Journal of Universal Computer Science, vol. 14, no. 2 (2009), 465-487
submitted: 20/10/08, accepted: 25/1/09, appeared: 28/1/09 © J.UCS

The Topology Change Attack: Threat and Impact

Mahdi Amine Abdelouahab and Abdelmadjid Bouabdallah
(University of Technology of Compiegne
HeuDiaSyc Laboratory, UMR CNRS 6599
P.O. Box 20529, 60205 Compiegne, France
mabdelou@hds.utc.fr, bouabdal@hds.utc.fr)

Mohamed Achemlal and Sylvie Laniepce
(Orange Labs
42 rue des Coutures PB 6243, Caen, France
mohammed.achemlal@orange-ftgroup.com
sylvie.laniepce@orange-ftgroup.com)

Abstract: Peer to peer (P2P) network has received in past few years a significant attention, especially such file sharing network as eDonkey [Kulbak and Kirkpatrick, 2005] or BitTorrent [Cohen 2008]. The shift from the classical client-server based paradigm of the Internet, with a clear distinction between information providers and consumers, towards consumers sharing information among each other led to the rise of the P2P paradigm. This distributed architecture, enables users to share content autonomously; Information remains at end-users' computers at the edge of the Internet and is not gathered and organized at central servers. While P2P has emerged as a new hot communication concept among the Internet users, security concerns still taking its first steps. The deployment of classic security protocols to provide services such as node authentication, content integrity or access control, presents several difficulties, most of them are due to the decentralized nature of these environments and the lack of central authorities. The fast emergence and the open nature of P2P applications, make appearing new attacks, so it is extremely important to study them and develop new counter measurements. Furthermore, existing studies focus on attacks that disrupt the overlay functioning and does not take in account their impact on ISPs (Internet Service Provider) infrastructure. In this paper, we present the Topology Change Attack [Abdelouahab et al. 2008] that harms the underlying networks (ISPs infrastructure) by unbalancing the P2P workload repartition. In order to evaluate and validate the TCA impact, we developed a new cycle-based simulator which simulates eDonkey clients hosted on different ISPs. The obtained results are very interesting and show the increasing of inter-ISPs traffic when a Topology Change Attack is conducted.

Key Words: Peer-to-Peer, Topology change, File sharing application, attack, PeerSim
Category: C.2

1 Introduction

The need for large-scale data sharing between autonomous and decentralized systems on the Web gave rise to the concept of P2P systems. A P2P system is a distributed collection of peer nodes (peer), when each node plays dual roles as Client and Server. Thus, making servers less used in the functioning of the system. P2P is designed to facilitate resource sharing (content, storage, CPU cycles) between a large number of computers, bypassing a centralized authority such as a centralized server. Today, different areas are using P2P architectures: *grid computing*, *IP telephony*, and *Instant Messaging*.

EDonkey is one of the most popular and widely-used file sharing P2P systems. It uses servers just to index shared files and not to store them. To join the eDonkey network, a client node has to (1) register with a known eDonkey server, (2) record its information on the server: *IP address* and port number, (3) publish its shared files: *name*, *size*, *extension*, etc. Once this connection is established, the client node is ready to download files. When a node wants to download a file, (1) it sends a query to the server using keywords; (2) the server replies with a list of shared files that may interest the client; (3) by choosing a file F from this list, the client sends a new query to the server; (4) the server replies with a list of nodes (IP addresses) from which the client may download the desired file; (5) finally, the client downloads the file from several sources simultaneously.

Due to the recent surge of P2P systems with a large number of users, P2P systems can be a potential vehicle for new attacks [Chellappan et al. 2005]. Some attacks are similar to the known Internet attacks as Malwares [Kalafut et al. 2006] and DoS attacks [Long et al.], while, other P2P attacks have been developed specifically for P2P networks [Hatahete et al. 2008]. These attacks are very dangerous, and may have an impact on the application and/or the underlying network. Consequently, the P2P service could be stopped. Besides, Peer-to-peer systems, which are realized as overlays on top of the underlying Internet routing architecture, contribute a significant portion of todays Internet traffic [Aggarwal et al. 2007]. The huge P2P traffic also poses a significant traffic engineering challenge to the ISPs [Nakao 2005]. Because P2P systems implement their own routing in the overlay topology independently from the Internet routing.

In this paper, we present an overview of the P2P attacks, discuss their impact on the ISPs network infrastructure, and then, study the Topology Change Attack introduced in [Abdelouahab et al. 2008]. This attack, when well conducted, allows an attacker to disrupt the eDonkey network functioning and may lead to a Denial of Service at the level of ISPs equipment(*routers*, *switches*). Furthermore, we developed a new P2P cycle-based simulator which simulates eDonkey clients hosted on different ISPs, in order to evaluate and validate the TCA impact. The rest of this paper is organized as follows: Section 2 summarizes the eDonkey protocol specifications. Section 3 lists P2P attacks and their impact. In section 4, we define the topology change attack in the case of eDonkey and synthesise this attack in three attack cases, showing their real impact. Section 5 presents our simulation model and validates the proposed attack by using a modified version of PeerSim. Section 6 concludes this paper with a summary of major research contributions and outlines future works.

2 The eDonkey Protocol

The eDonkey network is one of the most popular P2P file sharing applications. It is implemented by client softwares such as edonkey2000 and eMule. This client provides the user with advanced features such as:

- File searching functionality which is based on a metadata file (blocs). The user can specify the file type and size.
- Ability to download a file from one or several clients: this functionality reduces downloading and files propagation time in the eDonkey network.
- Corruption detection using the MD-4 hash for each bloc.

EDonkey is composed of two entities, an eDonkey server and the client software. In order to help other clients to locate shared files, the eDonkey server maintains a list of clients (IP addresses) and the files they propose to share. The client software is preconfigured with a list of known eDonkey servers. When the client establishes a connection with a single eDonkey server, the server assigns it an Identifier called Client-ID. This attribute identifies the client for the whole session. The client publishes then its shared files on the server.

Each client in the eDonkey network maintains a list of servers in the *server.dat* file, which can be obtained in different ways: (1) From the eMule's source code (preconfigured), (2) From known web sites, (3) From eDonkey servers (4) From eDonkey clients. When a client initiates a search request, it specifies one or several keywords. Client queries could be complex and restrictive, taking into consideration file size, extensions and availability. Upon receiving this request, the server draws up a list of files matching the keywords, and sends it back to the client. After choosing a file F from this list, the client generates a new query. Once this new request is received, the server replies with a list of sources sharing file F. The client can initiate the same search request to all or to some of the rest of the eDonkey servers, using UDP protocol. Once the client receives the list of sources, it initiates the download directly from these sources, bypassing the server. In eDonkey, files are divided into blocks of 9.5MB, and an MD-4 checksum is computed for each block. The checksums related to all blocks of a given file are then used to compute a new MD-4 checksum which is the file identifier Called file-ID. As soon as a block is downloaded, it is automatically shared over the eDonkey network. This smart mechanism spreads file sources rapidly. Block and file checksums can be propagated between clients to detect block and file corruption more efficiently.

3 P2P Attacks

Most of Peer-To-Peer attacks are well known in client-server environments. The deployment of classic attack preventing mechanisms and protocols, presents several difficulties, most of them due to the decentralized nature and the overlay routing of these environments. In practice, one real P2P attack may consist of a single attack, as described below, or a combination of several P2P attacks. The large number of connected clients and the P2P overlay application nature offer a propitious environment for attackers to propagate faster their attacks.

3.1 P2P Worms

A Worm is a program that propagates itself over a network, reproducing itself as it goes [Joukov et al. 2007]. Due to its recursive nature, the spread rate of a Worm is very huge and poses a big threat on the Internet infrastructure as a whole. The purpose of a Worm is to achieve a high infection rate within the targeted hosts (i.e. infects the largest number possible of vulnerable machines). Modern Worms may control a substantial portion of the Internet within few minutes. No human mediated response is possible to stop an attack that is so fast. The possible devastating effects on the Internet operation are hard to underestimate. It was reported in the FBI/CSI survey [Richardson et al. 2007], that in 2007 52% of the detected network attacks were viruses' attacks (worms/spyware). Besides, the traffic generated by the worm propagation is so huge that it can be considered as a DDoS attack on the whole Internet and could be used to bring down the Internet infrastructure of whole countries [Hatahete et al. 2008]. Therefore, a huge number of research were carried out in order to develop Worm detection and containment systems. However, there is a new trend of Worms that is emerging and which have a huge destruction potential, such Worms are called Peer-to-Peer Worms. A P2P Worm is a Worm that exploits the vulnerabilities of a P2P network in order to propagate itself over the network and accelerate its propagation throughout the Internet. P2P Worms could propagate faster than the old-fashion Worms. Furthermore, they are expected to be one of the best facilitators of Internet Worm propagation due the following reasons: [Staniford et al. 2002, Nassima 2002, Chen et al. 2003]

- P2P systems have a large number of registered active hosts which easily accelerate Internet Worm propagation, as hosts in P2P systems are real and active.
- Some hosts in P2P systems may have vulnerable network and system environments, e.g., home networks.

– Hosts in P2P systems maintain some neighbors identifiers information for routing purposes. Thus, infected hosts in the P2P system can easily propagate the Worm to their neighbors, which continue the Worm propagation to other hosts and so on.

– The programs are often executed on user's desktops rather than servers, and hence, tey have access to sensitive files such as passwords, credit card numbers, address booksetc

– The use of the P2P network often entails the transfer of "grey" content (e.g., pornography, pirated music and videos), arguably making the P2P users less inclined to draw attention to any unusual behavior of the system that they perceive.

In order to identify the characteristics of Worms, we need to understand how it propagates itself over a network. A typical Worm works as follows: it first scans the Internet to find potential victims (i.e. information collection). Once it locates a machine, the Worm tries to probe it by exploiting a common vulnerability. If succeed it transfers a copy of its malicious code to the new victim and so on. The key of a successful Worm is its propagation speed rather than the vulnerability it exploits. Since current deployed detection and containment systems are able to block the spreading of slow Worms, a Worm should propagate quickly, in order to achieve a high infection rate. Choosing an efficient scanning strategy enables the Worm to reach a large population in a very short time. Based on the scanning strategies of P2P Worms, they could be classified into two broad categories: *passive Worms* and *active Worms*. Passive Worms are identical to viruses in the sense that they do not search for new victims. However they await them. Besides, active Worms search for vulnerable targets. So, they are more dangerous and propagate faster than passive Worms.

Once a Worm have successfully infected a system, it continues to search for new victims [Chen et al. 2003, Zhou et al. 2005] and can spread through the network automatically. Usually, Worms spend a considerable time during their propagation to generate random IP addresses. Each generated address is the address of a potential victim to which the worm will propagate. In P2P, worms detect new victims by following the overlay topology.

In [Chellappan et al. 2005], authors studied the impact of Peer-To-Peer systems on active worm propagation over Internet. Their main goal is to develop an analytical methodology that can be used qualitatively to better understand the worm's impact using P2P on the Internet. The presented simulations show that a P2P system can be a vehicle for the active worm attacker to achieve fast propagation. P2P Worm propagation strategies are classified as follows [Chellappan et al. 2005]:

1. Pure Random-based scan (PRS): in this strategy, the P2P worm behaves as in classic networks. It selects an IP address randomly and tries to jump to it.

2. Offline P2P based Hitlist Scan (OPHLS): in this strategy, the worm collects IP addresses by acceding to the overlay routing table of its first victim. Thereafter, the Worm jumps to these addresses following a tree scheme. Each time the Worm reaches a new victim, it divides the Hitlist for each duplicate. This mechanism is used recurrently until the last machine in the Hitlist is reached. For a faster propagation, this Worm can use the PRS strategy to infect new victims.

3. Online P2P Scan (OPS): this strategy is similar to the previous one. The only difference is that the client to which the Worm will jump must be connected to the P2P network. In OPHLS, this condition does not exist. The Worm uses its capacity to reach the maximum number of peer neighbours.

To our knowledge, no research work have been conducted on Worms propagation over the eDonkey network. As soon as a new vulnerability is discovered, new malicious software(Worms, virus) exploiting this vulnerability will appear. So we forecast the emergence of new Worms which will exploit eDonkey network features. In [Hatahete et al. 2008], the authors analyzed the impact of a novel Worm propagation model on BitTorrent. BitTorrent is particularly vulnerable to topology aware active Worms. Topology aware Worms use the topologic information hold by their victims to find new victims. Such Worms are capable of quickly flooding the Internet while escaping current deployed intrusion detection systems. Moreover, in order to boost its initial propagation, the Worm uses a trackers hitlist consisting of the most crowded swarms. This mechanism allows the Worm to find newer victims even faster than traditional scanning Worms. This combination of both scanning strategies is fatal, because it provides the Worm with certainty discretion and speed. The analysis of this propagation scheme shows that it can achieve a 300% increase in its propagation speed in comparison with traditional scanning Worms. Their work provides important guidelines for P2P system design and control that address the concerns of active Worms and to develop efficient containment and intrusion detection systems.

3.2 The Sybil attack

The Sybil attack [Douceur 2002], involves creating several identities for one entity over the P2P network; in other terms, the redundancy of the same entity with several identities over the system. This attack does not have an important security impact, but it can be used as a first step to conduct other attacks.

Several P2P systems mitigate hostile peers relying on existence of multiple, independent remote entities. A P2P entity can accept new identities by trusting the collective assurance of multiple signatures, analogous to the PGP web of trust [Zimmerman 2006] for human entities. To prevent Sybil attacks, a P2P system must own the ability to determine whether two ostensibly different remote entities are actually different. Douceur [Douceur 2002] demonstrated that in the absence of a trusted identification authority, a Sybil attack can severely compromise the initial generation of identities, thereby undermining the chain of vouchers.

Identification authorities can take various forms, not merely that of an explicit certification agency such as VeriSign [Verisign]. For example, the CFS cooperative storage system [Dabek et al. 2001] identifies each node (in part) by a hash of its IP address. The SFS network file system [Mazieres et al. 1999] names remote paths by appending a host identifier to a DNS name. The EMBASSY [Lefebvre 2000] platform binds machines to cryptographic keys embedded in device hardware. These approaches may thwart Sybil attacks, but they implicitly rely on the authority of a trusted agency (such as ICANN [ICANN 2004] or Wave Systems [Lefebvre 2000]) to establish identity.

In [Kulbak and Kirkpatrick, 2005], authors reported that an eDonkey server accepts a maximum of three client connections from the same IP address. This mechanism helps to prevent the Sybil attack but still limited. An attacker can bypass it by possessing a large number of virtual IP addresses. If it succeeds, this attack has one of the following impacts:

- By registering a high number of clients, an attacker eforces a server to refuse future legitimate connection requests. Thus, it creates a DoS situation on the eDonkey network. An eDonkey network may be disrupted if a Sybil attack is launched against all its servers.
- For each identity, the attacker shares a large number of files. This forces the server to use all its CPU resources when refreshing its list of shared files.

Some researchers are interested in the Sybil attack, however they more investigated more structured P2P systems [Lesniewski 2008]. In [Konrath et al. 2007], authors present an attack strategy based on the Sybil attack which strive a BitTorrent swarm to fail. The results show that even modest amounts of resources are used to attack and hinder BitTorrent swarms, both with File Piece Lying and Sybil attack, the impact is very dramatic. The Sybil attack is still an open research topic in unstructured P2P systems like eDonkey and BitTorrent.

3.3 The Eclipse attack

Existing research work [Singh et al., Singh et al. 2006] are conducted more on structured than on unstructured P2P systems. Eclipse is an overlay attack. It should disrupt both P2P functionality and also the underlying network. Eclipse [Singh et al., Singh et al. 2006] is defined as an attack where one or several strategic nodes are under control with the intention of eavesdropping messages, elaborating a denial of service, or redirecting overlay flow. A

key question to deploy the Eclipse attack over a P2P network is which nodes are strategic on it. As shown in figure 1 strategic nodes in a pure P2P network should be those that are in the core of the network. In this case the attacker should apply a Sybil attack by introducing several identities for one entity. Otherwise, in hybrid P2P architectures, strategic nodes should be super-peers because of their position in the overlay network.

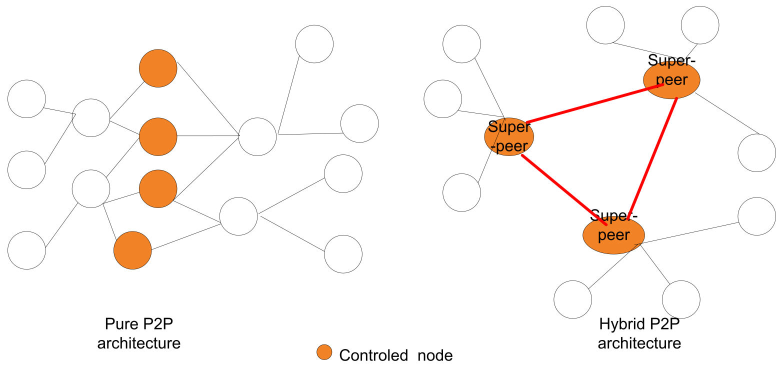


Figure 1: Strategic nodes on Pure and Hybrid P2P architectures

Redirection flow attack on eDonkey could be the consequence of an Eclipse attack. To accomplish this attack against peer_A, an attacker needs to control at least one eDonkey server. Upon receiving client requests, the malicious server replies systematically with peer_A IP adress as a source sharing the requested file. These replies do not take into consideration the keywords mentioned by clients. Thus, a Distributed Denial of Service attack is generated against peer_A.

3.4 Malwares

Malware or malicious software refers to a file that contains a virus, backdoors or a security-compromising program. P2P systems as eDonkey are vulnerable to Malwares since no detection system is integrated on their client software. Moreover, end-users are used to run the file as soon as it is completely downloaded without verifying its integrity. Often, viruses are split over several file blocks, and become active wherever the file is reassembled making them more difficult to detect.

In [Kalafut et al. 2006], authors investigated the existance of Malwares over two popular file sharing systems, Gnutella and FastTrack. Their test consisted of scanning a huge number of downloaded files using a free virus analyser over 45 days and obtained the following results:

– More than 68% of archived and compressed files were malicious on Gnutella. On FastTrack, the percentage was 3% for malicious files.

– 99% of malicious files detected on Gnutella contained only 3 different malicious codes. On FastTrack, 75% of malicious files were using three different codes.

These results show the extent of Malwares in P2P networks and the necessity to aware end users about the risk when using P2P applications.

Several attacks, such as backdoors, key loggers, password stealers, and other malwares functions represent the most visible and serious problems facing the Internet since its breakthrough. Anti-virus softwares (AV) are the core element of defense against these attacks and strain to detect, remove, and characterize these threats using viruses signature database [Bailey et al. 2007]. However, AV softwares still limited to counter all known malwares because of non consistent of these signatures. Intrusion Detection Systems (IDSs) are other kind of softwares that detect attacks caused by malwares. IDSs are software applications dedicated to detect intrusions against a target network, they are not intended to replace traditional security methods, but rather to complete them [Labib 2004]. P2P file sharing applications are an accelerator for malwares propagation due to the fact that end users are not security expert and often illegally download copyrighted files.

3.5 Free riding

The free-riding problem in P2P networks involves selfish nodes that obtain resources from other nodes in the network without themselves sharing any resources with other nodes [Konrath et al. 2007]. In the presence of such selfish nodes, not sharing resources becomes a dominant strategy among all nodes in the network and ultimately leads to a passive network without any resource exchange among nodes [Adar and Huberman 2000, Petrovic et al. 2007].

Each P2P application has a mechanism to prevent Free-riding in order to ensure the well functioning of the system and avoids unfairness between users. This mechanism depends on P2P protocols specifications. BitTorrent uses the *tit-for-tat* mechanism [Cohen 2008], meaning that a BitTorrent client will preferably cooperate with the peers cooperating with it. Practically, this means that each client measures how fast it can download from each peer and, in turn, will serve those from whom it has the better download rates. When a client has finished downloading a file, it no longer has to download from other Peers but it can still share (upload) pieces of the file. To encourage uploads, eMule employs a credit system. In other terms, when a client uploads files to his peer, the downloading client updates his credit according to the amount of data transferred [Kulbak and Kirkpatrick, 2005]. The credit system is not global, the credit for a transfer is kept locally by the downloading client and will be taken into account only when the uploading client (which earned the credit) will ask to download from this specific client [Kulbak and Kirkpatrick, 2005].

Due to their open nature, P2P file sharing applications are extremely exposed to Free-riding attacks. Attackers use modified P2P clients in order to bypass the described mechanisms. Several research works focused on Free-riding attacks over the BitTorrent network. In [Nielson et al. 2005], a taxonomy definition is given for rational attacks and illustrates how an attacker can maximize its utility, but does not harm the system unless it increases the benefits. Locher et al. [Locher et al. 2006] present BitThief, an agent which downloads but never contributes to the network. In [Sirivianos et al. 2007], authors introduced a free-riding attack which profits from having a wider view of the participating peers. This attack is validated by means of experiments in PlanetLab. In [Piatek et al. 2007], a modified version of a popular BitTorrent client (Azureus) is presented and is called BitTyrant. This malicious client adheres to the protocol but uses different policies to improve download performance while reducing upload contributions.

3.6 Poisoning file

A poisoning file [Christin et al. 2005] or a faked file is a file providing a content that does not match its description, it usually describes a popular movie or video clip. End users share poisoning files in order to be advantaged by the incentive mechanism in view of the fact that more peer A uploads to peer B, faster peer A will download from peer B. On the other side, A common technique to decrease the availability of a specific item (e.g., movie, song, software distribution) in a peer-to-peer network consists in injecting a massive number of decoys into the network in order to make users frustrated and then leave P2P file sharing applications [Liang and Kumar 2005]. This strategy is conducted by "pollution company" which works with major record labels, film studios, television networks, and game publisher to sabotage P2P systems [Liang and Kumar 2005]. The goal is to propagate the faked file over the P2P network and make it more available than clean.

Poisoning attack is a rational attack and does not disturb the network performance, almost it compels end users to request desired files from other sources and then generate additional Internet traffic. To limit this attack over eDonkey, eMule enables feature that allows previewing video making the client software requesting uppermost the first and the last blocks of the file. Previewing feature allow users to watch or listen to the first few minutes of the desired file and so, cancel the downloading if this last is faked.

4 Topology Change Attack Strategy

Topology change attack [Abdelouahab et al. 2008] attempts to redirect Internet traffic in order to congest or put out service one or several segments of the underlying infrastructure. This attack is analogous to traffic redirection attacks [Thing et al. 2005], where an attacker tries to launch a Denial of Service (DoS) [Long et al.] attack against Internet equipments. The DDoS attacks that shut down some high-profile Web sites (e.g. Yahoo, Amazon) in February 2000 [Garber 2000], demonstrated their severe consequences and the importance of efficient defense mechanisms.

In [Abdelouahab et al. 2008], we introduced the Topology Change Attack leaded against ISPs infrastructure gaining from eDonkey Network characteristics. This attacks consists of redirecting P2P traffic to a target ISP part. In TCA, an attacker needs to hijack an eDonkey server or to alter the *serverlist* file (see section 2) in order to eavesdrop eDonkey communication. A TCA could be lunched to redirect the signaling flow or the downloading folow using different methods:

- The attacker hijacks an eDonkey server, then disconnect it from the network. Clients which was connected to this server will connect to another server hosted on the target ISP. The attacker can also remove this server from the *serverlist* file on each connected client, then, these clients will connect to another eDonkey server.
- The attacker hijacks an eDonkey server, then request to client downloading replays with a malicious source list. The attacker forces a client in France to download a large content file from a client in Sydney, while the same information may be available at another client in France.

The Topology Change Attack may disturb the underling network by redirecting a part of the P2P communications to the target part of the network. Figure2 illustrates an eDonkey architecture deployed over two ISPs (ISP1, ISP2) and the overlay communications before and after the attack (1). For clarity concerns, we assume that each eDonkey client(eMule) is connected to an eDonkey server hosted in the same ISP. The attack consists of hiding server A from the P2P network (2), by removing it from the server list of clients A,B,C (see section2). Another attack, as a Worm or a backdoor trojan horse is needed to allow an attacker to remotly modify the server list file. Removing server A from the server

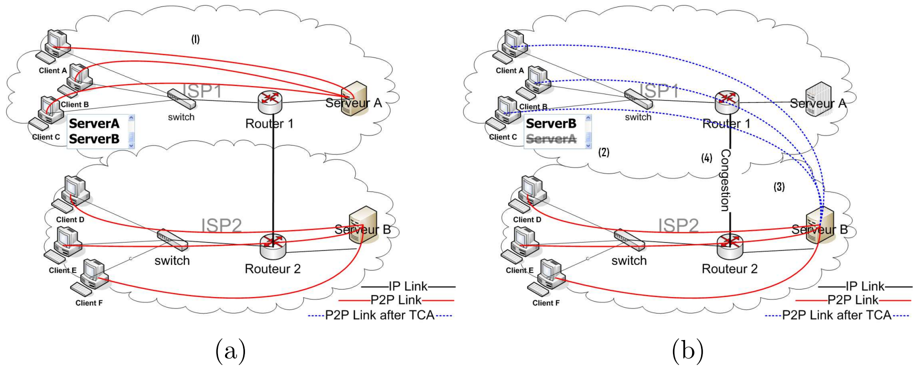


Figure 2: Topology change attack example

list enforces these clients to connect to an other eDonkey server (server B) existing in their list (3). These new connections (figure2.b) go over the link which is connecting router1 and router2 to ISP2, so overloading them with an additional payload (4). The more communications are diverted, the higher the impact of this attack is. The feasibility of a such attack is due to the large number of participating peers in the eDonkey network, and the weakness point that represents the server list file.

The topology change attack is an overlay attack and does not have the same approach of classical traffic redirecting attacks in launching a larger scale DoS. No spoofing IP addresses is needed to change the P2P overlay topology, hence, the existing solutions to prevent DDoS attacks [Snoeren 2001, Drew et al. 2001, IP source, Li et al. 2002]can not be used to prevent a TCA. Indeed, in DDoS attacks, the attack packets are often sent with spoofed IP addresses to hide the attackers' identity. Detecting and limiting TCA still an open research issue topic. A trivial solution should be inter-ISPs traffic analyzing to determine the number of eDonkey participants and the downloading flow generated by them. However the huge volume of traffic generated over Internet make this solution limited. In addition, P2P application are overlay protocols which can use other protocols as *FTP* to transfer data, so it is very difficult to determine whether this traffic result from P2P applications or not.

In table 1, we present the difference between TCA and other P2P attacks (see section 3) by classifying these attacks according to their impact. In comparison with the rest of existing attacks, TCA is more harmful from the point of view of ISPs and should have more research attention. P2P Worms and Malwares are designed to infect end-users machines in order to alter their normal functioning. Worms and Malwares concepters take profit from P2P characteristics to propagate their malicious programs in a faster and more clever manner

P2P attacks	Users machine	P2P network	Network resources
TCA		X	X
Worms	X	X	
Sybil		X	
Eclipse		X	
Free-riding		X	
Malwares	X	X	
Poisoning file	X	X	

Table 1: P2P attacks classification

than in client-server architectures. Sybil, Eclipse, Free-Riding and Poisoning files attacks have seen the light with the appearance of large scale P2P networking. These attacks intend to disturb P2P functioning and to decrease downloading performances. Furthermore, P2P file sharing application depends on end-users participating number, in this way a bad performance caused by one of theses attacks makes this application less popular, so less used.

In what follows, we addresse the different methods which allow to conduct a TCA On ISPs infrastructure and explain the different strategies that lead to alter the signal and the download flow, or both of them in order to harm ISPs network and equipments.

4.1 Signal flow redirection

This attack when conducted, consists in redirecting eDonkey signaling (client-server communication) to the target ISP. To lunch this attack, the attacker hijacks an eDonkey server, then disconnect it from the network. Clients which was connected to this server will connect to another server hosted on the target ISP. The attacker can also remove this server from the *serverlist* file on each connected client, then, these clients will connect to a server hosted on the target ISP. All or part of the signal flow is deviated towards the targeted ISP in order to unbalance the overlay topology. The impact of such attack is critical, especially, for ISP infrastructures. On eDonkey, client server TCP connections are kept open for the whole session, this fact worsens the impact. In order to a large succeed, this attack requires as a first step the success setting all servers hosted on the targeted ISP to high priority level and the rest to low priority. For a greater impact, the attacker should insert all servers not hosted on the targeted ISP in the *IP filter list* (see section 2).

Figure 3 shows a typical example of redirecting signal flow. It illustrates three ISPs (ISP1, ISP2, ISP3) interconnected over the Internet. Each ISP hosts

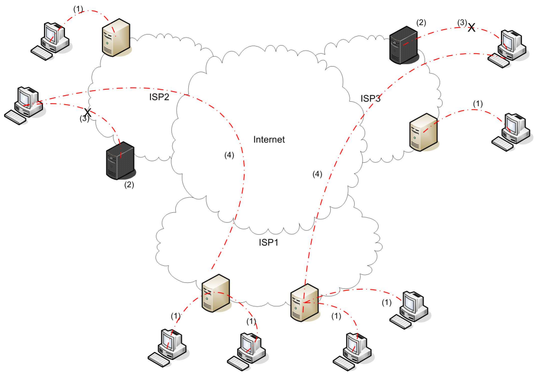


Figure 3: Example of signalling flow Topology Change Attack

eDonkey clients and servers. The signalling flow is represented by a discontinuous red line. For clarity's sake, we assume that the two parts of a client-server communication are hosted on the same ISP. In fact this is very close to reality.

1) In this example, two client-server communications are considered.2) The attack makes non-ISP1 servers (black) invisible to their clients. 3) The attack breaks these flows down by inserting, into the list-filter, the IP address of a server which does not belong to ISP1. 4) These clients will initiate new connections to ISP1 servers. When this attack is launched, a huge number of clients will reinitiate connections to servers hosted by ISP1. Client-server communications will then overload the ISP1 infrastructure. Therefore, this attack is considered as a *Topology Change Attack*.

4.2 Download flow redirection

This attack consists of redirecting eDonkey client to client communication. The attacker hijacks an eDonkey server, then replays to client downloading requests with a malicious source list. All sources of the malicious list must be hosted on the targeted ISP, the attacker forces a connecting eDonkey clients to download a large content file cross their ISP, while the same information may be available on their ISP.

In figure 4, we illustrate with an example this case using the same network scheme as in figure 3. The signalling flow is represented by a discontinuous red line, and the download flow by a continuous blue line. 1) We assume that the two parts of a client-server communication are hosted by the same ISP. 2) The first step for an attacker is to hijack eDonkey servers and make them malicious. Malicious servers act as legitimate clients for the ISP1. 3) When receiving a search request for a file F, a malicious server forwards it to an ISP1 server. The

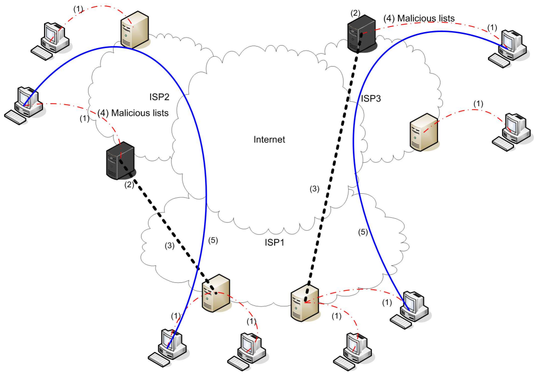


Figure 4: Example of download flow Topology Change Attack

ISP1 server will then reply with a list of ISP1 sources sharing the file F. Upon receiving this list, 4) the malicious server forwards it to the client that initiated the research request. Recognising only ISP1 sources, 5) this client will download the F blocks from ISP1 clients only. Successful initiation of this attack results in a dramatic impact on ISP1.

4.3 Signal and download flow redirection

This attack is the combination of both of the previous cases. It has a higher impact compared with a single attack, because both client-client and client-server communications are redirected toward the targeted ISP.

In figure 5, we illustrate this attack case using the same scheme as figure 3. 1) The attack needs to set ISP1 servers only to visible and 2) make the rest invisible. The non-ISP1 servers may be rendered invisible by inserting them into the list-filter. The greater the number of modified client lists, the more dangerous is the impact. 3) Recognising only ISP1 servers, a client can make a reconnection only to one of these. Once this step is accomplished, all the signalling flow over the eDonkey network will transit by the ISP1 infrastructure, thus overloading it. Furthermore, if the ISP1 servers are hijacked, they may reply to research requests with only source lists hosted on ISP1. 4) So, at least one of the two clients in a download communication is hosted by ISP1. Now, both signalling and data flows transit through the ISP1 infrastructure. Since it is receiving all the eDonkey flow, the ISP infrastructure may be badly overloaded and congested.

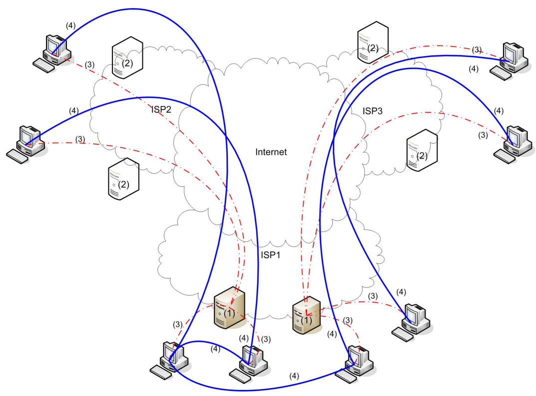


Figure 5: Example of signal and download flow Topology Change Attack

5 Simulation Model and Results

In this section, we present our model that we used to simulate the eDonkey protocol and the Topology Change Attack. This model is based on modular architecture based on PeerSim [Jelasity et al.] simulator. PeerSim is an extendable P2P simulator. In fact, the simulator is a core over which can be connected other units depending on simulation objectives. The configuration is done using a simple text file. PeerSim can be used in two modes, the cycle-based and the event-based. The second one is less used in literature and bad documented by PeerSim developers. In the cycle-based mode, the simulator runs the protocol specified in every node and go to the next cycle. Controls can be executed every cycle to supervise the simulation. Further, we discuss and motivate our assumptions related to this model. We believe that PeerSim is the more appropriate simulator to implement the eDonkey protocol for the following reasons:

- PeerSim allows a good scalability (one million), while other simulators are less scalable.

- The modular architecture is useful to extend and implement the eDonkey protocol;

We done our simulations by adding several modules to the core of PeerSim. The final simulator is composed of three major parts: Configuration file, simulation modules and real time monitoring. The configuration file, which already exists in PeerSim, contains several simulation parameters, the most important are: Number of nodes, number of simulation cycles, and the number of simulated ISPs. These parameters are used by simulator modules and gives as result statistics about cross ISPs generated flow. Therefore, the monitoring part uses collected statistics to draw eDonkey flows graphs on each cycle. These graphs will be helpful when simulating attacks on eDonkey network, by comparing comportment before and after the attack.

5.1 Traffic vs file

The proposed eDonkey simulation model is based on peers behaviour. More this the simulated behaviour is close to the real behaviour of eDonkey clients, more the obtained results are close to reality. Unlike emulation, the simulation do not keep in consideration all existing mechanisms on an eDonkey client. To evaluate the impact of the Topology change Attack, simulating the flow generated by clients on the eDonkey network is more appropriate than simulating other mechanisms like I/O to the hard disk. The eDonkey flow can be generated using two techniques: the first one is file transfer simulation and second one is to simulate the traffic generated by peers. Our first assumption is the simulation of eDonkey traffic instead files simulation. Thus, allow to omit in consideration the queuing system of file blocks on each node. The realism will be the same whereas we based our traffic simulation on existing network analysis. In [Tutschku 2004], is presented a traffic profile of the eDonkey Network. The measurement-based study revealed a strong distinction between download flows and non-download stream. eDonkey participants are over one million around the network. So, simulate a queuing system for each node, makes material consumption non-realistic.

5.2 Population repartition

In [Tutschku 2004], Kurt Tutscku studied exchanged data flows in the eDonkey network. The collected statistical values using tcpdump were used to determine the average downloading and uploading stream and client position over the eDonkey Network. We used these statistics to simulate a realistic client repartition over our sub-networks. For example, if considering a group of clients in ISP1, 45% of the downloaded or uploaded flow related to this group is in ISP1. In our model, the client repartition on the eDonkey network follows a discrete normal repartition. In order to introduce this assumption

5.3 Client behaviour

Three approaches can be used in P2P simulation: multiple peer characteristics models, the content or the shared resources models, the individual peer behaviour model. In our simulations we used the individual peer behaviour approach and collect statistics related to generated flow which is detailed in the next section.

Each client, is initialized with two reference: the eDonkey server and the ISP to which it is connected. The sequence to download a file is as follows:

- The client asks the server to obtain the client address from where he can download the desired file.
- The server generates randomly an ISP from the client can download the desired file and the number of block which contain this file.
- Every cycle, this client downloads one block.
- The client saves the generated flow size to make statistics in the end of the simulation.

We assume that downloaded files are present over all sub-networks (all ISPs). In eDonkey, downloading file can be done from multiple sources, whereas our simulator generates a realistic overlay random traffic and do not take in account files transfer.

5.4 Simulated network topology

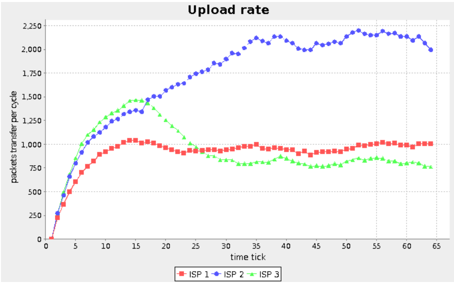
Our simulator generates different topologies depending on the number of ISPs to be simulated. This parameter can be modified using the configuration file. All ISPs are interconnected to Internet and could be neighbors or not and each ISP contain eDonkey Clients and Servers. We assume, if a node A is hosted on ISP1 then it connects to a server hosted in the same ISP. This assumption means that all eDonkey Client-server communications are in the same ISP before the attack.

5.5 Simulation results

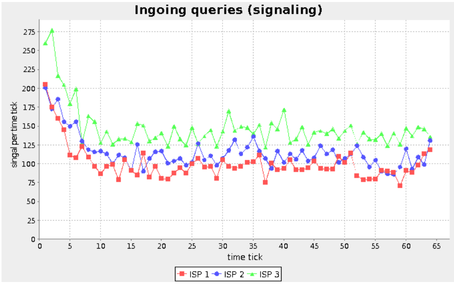
In order to evaluate the impact of the topology change attack, we simulated the eDOnkey network and compare its behavior before and after lunching this attack. We configured three interconnected ISPs and each one of them hosts the same number of eDonkey clients and servers.

Figure6 displays the uploading rate flow and the signaling flow generated over three ISPs during 65 cycles. Uploading flow are quantified by the number of packet send over each simulation cycle. We set up the fifteenth cycle as the attacking step. The attack consists of a TCA attack which redirect intra-ISP3 communications over ISP2. This attack is done by altering the server-list file.

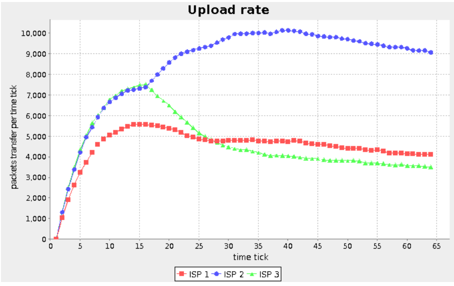
We note that before cycle 15, all graphs evolve similarly, this fact is due to the symmetry of client repartition. Until the attacking cycle, the downloading flow growth rapidly and then the overlay network is considered as stable and the downloading and signaling flows reach a steady step. This state is caused by client server connections. Before starting to download from the eDonkey network, each client has to connect to a server which can not accept all connecting request simultaneously. We observe a considerable growth of the uploading rate over



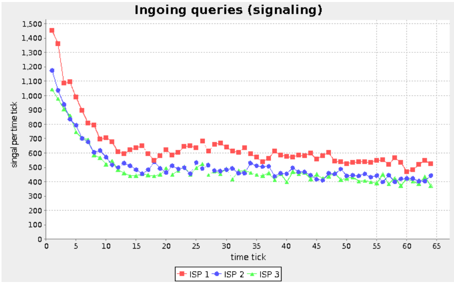
(a) 10.000 peers



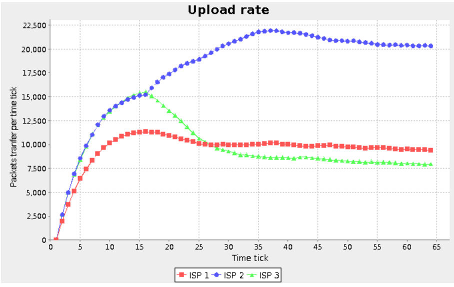
(b) 10.000 peers



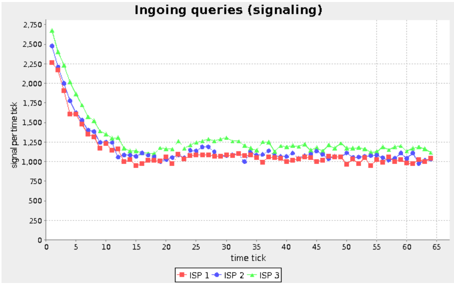
(c) 50.000 peers



(d) 50.000 peers



(e)100.000 peers

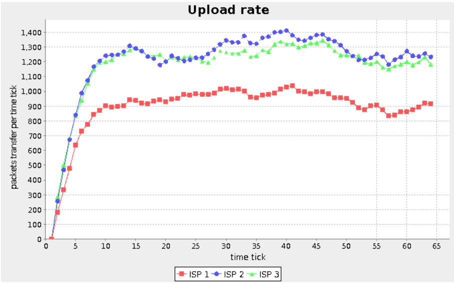


(f) 100.000 peers

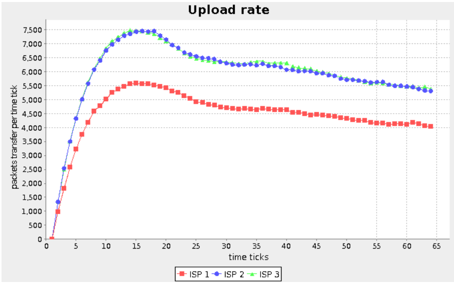
Figure 6: Simulation under scenario attack 1

ISP2 and a sudden decreasing over ISP3. Our simulations results show that, more peers are participating to the overlay network, more higher is the impact.

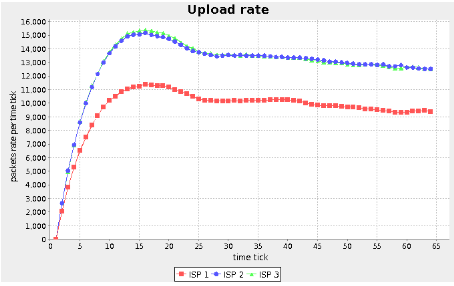
Figure7 displays the uploading flow for each ISP under the assumptions as in figure6. We lunched a second TCA scenario, when both of ISP2 and ISP3 communications are under attack by redirecting ISP1 traffic. Results show that TCA has more impact when an attacker chose to target just one ISP. This attack can be done with out taking in consideration the participating peers number, however the higher this number is, the worst the impact is. Simulation results validate our expectancy and demonstrate the real threat against ISPs



(a) 10.000 peers



(b) 50.000 peers



(c) 100.000 peers

Figure 7: Simulation under scenario attack 2

infrastructure when a topology change attack is deployed.

6 Conclusions and Future Work

We have presented the Topology Change Attack cases that can be lunched taking profit from the eDonkey application in order to unbalance ISPs infrastructures. The application studied is eDonkey, because of its popularity and large use. Due to P2P simulators state, we developed our own simulator in order to evaluate TCA imapct. By taking into account eDonkey architecture and functioning, we extended *PeerSim* to simulate the eDonkey network. Results indicate that eDonkey and ISPs infrastructure are vulnerable to such attack and demonstrate that *server-list* file represents a weakness point.

Surveyed attacks are well known in client-server architectures, however they became more dangerous gaining from P2P in the point of view propagation rate and participants large scale. In this paper, we focused on ISPs' infrastructure impact causing by TCA. In order to compare this impact with other P2P attacks impact, we are planing to describe and model a new P2P Worm which generates an additional and non desired Internet traffic. This Worm forces two infected machines to communicate, although any request is done by their end-users. Furthermore, TCA attack techniques depends on protocol specifications, we are studying strategies to deploy it and potential impact over other popular file sharing applications as BitTorrent.

References

[Abdelouahab et al. 2008] Abdelouahab, M., Ragab Hassen, H., Bouabdallah, A., Achemlal, M., Laniepce, S.: Tca: Topology change attack in peer-to-peer networks. In Mobile and Wireless Networks Security workshop (MWNS), March 2008.

[Adar and Huberman 2000] Adar, E., Huberman, B. A.: Free riding on Gnutella. First Monday, 5(10), October 2000.

[Aggarwal et al. 2007] Aggarwal, V., Feldmann, A., Scheideler, C.: Can isps and p2p users cooperate for improved performance? SIGCOMM Comput. Commun. Rev.,2007.

[Bailey et al. 2007] Bailey, M., Andersen, J., Morleymao, Z., Jahanian, F.: Automated classification and analysis of internet malware. Technical report, In Proceedings of Recent Advances in Intrusion Detection, 2007.

[Chellappan et al. 2005] Chellappan, S., Yu, W., Boyer, C., Xuan, D.: Peer-to-peer system-based active worm attacks: Modeling and analysis. In Proceedings of IEEE International Conference on Communications (ICC'05), 2005.

[Chen et al. 2003] Chen, Z.: Modeling the spread of active worms. 2003.

[Christin et al. 2005] Christin, N., Weigend, A. S., Chuang, J.: Content availability, pollution and poisoning in File sharing peer-to-peer networks. In 6th ACM conference on Electronic commerce (EC '05), pages 68-77, New York, NY, USA, 2005. ACM Press.

[Cohen 2008] Cohen, B.: Bep 3: The bittorrent protocol specification, February 2008.

[Dabek et al. 2001] Dabek, F., Kaashoek, F. M., Karger, D., Morris, R., Stoica, I.: Wide-area cooperative storage with cfs. In SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles, volume 35, pages 202-215, New York, NY, USA, December 2001. ACM Press.

[Douceur 2002] Douceur, J. R.: The sybil attack. In IPTPS'01: Revised Papers from the First International Workshop on Peer-to-Peer Systems, pages 251-260, London, UK, 2002. Springer-Verlag.

[Drew et al. 2001] Dean, D., Franklin, M., Stubblefield, A.: An algebraic approach to ip traceback. In ACM Transactions on Information and System Security, pages 3-12, 2001.

[Garber 2000] Garber, L.: Denial-of-service attacks rip the internet. Computer, 2000.

[Hatahete et al. 2008] Sinan, H., Yacine, C., Abdelmadjid, B.: A new worm prop- agation threat in bittorrent: Modeling and analysis. pages 791-798, Univerisity of Technology of Compiegne, 2008. Computer Science and Information Technology, 2008. IMCSIT 2008.

[ICANN 2004] ICANN. The internet corporation for assigned names and numbers, 2004.

[IP source] Ip Source and Address Spoofing. Status of this memo network ingress filtering: Defeating denial of service attacks which employ.

[Jelasity et al.] Jelasity, M., Montresor, A., Jesi, G. P., Voulgaris, S.: The Peersim simulator. http://peersim.sf.net.

[Joukov et al. 2007] Joukov, N., Chiueh, T. C.: Internet worms as internet-wide threat. Com- puter Security Institute, 2007.

[Kalafut et al. 2006] Kalafut, A., Acharya, A., Gupta, M.: A study of malware in peer-to-peer networks. In IMC'06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, pages 327-332, New York, NY, USA, 2006.

[Konrath et al. 2007] Konrath, M. A., Barcellos, M. P., Mansilha, R. B.: Attacking a swarm with a band of liars: evaluating the impact of attacks on bittorrent. In The Seventh IEEE International Conference on Peer-to-Peer Computing. IEEE, September 2007.

[Kulbak and Kirkpatrick, 2005] Kulbak, Y., Kirkpatrick, S.: The emule protocol specification. Technical report, 2005.

[Labib 2004] Labib, K.: Computer security and intrusion detection. Crossroads, 2004.

[Lefebvre 2000] Lefebvre, K. R.: The Added Value of EMBASSY. Wave Systems Corp. white paper, in the digital world www.wave.com edition, 2000.

[Lesniewski 2008] Lesniewski-Laas, C.: A sybil-proof one-hop dht. In SocialNets '08: Proceedings of the 1st workshop on Social network systems, pages 19-24, New York, NY, USA, 2008.

[Liang and Kumar 2005] Liang, J., Kumar, R.: Pollution in p2p File sharing systems. In In IEEE INFOCOM, pages 11741185, 2005.

[Li et al. 2002] Li, J., Mirkovic, J., Wang, M., Reiher, P., Zhang, L.: Save: Source address validity enforcement protocol. In In Proceedings of IEEE INFO- COM 2002, pages 1557-1566, 2002.

[Locher et al. 2006] Locher, T., Moor, P., Schmid, S., Wattenhofer, R.: Free Rid- ing in BitTorrent is Cheap. In 5th Workshop on Hot Topics in Networks (HotNets), Irvine, California, USA, November 2006.

[Long et al.] Long, N., Houle, K. J., Weaver, G. M., Thomas, R.: Trends in denial of service attack technology.

[Mazieres et al. 1999] Mazieres, D., Kaminsky, M., Kaashoek, M. F., Witchel, E.: Separating key management from file system security. SIGOPS Oper. Syst. Rev., 124-139, 1999.

[Nassima 2002] Nassima, K., Yannick, C., Nazim, A.: How to own the internet in your spare time. In Proceedings of the 11th USENIX Security Symposium, 2002.

[Nakao 2005] Nakao, A.: A routing underlay for overlay networks. PhD thesis, Princeton, NJ, USA, 2005. Adviser-Peterson,, Larry.

[Nielson et al. 2005] Nielson, S., Crosby, S., Wallach, D.: A taxonomy of rational attacks. In The 4th Annual International Workshop on Peer-To-Peer Systems (IPTPS 2005). Springer Berlin / Heidelberg, February 2005.

[Petrovic et al. 2007] Petrovic, S., Brown, P., Costeux, J. L.: Unfairness in the e- Mule File Sharing System, volume 4516-2007. Computer Science, 2007.

[Piatek et al. 2007] Piatek, M., Isdal, T., Anderson, T., Krishnamurthy, A., Venkataramani, A.: Do incentives build robustness in bittorrent? In Proceedings of 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2007), Cambridge, MA, April 2007. USENIX.

[Richardson et al. 2007] Richardson, R.: Computer crime and security survey. Computer Security Institute, 2007.

[Singh et al. 2006] Singh, A., Ngan, T. W., Druschel, P., Wallach, D. S.: Eclipse attacks on overlay networks: Threats and defenses. In INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pages 1-12, 2006.

[Singh et al.] Singh, A., Castro, M., Druschel, P., Rowstron, A.: Abstract de- fending against eclipse attacks on overlay networks.

[Sirivianos et al. 2007] Sirivianos, M., Park, J. H., Chen, R., Yang, X.: Free-riding in bittorrent with the large view exploit. In 6th International Workshop on Peer-to-Peer Systems (IPTPS 2007), Bellevue, WA, US, February 2007.

[Snoeren 2001] Snoeren, A. C.: Hash-based ip traceback. In SIGCOMM'01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pages 3-14, New York, NY, USA, 2001. ACM.

[Staniford et al. 2002] Staniford, S., Paxson, V., Weaver, L.: How to own the internet in your spare time. In Proceedings of the 11th USENIX Security Symposium, pages 149-167, Berkeley, CA, USA, 2002. USENIX Association.

[Thing et al. 2005] Thing, V., Lee, H., Sloman, M.: Traffic redirection attack protection system (TRAPS). In 20th IFIP International Information Security Conference (SEC), Makuhari-Messe, Chiba, Japan,, pages 309-325, April 2005. Best Student Paper Award.

[Tutschku 2004] Tutschku, K.: A measurement-based traffic profile of the edonkey filesharing service. pages 12-21. 2004.

[Verisign] Verisign Inc. www.verisign.com.

[Zhou et al. 2005] Zhou, L., Zhang, L., Mcsherry, F., Immorlica, N., Costa, M., Chien, S.: A First look at peer-to-peer worms: Threats and defenses. In In Proceedings of the IPTPS, 2005.

[Zimmerman 2006] Zimmerman, P.: PGP user's guide. MIT, 1994.