $R = 3$ we found none false positives.

In a nutshell, the performed evaluation suggests that TorrentGuard present a negligible rate of both false positive and false negative events.

7.4. Impact of TorrentGuard external dependencies

In this subsection we discuss the external dependencies of TorrentGuard and demonstrate that they represent a minor limitation for the system.

7.4.1. Dependency in The Pirate Bay

We have explained above that TorrentGuard bases its operation in The Pirate Bay portal. We have selected The Pirate Bay to conduct our investigation and to implement the first version of our tool due to the reasons discussed in Section 2.4. However, TorrentGuard and our measurement methodology can be easily adapted to other portals. The requirements for these portals are[8]: (i) having a service to announce new published torrents (e.g., RSS or a webpage) and (ii) having a system to report fake publishers (e.g., removing their accounts as in The Pirate Bay or marking fake content with special flags). It is worth to mention, that these two requirements are prety standard and widely offered by the most significant BitTorrent portals such as Mininova [7] or IsoHunt [3] and thus TorrentGuard and our measurement methodology can be easily adapted to them. Indeed, as future work, we plan to extend TorrentGuard to these other portals.

7.4.2. Dependency on users' reports

To the best of the authors' knowledge none existing software has the capacity of identifying a fake content under this context, i.e., the software should discern if the content is fake or not using as input the title of the content. For this purpose, we require the intervention of a human being. Hence, in practice we need at least one user report to identify a fake content and its associated fake publisher. As discussed earlier, TorrentGuard can be configured to mark a fake publisher's IP address as fake after the first user report (i.e., first removed fake username account). Note that our results in Section 7.3 indicate that this more aggressive configuration may incur in a slight increase of the false positives rate.

In summary, the previous discussion demonstrates that the external dependencies of TorrentGuard do not affect seriously its performance. On the one hand, the dependency of TorrentGuard in a single portal can be overcome by extending the operation of TorrentGuard to multiple portals. It is worth to mention that the effectiveness of TorrentGuard will be directly related to the significance of the associated portals. On the other hand, the dependency on users' reports is inherent to any fake content detection system and cannot be removed until new semantic-enhanced software is implemented. Hence, the best we can do is minimize the dependency in users' reports and TorrentGuard achieves this objective.

7.5. Potential countermeasures of fake publishers against TorrentGuard

TorrentGuard implements detection mechanisms that are efficient against the current publishing strategies of fake publishers. However, if TorrentGuard becomes widely used, it is likely that fake publishers will react by defining new strategies (i.e., *countermeasures*) that allow them to escape the control of TorrentGuard. In this section we

[7] For instance, http://www.openbittorrent.com/, http://www.publicbt.com/ that are the two major Trackers in the BitTorrent ecosystem.

[8] Note that these requirements apply also for a measurement study to be conducted in a different portal than The Pirate Bay.

present briefly the possible countermeasures that fake publishers may take and discuss their potential efficiency.

In particular, our tool identifies the fake publisher based on the IP address that it uses to publish fake content. Hence, fake publishers can use two reactive strategies. First, they can try to hide their IP address and second, they can try to perform their activity from a large number of IP addresses.

7.5.1. Hiding the fake publisher's IP address

The most straightforward way to hide an IP address is the utilization of a proxy service [5,6]. In this case TorrentGuard would interpret that the fake activity is being performed from the proxy IP address and would ban this one. Hence this technique is not efficient against TorrentGuard.

The next option would be to consider a network of proxies so that the fake publisher can use different proxies for publishing different fake contents. This type of anonymisation services exists in the current Internet and are commonly used by regular BitTorrent users to hide their IP addresses during the process of illegal content downloads. TOR is an example [4]. In TOR, traffic from a source (a fake publisher in our case) is bounced through several relays until it reaches the destination. Hence, the destination see that packets are coming from the IP address of the last (or *egress*) proxy and the IP address of the source cannot be identified. Furthermore, the egress proxy changes from one communication to another. Fake publishers could exploit the functionality of TOR to avoid its IP address being detected by TorrentGuard. TorrentGuard would then mark the IP addresses of TOR egress proxies as fake. Hence, if some non-fake publishers would use TOR, TorrentGuard would also mark their content as fake, thus increasing the false positives rate.

However, it is important to highlight that these anonymity services were not designed for supporting heavy traffic applications such as BitTorrent so that the performance offered to these services is typically poor. Indeed, TOR developers specifically state that TOR does not perform well with BitTorrent and is not designed for handling that type of traffic [8]. To evaluate the performance degradation that a fake publisher would experiment using TOR we have run a very simple test that compare the performance of a regular BitTorrent download vs a download done using TOR. For this purpose we have chosen a mid-popular torrent from The Pirate Bay (around 200 seeders and 300 lechers and 350.5 MB) and downloaded it 10 times with and without TOR. We have run the experiment in premises of our University (with a symmetric connection of 100 Mbps) and using a home ADSL (with a download and upload bandwidth of 6 Mbps and 320 kbps, respectively). The results are presented in Table 2. They suggest that operating BitTorrent over TOR reduces the performance around 3 times independently of the speed of the access link. Therefore, the utilization of anonymisation networks by fake publishers would notably impact the performance (i.e., content download time) of the swarms associated to fake content.

The fact that top fake publishers perform their activity from high speed connections suggest that the performance is a key aspect for the success of their activity. Hence, the utilization of anonymization networks poses a clear trade-off for a fake publisher between reducing the risk of being identified and reaching a larger number of victims. Furthermore, it is worth noting that a previous work by Le Blond et al. [23] have shown that the utilization of anonymization networks such as TOR does not guarantee full anonymity for BitTorrent users. For instance, a fake publisher that uses a BitTorrent DHT in order to reach a larger number of users could be identified even if it uses TOR. Hence, if in the future fake publishers start using anonymization networks, TorrentGuard could be extended to implement the techniques described in [23] to improve the efficiency of our tool in the detection of those fake publishers using anonymization networks.

Table 2
Average speed and download time of the file using BitTorrent with and without TOR.

Type of connection	Average time	Average speed (Mbit/s)
University	6 m 46 s	6.9
University (with TOR)	20 m 31 s	2.27
Home ADSL	9 m 59 s	4.68
Home ADSL (with TOR)	31 m 15 s	1.49

7.5.2. Using multiple IP addresses

The second countermeasure that a fake publisher could opt for is using a large number of IP addresses so that it always have undetected IP addresses to use for publishing fake content. Next, we estimate the number of IP addresses that a fake publisher would need to perform its activity in the presence of our tool. TorrentGuard identifies an IP address as fake after detecting 3 fake user accounts in The Pirate Bay. Thus, TorrentGuard marks a content as fake starting from the 4th account used by the publisher. We demonstrated in Section 4 that top 5% of fake publishers use in average 4 user accounts per day. Hence, a top fake publisher would need roughly 1 IP address per day in order to perform its activity and to avoid being blocked by TorrentGuard. In addition, we have seen that the activity of these publishers is performed from high speed servers located in data centres. Hence, these users would need to have access to around 30 IP addresses associated to high speed access links per month.

A second option would consist on the utilization of a botnet. In this case, the fake publisher would use the nodes in the botnet as proxies to forward the data packets from its main server. However, the uplink speed of a bot is likely to be significantly smaller to that of a server in a hosting facility. In particular, servers in hosting facilities offer uplink capacities between 100 and 1000 Mbps whereas residential computers (that are those more likely to form part of a botnet) present upload speeds in the range of few hundred kbps to few Mbps. Therefore, in order to forward the data from the main server the fake publisher should use concurrently between dozens and thousands of bots (depending on their speed). The IP addresses of these bots would be marked accordingly by TorrentGuard and, after they have been used to seed N (3 by default) fake content, those IPs would be blacklisted. Hence, a fake publisher should have access to a large botnet accounting with hundreds to thousands of infected computers in order to perform its activity over a period of a year without being detected. Furthermore, it needs to be able to continuously infect new computers in order to replace those already detected by TorrentGuard.

We conclude that the studied countermeasures would pose some, and in some case serious, difficulties for TorrentGuard to identify fake publishers. However, applying them would result in important technical challenges, high financial costs or higher risk of exposure due to unusual behaviour (e.g., requests and renewal of large number of IP address) for fake publishers.

7.6. TorrentGuard support to offline actions

The main goal of TorrentGuard is to identify IP addresses performing a malicious activity (i.e., injecting fake content) with different purposes such as malware propagation. These IP addresses can be reported to different third parties such as: Hosting Providers of identified IP addresses, cyber-crime police units, Internet security/antivirus companies, etc. Each one of these players could use the information reported by TorrentGuard to take different actions against the users behind those suspicious IP addresses. For instance, the Hosting Provider (after double-checking the activity performed from the reported IP address) could, for instance, ban the user access to the service and blacklist his credit card. Furthermore, police could initiate an investigation using the IP address reported by TorrentGuard to identify the person performing such activity. Finally, security/antivirus companies could include the reported IP addresses in a blacklist of potentially malicious IP addresses. Note that the aim of the previous discussion is not to make an exhaustive list of *what can be done* with the information gathered by TorrentGuard but just to provide few examples to illustrate the great potential that this information has further than our own tool. Hence, TorrentGuard can also serve as the source to trigger offline actions to defeat fake publishers.

7.7. TorrentGuard future deployment

In the previous subsections we have demonstrated the enormous potential of our TorrentGuard prototype. However, we believe that there is still room for improvement if BitTorrent portals and Trackers get involved in a next stage for the development of TorrentGuard. In this case, TorrentGuard could be extended to be a distributed platform in which trackers would identify the IP address of the initial seeder for every content and BitTorrent portals would identify the infohash of fake torrents. BitTorrent portals would provide the infohash of fake torrents to trackers so that these would be able to blacklist the IP address associated to fake publishers and eliminate their associated swarms. Furthermore, trackers would report back to portals the infohash of every new fake torrent published from a blacklisted IP address so that portals can immediately remove the associated .torrent file. The described system could store the information in a central server that interacts with both portals and trackers and maintain a central repository that can be accessed by users as well. Another option is running a complete distributed system in which trackers and portals exchange the information without the necessity of any central server. We believe that the involvement of major BitTorrent portals and Trackers in this project would lead to reduce the presence of fake content to negligible levels.[9] Furthermore, the involvement of Hosting Providers, police or Internet security companies would be useful to develop the potential of TorrentGuard as a reporting tool of suspicious activity from specific IP addresses that can be of high value to perform offline actions to defeat fake publishers as discussed in Section 7.6.

8. Related work

8.1. BitTorrent measurement

Several authors have used real data collection in order to understand different aspects of BitTorrent [15,17,18]. Different methods for measuring BitTorrent are described in [21]. However, only few works have looked at content publishers [10,35]. The most extensive study of characterisation of BitTorrent ecosystem is presented in [35]. This work includes discussion about BitTorrent publishers, defined by its username. We demonstrate in this paper that fake publishers cannot be identified by its username, instead they are identified by its IP address. The presence of the fake publishers was firstly mentioned in our previous work [14]. Based on our initial observation, in this paper we perform a thorough analysis of fake publishers and their published content revealing their target, incentives and strategies and propose a novel solution to prevent users from downloading fake content.

8.2. Fake content

There are several studies presenting the possible threats in the Internet. In [37] authors state that 40% of all computers are infected by botnets and can be controlled by attackers. Another study [27] reports high presence of malware and spyware content in the Internet. Few previous works have studied the malware propagation through P2P systems [20,31,36]. Specifically, Kalafut et al. [20] analyse LimeWire whereas Shin et al. [31] analysed KaZaa. These authors look at the problem from the content perspective instead of the fake publisher perspective used in this paper. This prevents them from discovering more sophisticated strategies as those reported in our study in which content is not the malware itself but includes a link to the malware. A similar content-based approach is applied by the FakeDetector program [16] that looks for fake hashes in DirectConnect hubs (central servers to which downloaders connect) and reports found fake content to

[9] The authors of this paper have started a process to contact different Trackers and Portals to sense their interest in participating in the deployment of the described project. Few major portals as The Pirate Bay and IsoHunt have shown an initial interest on our tool.

users and hub administrators. Finally, the authors of [20] propose to filter those content with a specific size since most of the malware content has specifically this size. Unfortunately, this solution is not valid for BitTorrent. Instead, we propose a more sophisticated solution (TorrentGuard) that provides early detection of fake content.

9. Conclusions

This paper presents the first comprehensive study about fake content in the BitTorrent ecosystem. For this purpose we use real data collected during a large-scale measurement study. The obtained results demonstrate that 35% of all the content is fake. Moreover, just a few tens of users are responsible for most of the published fake content. Furthermore, more than 99% of the fake torrents are associated with either malware or scam websites. This represents a serious threat for the BitTorrent ecosystem that must be eliminated or at least mitigated. Towards this end, we have implemented TorrentGuard, a novel tool for early detection of fake content. Based on our initial evaluation the widely usage of this tool may prevent the download of millions of fake content every year, thus contributing to reduce the number of computer infections and scam episodes faced by BitTorrent users.

Acknowledgments

The research leading to these results has been partially funded by the European Union's FP7 Program under the project eCOUSIN (318398), the Spanish Ministry of Economy and Competitiveness under the eeCONTENT project (TEC2011-29688-C02-02), and the Regional Government of Madrid under the MEDIANET project (S2009/TIC-1468).

References

[1] http://en.wikipedia.org/wiki/Magnet_URI_scheme.
[2] http://thepiratebay.se/blog/206/.
[3] http://www.isohunt.com.
[4] https://www.torproject.org/.
[5] http://btguard.com/.
[6] http://www.gtguard.com/.
[7] http://mininova.org/.
[8] https://blog.torproject.org/blog/bittorrent-over-tor-isnt-good-idea.
[9] Alexa. http://www.alexa.com/topsites/.
[10] S. Le Blond, A. Legout, F. Lefessant, W. Dabbous, M. Ali Kaafar. Spying the world from your laptop. LEET'10, 2010.
[11] CAPTCHA. http://www.captcha.net/.
[12] D.R. Choffnes, F.E. Bustamante, Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems, ACM SIGCOMM (2008).
[13] S.A. Crosby, D.S. Wallach. An analysis of bittorrent's two kademlia-based dhts. Technical Report TR-07-04, Department of Computer Science, Rice University, 2007.
[14] R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, C. Guerrero, R. Rejaie. Is content publishing in bittorrent altruistic or profit-driven? ACM CONEXT 2010, Philadelphia, USA.
[15] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, P. Rodriguez. Deep diving into bittorrent locality. IEEE INFOCOM 2011, Shanghai, China.
[16] FakeDetector. http://www.sourceforge.net/projects/fakedetector/.
[17] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, X. Zhang. Measurements, analysis, and modeling of bittorrent-like systems. In ACM IMC'05.
[18] T. Isdal, M. Piatek, Krishnamurthy. A, T. Anderson, Leveraging bittorrent for end host measurements, In PAM, 2007.
[19] R. Izhak-Ratzin, H. Park, M. van der Schaar, Reinforcement learning in bittorrent systems, in: Proceedings of INFOCOM, 2011.
[20] A. Kalafut, A. Acharya, M. Gupta. A study of malware in peer-to-peer networks, in: 6th ACM SIGCOMM Conference on Internet measurement, IMC, 2006.
[21] M. Kryczka, R. Cuevas, A. Cuevas, C. Guerrero, A. Azcorra, Measuring Bittorrent ecosystem: techniques, tips and tricks, IEEE Commu. Mag. (2011).
[22] N. Laoutaris, D. Carra, P. Michardi. Uplink allocation beyond choke/unchoke or how to divide and conquer best, in: Proceedings of ACM CoNEXT, 2008.
[23] S. Le Blond, P. Manils, A. Chaabane, M. Kaafar, C. Castelluccia, A. Legout, W. Dabbous. One bad apple spoils the bunch: exploiting p2p applications to trace and profile tor users. USENIX LEET'11, 2011.
[24] N. Liogkas, R. Nelson, E. Kohler, L. Zhang. Exploiting bittorrent for fun (but not profit). in: IPTPS, 2006.
[25] T. Locher, P. Moor, S. Schmid, R. Wattenhofer. Free riding in bittorrent is cheap, in: HotNets, 2006.
[26] MaxMind. http://www.maxmind.com/.
[27] A. Moshchuk, T. Bragin, S. Gribble, H. Levy. A crawler-based study of spyware on the web. Internet Society Network and Distributed System Security Symposium (NDSS), 2006.
[28] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, A. Venkataramani. Do incentives build robustness in bittorrent? in 4th USENIX Symposium NSDI, 2007.
[29] Sandvine. Fall 2010 Global Internet Phenomena Report. Available at: http://www.sandvine.com/news/global_broadband_trends.asp.
[30] Alex Sherman, Jason Nieh, Clifford Stein. Fairtorrent: Bringing fairness to peer-to-peer systems. in: Proceedings of the ACM CoNEXT, 2009.
[31] S. Shin, J. Jung, H. Balakrishnan. Malware prevalence in the kazaa file-sharing network, in: 6th ACM SIGCOMM conference on Internet measurement, IMC 2006, 2006.
[32] M. Sirivianos, J.H. Park, R. Chen, X. Yang. Free-riding in bittorrent networks with the large view exploit, in International Workshop on Peer-to-peer Systems (IPTPS), 2007.
[33] M. Triola. Elementary Statistics. Addison-Wesley.
[34] H. Xie, Y.R. Yang, A. Krishnamurthy, Y. Liu, A. Silberschatz, P4p: Provider portal for applications, ACM SIGCOMM (2008).
[35] C. Zhang, P. Dhungel, D. Wu, K.W. Ross. Unraveling the bittorrent ecosystem. IEEE Trans. Parallel Distrib. Syst.
[36] L. Zhou, L. Zhang, F. McSherry, N. Immorlica, M. Costa, S. Chien. A first look at peer-to-peer worms: Threats and defenses, in: Proceedings of the IPTPS, February, 2005.
[37] Zhaosheng Zhu, Guohan Lu, Yan Chen, Z.J. Fu, P. Roberts, Keesook Han. Computer software and applications, COMPSAC '08, 2008.

Rubén Cuevas obtained his PhD and MSc in Telematics Engineering and MSc in Telecommunications Engineering at University Carlos III of Madrid (Spain) in 2010, 2007 and 2005 respectively. Furthermore he received his MSc in Network Planning and Management from Aalborg University (Denmark) in 2006. He is member of the Telematics Engineering Department and NETCOM research group at University Carlos III of Madrid since 2006, where he is currently Assistant Professor. Between January and December 2010 he has been Courtesy Assistant Professor at the CIS Department at University of Oregon. He has been involved in several international and national research projects. Moreover, he is coauthor of more than 20 papers in prestigious international journals and conferences such as IEEE Network, Elsevier Computer Network, ACM CoNEXT, IEEE Infocom, IEEE P2P. His main research interests are Content Distribution, P2P Networks, Online Social Networks and Internet Measurements.

Michal Kryczka received his Master degree in Computer Science in 2008 in Technical University in Lodz (Poland) and in 2009 in University of Carlos III in Madrid (Spain). Since 2008 he is Research Assistant (financed by FPU scholarship from Spanish Ministry of Education) in Institute IMDEA networks, and a PhD candidate at University of Carlos III in Madrid. Between January and August 2012 he has been a Visiting Research at Columbia University. His current research lines include peer-to-peer technologies, Internet measurements and social networking.

Roberto González Sanchez was born in Madrid, in 1984. He obtained his MSc in Telematics Engineering and MSc in telecommunications engineering at University Carlos III of Madrid (Spain) in 2011 and 2009, respectively. He is member of the Telematic Engineering Department and NETCOM research group since 2010, where currently he is Teaching Assistant and PhD student. He has been teaching during the last 2 years in different degrees (Telecommunication Systems, Audiovisual Systems, etc.).

Angel Cuevas received his Master in Telecommunication Engineering, his MSc and Ph.D in Telematic Engineering from the University Carlos III of Madrid in 2006, 2007 and 2011, respectively. He is currently a Postdoc Researcher at the Department of Wireless Networks and Multimedia Services in the Network and Service Architecture group. His research is focused on the areas of social networking, peer-to-peer systems and sensor networks. Dr. Cuevas has published more than 20 papers in high quality international conferences such as ACM CoNEXT, ACM MSWiM, IEEE ICC, IEEE ISCC, etc. and journals such as Elsevier Computer Networks, IEEE Communications Letters, IEEE Communications Magazine, Sensors, etc. He was recipient of the Best Paper Award in ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems 2010 (ACM MSWiM '10).

Arturo Azcorra received his M.Sc. degree in telecommunications engineering from UPM in 1986 and his Ph.D. from the same university in 1989. In 1993 he obtained an M.B.A. from the Instituto de Empresa. He holds a double appointment as full professor (with chair) at the Telematics Engineering Department of University Carlos III of Madrid and director of Institute IMDEA Networks where he conducts his research activities. He has participated in and directed 49 research and technological development projects, including European ESPRIT, RACE, ACTS, and IST programs. He has served as a Program Committee member in many international conferences, including several editions of IEEE PROMS, IDMS, QofIS, CoNEXT, and IEEE INFOCOM. He has published over 100 scientific papers in books, international magazines, and conferences.