# How Many Eyes are Spying on Your Shared Folders?

Bingshuang Liu[12], Zhaoyang Liu[13], Jianyu Zhang[12][*], Tao Wei[124] and Wei Zou[12]

Beijing Key Laboratory of Internet Security Technology, Peking University[1]
Institute of Computer Science and Technology, Peking University[2]
Beijing 100871, China
{liubingshuang, zhangjianyu, zou_wei}@pku.edu.cn
Beijing University of Posts and Telecommunications[3], Beijing 100876, China
liuzy.bupt@gmail.com
EECS, UC Berkeley[4], Berkeley, CA 94720, USA
wei_tao@pku.edu.cn

## ABSTRACT

Today peer-to-peer (P2P) file sharing networks help tens of millions of users to share contents on the Internet. However, users' private files in their shared folders might become accessible to everybody inadvertently. In this paper, we investigate this kind of user privacy exposures in Kad, one of the biggest P2P file sharing networks, and try to answer two questions: *Q1.* Whether and to what extent does this problem exist in current systems? *Q2.* Are attackers aware of this privacy vulnerability and are they abusing obtained private information?

We build a monitoring system called Dragonfly based on the eclipse mechanism to passively monitor sharing and downloading events in Kad. We also use the Honeyfile approach to share forged private information to observe attackers' behaviors. Based on Dragonfly and Honeyfiles, we give affirmative answers to the above two questions. Within two weeks, more than five thousand private files related to ten sensitive keywords were shared by Kad users, and over half of them come from Italy and Spain. Within one month, each honey file was downloaded for about 40 times in average, and its inner password information was exploited for 25 times. These results show that this privacy problem has become a serious threat for P2P users. Finally, we design and implement Numen, a plug-in for eMule, which can effectively protect user private files from being shared without notice.

## Categories and Subject Descriptors

D.4.6 [**Software**]: OPERATING SYSTEMS—*Security and protection*; C.2.4 [**Computer Systems Organization**]: Computer-Communication Networks[Distributed Systems]

## General Terms

Measurement, Security

## Keywords

P2P, File-sharing, Privacy, Kad

## 1. INTRODUCTION

Since Napster was founded in 1999, file sharing through peer to peer (P2P) has been turned out to be practical and popular. At present, the concurrent user population reaches tens of millions [27] [24]. According to 2009 Internet Study provided by ipoque [5], the traffic volume consumed by P2P file sharing has already exceeded the sum of all other applications over the Internet, up to 51.6% of total Internet traffic. P2P file sharing has become an important application on the Internet.

However, the ease-of-use and high performance bring not only the great success of P2P file sharing, but also a big concern for user privacy exposures.

Most previous studies [23] [8] [16] focused on downloading and sharing behavior privacy, i.e., who shares or downloads some contents, e.g., Motion Picture Association of America (MPAA) is eager to know which users of some ISPs have downloaded pirated versions of movies using P2P. We call this kind of user privacy "location privacy". To protect users' location privacy, many anonymous communication technologies are developed, such as Tor [14], DC-Nets [10] and Mixes [9]. Despite some performance loss, the location privacy can be protected.

In this paper, we focus on another kind of user privacy in P2P file sharing systems, which can be called "content privacy". Usually, P2P systems need every participator to dedicate his bandwidth, storage and contents. And almost all of these encourage users to share their contents as many as possible. The operational cost of sharing a file, a folder, and even the whole hard drive is very low, which maybe needs only a simple mouse click. As a consequence, it brings potential privacy risks: people probably make their private and confidential files accessible to everybody in the P2P networks, inadvertently and unknowingly.

Content privacy exposures most likely happen under such a scenario: in a single machine, multiple user environment. Imagine that in a family, all members share one computer.

[*] Corresponding author.

*WPES'12,* October 15, 2012, Raleigh, North Carolina, USA.
Copyright 2012 ACM 978-1-4503-1663-7/12/10 ...$15.00.

The father processed some confidential files by a secure SSH connection to his company and stored them in the folder "My Documents"; His teenage son wants to download a game program, so he sets up a P2P file sharing client and casually selects "My Documents" as the default shared folder. Thus his father's confidential files have been inadvertently shared, without either party's knowledge.

In this work, we mainly investigate this problem in the Kad network. Kad is one of the most popular DHT (Distributed Hash Table) networks [20], implemented by eMule [2] and aMule [1], and holds millions of concurrent users. Based on Kad, we try to answer the following two questions:

*Question 1 (Q1)*: whether and to what extent the content privacy exposures exist in current systems;

*Question 2 (Q2)*: whether someone (may be a hacker or a curious user, who is referred as "potential attacker") has observed such privacy exposures of benign users, and further abused others' private information obtained from file sharing networks to carry out some attacks like identity theft.

To monitor sharing and downloading events in Kad, we build the monitoring system called Dragonfly based on the eclipse mechanism [18]. By using Dragonfly, we found that within two weeks more than five thousand private files related to ten predetermined keywords were shared by Kad users, over half of whom come from Italy and Spain. Further, by deploying five honey files [28], we detected that 192 distinct potential attackers tried to download these honey files through Kad within one month. And at least 45 attackers further abuse honey accounts to carry out identity-theft attacks for 125 times. In average, each honey file was downloaded for about 40 times, and its inner password information was exploited for 25 times. These results show that this privacy problem has become a serious threat for P2P users. To the best of our knowledge, this is the first systematic study to evaluate the problem of content privacy exposures in P2P file sharing networks.

Finally, we implement Numen, a plug-in for eMule. By utilizing the uniqueness of private files, it can effectively protect user private files from being shared without user knowledge.

Though this paper mainly focuses on Kad, we believe that this privacy problem exists in other P2P file sharing networks and our study can be applied to them with modest modifications.

The rest of this paper is organized as follows. Section 2 describes the prerequisite knowledge of Kad. Section 3 presents the monitoring system, Dragonfly. Section 4 analyzes content privacy exposures in Kad. Section 5 evaluates the exploitation of this privacy vulnerability. Section 6 presents our solution, Numen, to mitigate the problem. Section 7 discusses related work on user privacy in file sharing networks, and Section 8 concludes the paper.

## 2. OVERVIEW OF KAD

In Kad, every participating node has a unique identifier, referred as *node ID*. The node ID is a 128-bit random number. Kad has two kinds of objects, keyword and file. A 128-bit identifier called *key ID* is assigned to every keyword object, and a 128-bit identifier called *file ID* to every file object. Key ID is got from the hash of *keyword* and file ID is the hash of *file* content. Ideally, every ID is globally unique.

Kad applies a 2-level index structure shown in Figure 2, i.e., keyword-to-file index and file-to-source index. Given a specific keyword, the former gives a certain number of file information in Kad, whose tags contain this keyword; and the latter tells which users (sources) own the file and you can download it from them. Based on the index structure, Kad supports two DHT primary operations: *Publish* and *Search*. *Publish* is the process of storing level-1 or level-2 indexes on other nodes when sharing a file, and *Search* is the reverse process while downloading the file. When a user wants to download the movie "Titanic.avi", for example, the process is as follows: firstly the user launches a query by keyword "Titanic" and retrieves several candidate video files by level-1 indexes; secondly he selects one file to query who could provide it through the level-2 indexes; finally, after getting some sources, the user begins to download the file directly from them. On the other hand, sharing a file is the reverse of downloading process, where the file owner publishes 2-level indexes to corresponding nodes.

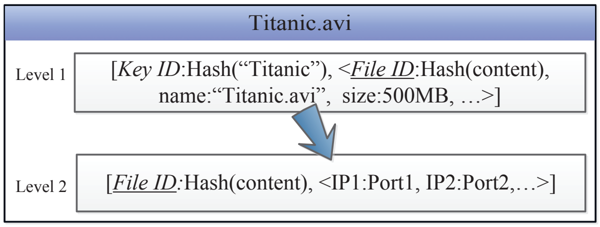
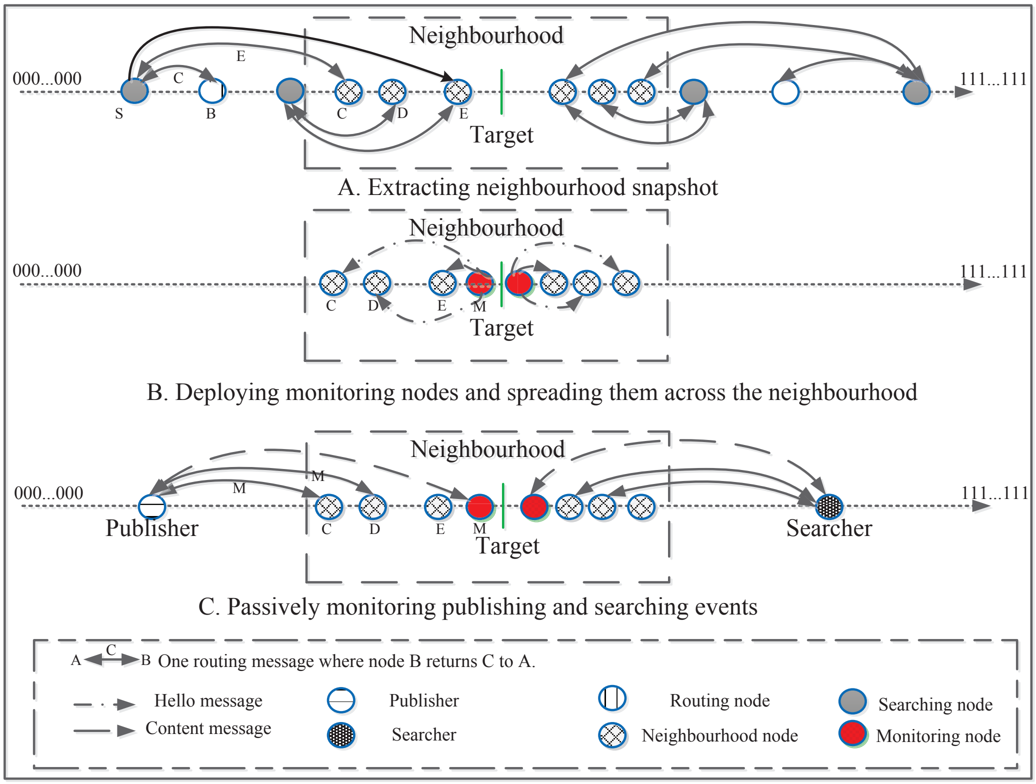

Titanic.avi
Level 1
[*Key ID*:Hash("Titanic"), <*File ID*:Hash(content), name:"Titanic.avi", size:500MB, …>]
Level 2
[*File ID*:Hash(content), <IP1:Port1, IP2:Port2,…>]

**Figure 2: The 2-level index structure of Kad.**

In the 128-bit ID space, the distance of two IDs is calculated using bitwise $XOR$ operation. Based on this definition, Kad maps every object to those nodes that are closest to its ID, and every node takes charge of a set of adjacent objects. Both *Publish* and *Search* need a routing process which is in charge of locating nodes which are enough close to the target. The routing algorithm used in Kad is iterative and greedy. During routing process, the intermediate nodes return some closer nodes to the initiator; and then the initiator picks up several appropriate nodes among them as next hops, and repeats the process until no closer nodes are learned. Only when the routing process stops, the real content operations (*Publish* and *Search*) can be carried out on these candidates.

Here, Kad is selected as monitored target, based on the following considerations:

- Kad has a huge user base, nearly four million [24], so the impact of the privacy problem is significant;
- All files in the default shared folders or download folders are acquiescently shared by eMule, the most popular Kad client, even if files are not downloaded from Kad;
- In eMule, only one click is needed to share one folder, including all its recursive subfolders. Even when the whole hard disk is selected by mistake, there is no any privacy warning raised by the client;
- The sharing file list of a Kad user is public for anyone connecting to him, so it might amplify user privacy risk.



A. Extracting neighbourhood snapshot

B. Deploying monitoring nodes and spreading them across the neighbourhood

C. Passively monitoring publishing and searching events

**Figure 1: The workflow diagram of Dragonfly**

From above Kad's features, we infer that the privacy problem in Kad is more serious than that in other networks. So Kad is better to be used to reveal the privacy problem.

## 3. MONITORING SYSTEM: DRAGONFLY

In order to monitor publishing (sharing) and searching (downloading) events for specific keywords or files in Kad, We build a monitoring system called Dragonfly, which combines eclipse mechanism [18] with passive measurement technique [21]. The former generates enough nodes surrounding the target and captures all incoming requests; and the latter sets up a number of nodes to listen Kad traffic passively on a large scale. For a keyword target, Dragonfly reports information of files related to the keyword in Kad; and for a file target, Dragonfly reports who are sharing or downloading the file.

We can also search keywords directly using the 2-level indexes in Kad (active measurement). However, there are some limitations making this method unpractical. First, there is a maximum threshold (300) for the count of Kad search results. Further, according to previous work [19] [17], Kad lookup is incomplete due to its lookup mechanism and some external factors. Besides, [26] pointed that nearly 35.1% of Kad nodes may be temporarily unreachable due to the maintenance scheme of routing table, firewall or NAT devices. Finally, directly searching cannot get the information of downloaders. So searching directly is not a good choice.

The work process of Dragonfly is illustrated in Figure 1 and can be described as follows:

A. In the 128-bit ID space of Kad, 32 searching nodes are evenly deployed and continuously search the given target ID. Periodically, this procedure exports a current neighborhood snapshot which contains nodes adjacent to the target, i.e., more than 18 common prefix bits with the target ID;

B. We set up 40 monitoring nodes which have at least 28 common prefix bits (enough close) with the target ID. Then with help from Step A, these nodes continuously send hello messages to neighborhood nodes in order to permeate into their routing tables;

C. All monitoring nodes "eclipse" the monitored target and passively wait for others to connect. When a user launches publish or search operations aimed at the target, the routing process would be lead to these monitoring nodes in the end. Then these nodes could passively record subsequent publishing or searching events.

During monitoring process, searching nodes and monitoring nodes provide normal Kad functions for others, such as routing, storing and providing indexes. Thus our monitoring system has negligible impact on Kad.

We implement Dragonfly based on the aMule client (a popular open-source and cross-platform project, the latest stable version is 2.3.1) to meet demands of searching and monitoring nodes. All these nodes are set up on a PC server with two Intel Xeon CPUs (E5645, 2.40GHz, 24GB RAM). 150 IP links are rented from ISP and can be multiplexed by these nodes. The bandwidth of each IP link is 2Mbps.

We use the following experiment to verify the effectiveness of Dragonfly. In each experiment, we emit 1000 publishing events and searching events respectively by normal Kad clients for a monitored target. Then we count how many events issued by our clients can be caught by Dragonfly. We have conducted 500 experiments and the results show that the average capture ratio is about 87%, i.e., 87 out of 100 events can be listened by Dragonfly. We believe that this ratio is good enough to evaluate the privacy problem in Kad.

# 4. CONTENT PRIVACY EXPOSURES

In this section, we will try to answer **Q1**: **Whether and to what extent content privacy exposures exist in current systems.** Making use of our monitoring system "Dragonfly", we continuously monitor ten privacy-related keywords in Kad for two weeks in June 2012. Dragonfly has monitored tens of thousands of publishing events from all over the world, which indicates lots of private and confidential files are being inadvertently shared by careless users.

## 4.1 Methodology

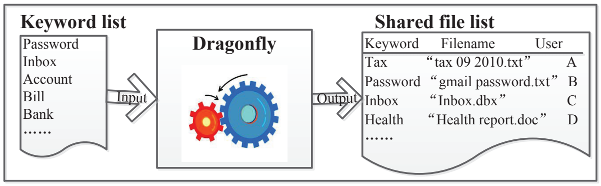


**Figure 3: The process of monitoring keyword in Kad through Dragonfly.**

The process of monitoring privacy-related keywords is illustrated in Figure 3, which can be summarized as follows. First, a privacy-related keyword list (all in English) is constructed, which involves several typical user privacy, such as tax records, health records, Web accounts and passwords, contacts and company contracts; Then, this list is inputted into Dragonfly and the monitoring system can continuously monitor corresponding file sharing events in Kad; Through $< keyword, file, user >$ triples in output file, we get the file information (filename, filesize, filetype, fileID, etc) and the file owner information (Kad ID, IP, Port, etc). With the help of such information, the situation of user privacy exposures in Kad could be deduced.

To protect users' privacy, we didn't directly download suspected files to verify whether they are private files or not. Generally speaking, the filename can exhibit file's content to a certain extent. If a file is named "my Gmail password.txt", it certainly relates to user privacy. Therefore, the filename can naturally be used to judge the property of one file. Besides, the uniqueness in the network, as one of the most important attributes of private files, can help us identify them.

We carried out experiments for two weeks using Dragonfly, from 13th to 26th in June, 2012. The result will be presented and analyzed in the next section.

## 4.2 Experimental Result

Dragonfly has got more than 220K suspected files from 530K Kad users within two weeks. Table 1 presents results in detail, for every privacy-related keyword. It gives a clear answer to **Q1**: this kind of privacy problems exists widely in Kad and a lot of user privacy is publicly exposed.

**Table 1: Results of monitoring privacy-related keywords in Kad from 13th to 26th in June, 2012.**

| **Keyword** | **#users** | **#files** | **#private_files** |
|---|---|---|---|
| password | 85,051 | 43,355 | 3,446 |
| mail | 62,148 | 16,804 | 664 |
| account | 11,718 | 3,683 | 398 |
| bill | 190,803 | 96,990 | 277 |
| bank | 59,332 | 15,166 | 172 |
| health | 70,733 | 34,153 | 151 |
| contract | 29,595 | 6,169 | 136 |
| contacts | 6,621 | 2,977 | 87 |
| tax | 13,842 | 3,145 | 43 |
| inbox | 474 | 515 | 7 |
| **SUM** | 530,317 | 222,957 | 5,381 |

There are too many files related to one keyword, in particular, nearly 100K files for keyword "bill". By carefully inspecting file information, we found there are lots of non-privacy files. The most obvious ones are plenty of multimedia files, e.g., movie and music. Based on several heuristic rules, we filter out these irrelevant files. The process is depicted as follows:

1. Document files are more likely to contain user privacy for these keywords selected in this work. Non-document files, such as multimedia and program, are firstly filtered out;

2. For a normal user, it is impossible that numerous private files related to one keyword are shared at the same time. Therefore if a user publishes multiple files related to the same keyword at one publishing event, all files are excluded;

3. Private files should be unique. If a suspected file is shared by more than one user in Kad, we think it is not private.

These rules are used to filter irrelevant files, but false negatives are inevitable. Some private files, e.g., multiple password files in different shared folders of a Kad user, would be wrongly filtered out. However, we believe that these false negative rate is very low, because we found most Kad victims share only one private file during the monitoring process.

After the filtering process, we analyzed and verified left files manually. In the end, more than five thousands files are identified related to user privacy, as shown in column "#private_files". The result shows that there are still many private files for every keyword, more than 500 in average. In particular, the number of private files involving web accounts is obviously more than others, referring to the top 3 keywords "password", "mail" and "account". It means that some users are being faced with big risks of identity theft on the Internet when using P2P file sharing networks.

After getting the basic answer to the Q1, we further investigate how many days a private file would be shared by the victim during our monitoring window of two weeks. There are two ways for the private files to be removed from Kad. The first one is that the owner realizes the privacy exposures and actively removes access to the private files. And another is that the user simply goes offline for a while and returns back with them after the monitoring window. Obviously, the latter is even worse for the victim's privacy.

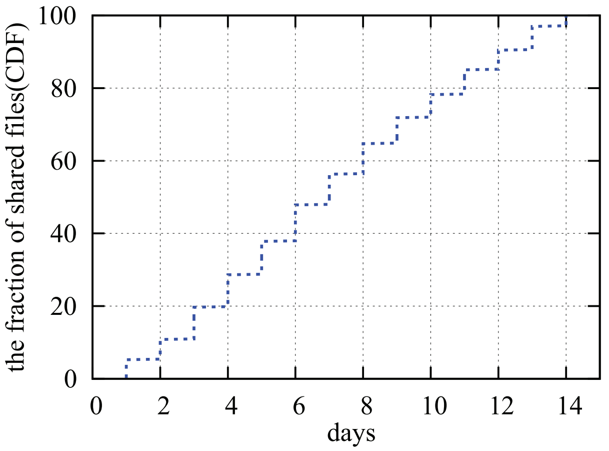


**Figure 4: The distribution of file duration within two weeks.**

Taking "password" as an example, Figure 4 presents the distribution of file duration. In Figure 4, only the 630 private files appearing on the first day are considered. The result shows that the average number of days is 7 and nearly 25% of files are shared for more than 10 days, which means there is enough time for others to download these files. Considering the second case and the limited monitoring window, the actual situation could be perhaps much worse.

Furthermore, the 5,381 (The above filter process ensures that one owner only has one private file leaked) users who share their private files in Kad are mapped to respective countries and regions according to IP addresses, using the MaxMind Database[4]. We give the top 10 distribution on geographic-level in Figure 5. The result shows the user distribution is clearly not uniform. More than 51% of users come from Italy and Spain, followed by France and China. According to [25], the top 4 countries of Kad users are China (24%), Spain (18%), France (13%) and Italy (10%), respectively. It indicates that users of Italy and Spain lack the awareness of privacy protection in P2P file sharing networks, to a higher degree than others.

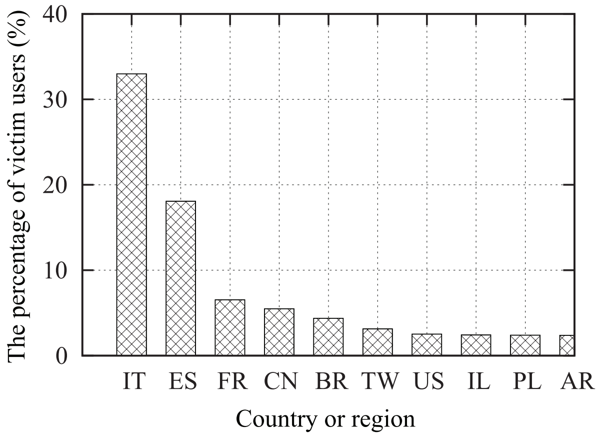


**Figure 5: The distribution of victim users on geographic-level.**

# 5. EXPLOITS

Since the first question is confirmed, we will try to answer **Q2**: **whether potential attackers have already observed such privacy exposures and furtively abused others' personal information**. With the help of Dragonfly and Honeyfiles, we found that 192 potential attackers all over the world tried to download five honey files created by us and at least 45 attackers abused these honey accounts to carry out identity-theft attacks for 125 times, within one month.

## 5.1 Methodology

The methodology can be summarized as follows. Firstly, inspired by Honeyfiles [28], we create five honey files as decoys. Each honey file contains one honey account and the corresponding password. Honey accounts are the accounts registered by us for prevalent web services, including e-finance, instant messages (IM), email and social network. The filenames of Honeyfiles follow the pattern "*service_name* password.txt", which is attractive to any attacker.

Secondly, we set up five eMule clients to share these honey files in Kad. Each client takes charge of one honey file and periodically publishes the 2-level indexes in Kad. Then Dragonfly will record downloading events for these files by monitoring source searching messages, which is similar with the process in Figure 3.

Finally, we log in these honey accounts every day to trace further account abuse. At present, most Web services provide the function of login history, which gives previous login locations, login time or login devices. In particular, Gmail gives users' IPs for last 10 logins and QQ tells user the last login location. Therefore, using login histories, we can capture detailed abusing information after attackers download these honey files.

**Table 2: Results of monitoring honey files from 21st May to 20th June in 2012. "–" means that the login history of this service does not provide location or IP information. QQ and AliPay respectively are the most popular IM and online payment service in China.**

| File | #Downloaders | #Attackers | #Login times |
|---|---|---|---|
| Gmail | 58 | 29 | 59 |
| QQ | 53 | 15 | 47 |
| AliPay | 24 | – | 14 |
| PayPal | 40 | – | 4 |
| Facebook | 39 | 1 | 1 |
| **SUM** | 213/192 | 45+ | 125 |

## 5.2 Experimental Results

The experiment has been conducted for one month, from 21st May to 20th June in 2012. Table 2 presents monitoring results of the five honey files. In total, 192 distinct potential attackers (downloaders) who were monitored in Dragonfly tried to download these honey files through Kad. Further, according to login histories provided by Web services, at least 45 attackers abused honey accounts to carry out identity-theft attacks for 125 times. All honey accounts have been illegally accessed and some were logged in multiple times by the same user. Beyond expectation, attackers seem to be more interested in Gmail and QQ, with a little less login times for e-finance and social network.

In the aspect of geographical distribution, more than 80% of downloaders come from China. For Gmail honey account, 58% of the attackers come from China, followed by Germany (10%) and Poland (10%). It states that more Chinese hackers have been aware of this kind of user privacy exposures in P2P file sharing networks, and begun to exploit it to seek for illegal interests.

Next, we will further figure out the behavioral characteristics of these attackers. After logging into an account successfully, most of attackers would like to change the password in order to control this account. Then according to the service type, subsequent abuses are different. For Gmail, QQ and Facebook accounts, the attacker looks over user private conversations, and adds some ad accounts as friends to conduct spam attacks. For e-finance, users' money is likely to be stolen. During this experiment, we put $5 into the AliPay account on purpose. As expected, it was transferred to another account after two days. This can be used as a clue to trace underground economic chains.

**Table 3: The distribution of the number of honey files downloaded.**

| #Downloaders | #Files |
|---|---|
| 0 | 5 |
| 3 | 4 |
| 1 | 3 |
| 10 | 2 |
| 178 | 1 |

Table 2 shows that the 192 distinct potential attackers tried to download these honey files for 213 times. Some attackers downloaded more than one honey file. Table 3 presents the distribution of the number of honey files downloaded by 192 potential attackers. Though most attackers downloaded only one file, the others who have downloaded more than one should be paid more attention to. Three attackers downloaded nearly all honey files. Because every file is shared by only one client and is much less popular than normal files in Kad, it means some attackers might be powerful enough to observe the entire Kad.

In a nutshell, some attackers have observed and abused this kind of privacy exposures in Kad. P2P file sharing networks probably have become a component of the Internet underground economy chain, and some individuals, even organizations are exploiting it to seek for illegal interests.

## 6. COUNTERMEASURES

For users, the root cause of content privacy exposures is inappropriate usage and configurations on P2P file sharing networks. The experiment results show lots of users do not know exactly which and how many files are being shared from their computers. Therefore the best way to change this situation is to improve users' awareness of privacy protection. Besides some policies and regulations [3] [6]that give users guidelines in safely using P2P file sharing systems, we present a technical solution, **Numen**, to mitigate this problem.

Numen can be implemented as a plug-in for P2P file sharing clients to prevent the content privacy exposures. For most private files, there should be no copies in Kad before sharing them. So the uniqueness can be used as an important attribute of private files, and it plays a key role in Numen.

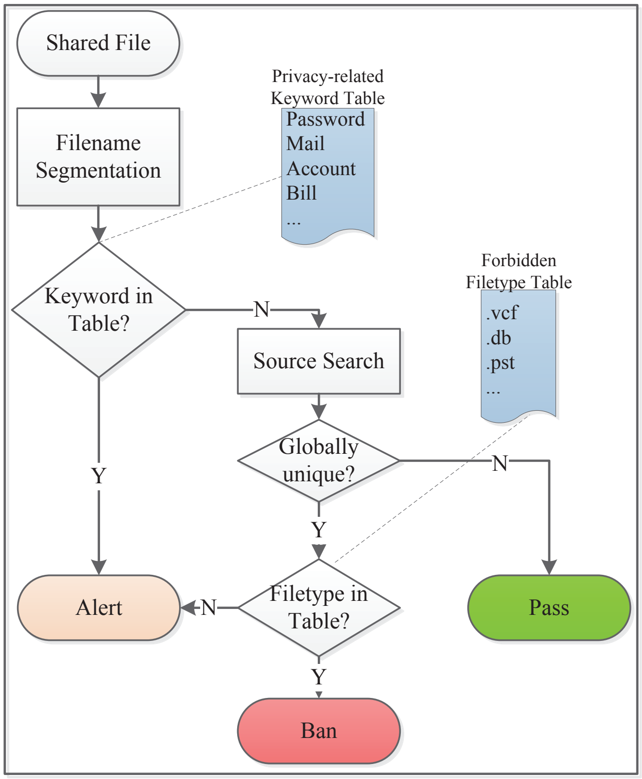


**Figure 6: The process flow diagram of Numen.**

Numen also uses privacy-related keyword table and forbidden filetype table to ease the identification process. The privacy-related keyword table includes keywords similar to Table 1 and the other one contains forbidden filetypes which definitely means private files, e.g., Microsoft Money file ".mny" and Microsoft Outlook contacts file ".vcf".

The process flow diagram of Numen is presented in Figure 6. When the user starts to share a file, the tags (filename segmentation) must pass the examination of privacy-related keyword table. If some privacy-related keyword is found in the tags, a dialog box alerts the user and allows user to decide whether this file can be shared or not. Then Numen launches a source search in the file sharing network to detect whether this file is globally unique, i.e. nobody has shared the same file before. If so, this file is likely to be private due to the uniqueness of private files. The last step is checking the filetype. If the filetype is in the forbidden filetype table, Numen directly forbids this file to be shared. Otherwise, the alert dialog box is given.

We have implemented Numen on eMule. It takes eMule 45s to finish one source search, but multiple searches can be concurrently executed and the results will be cached. Therefore, the delay of sharing files is acceptable. On the other hand, there are somewhat false positives, while the rigorous censorship policies in Numen can eliminate false negatives as much as possible (nearly zero false negatives). In other words, some non-privacy files should pass the filter, but fail to do so. For example, a file containing a sensitive keyword "password" in its filename, e.g., "How-to-protect-your-password.doc", may not be related to user privacy. Another case is that a user shares non-privacy files, which appear in Kad for the first time. To give a quantitative analysis of false positives, it needs lots of efforts to study on the behavioral characteristics of sharing files. The preliminary experiment results show that user content privacy is well protected under Numen, while the ease-of-use feature of eMule can be kept at the same level for most of users.

# 7. RELATED WORK

Since P2P file sharing has become an important application on the Internet and serves tens of millions of users, numerous studies about user privacy have been motivated. However, most previous work focused on location privacy, i.e., who shares or downloads some contents. Based on a number of anonymous communication technologies, e.g., onion routing (Tor), mixes, DC-Nets and secret sharing, some anonymous P2P file sharing systems are proposed and implemented, including APFS [23], Free Haven [13], Freenet [11], GNUnet [7], etc. Though this kind of user privacy can be effectively protected, the considerable performance loss by anonymous communication hinders deployments on a large scale. Recently, Tomas et al. [16] proposed a new P2P data sharing protocol, called OneSwarm. It promises an attractive tradeoff between privacy and performance by flexible and explicit configurable sharing policy. The problem of location privacy has been gradually solved.

In the aspect of content privacy discussed in this paper, only a small amount of studies involved this area. In 2004, Good et al. [15] investigated the similar privacy problem in Kazaa network. By user study, the authors found that a large number of users were unable to know exactly which files they are sharing. They implied that this problem was caused by the unreasonable and confusing user interface of Kazaa client and provided some suggestions to mitigate it.

Davidson et al. [12] in 2003 published a report about the concern of privacy and security in P2P file sharing systems. They also argued that users often share very sensitive private information, by mistake or unknowingly. Some initial approaches, which could improve users' awareness of privacy risks and protection, were suggested from both policy laws and technology aspects. Most recently in 2010, the House of Representatives of USA passed a bill [6] that gave guidelines for securely using P2P file sharing networks. It even prohibited government employees and contractors from using P2P on government computers or when accessing government networks remotely.

Yuil et al. [28] introduced Honeyfiles as an intrusion detection tool to identify attackers. Honeyfiles are bait files which are hosted and monitored by the victim server. Once these files are opened by attackers, the server emits an alert to report an attack. In [22], Nikiforakis et al. tried to reveal privacy risks in file hosting services. It showed that a significant percentage of studied services generated private URIs in a predictable fashion, allowing an attacker to enumerate all stored files. Similarly, Honeyfiles were utilized to identify unauthorized downloading. When these files were downloaded and opened by the attackers, they would automatically connect back to the monitor server. In our work, making use of login histories provided by well-known Web services, we can further know how honey files are abused after attackers download files from P2P file sharing networks.

# 8. CONCLUSION

In this paper we try to evaluate the problem of content privacy exposures in P2P file sharing networks. Due to inappropriate usage and configurations, users probably make their private and confidential files accessible publicly to everyone in the P2P networks, inadvertently and unknowingly. In this paper, we answered two questions on Kad, one of the biggest P2P file sharing networks: *Q1*: Whether and to what extent this privacy exposures exist in current systems, and *Q2*: Whether potential attackers have observed this privacy vulnerability, and abused others' private information obtained from P2P file sharing networks to carry out further attacks.

To monitor sharing and downloading events in Kad, we build the monitoring system called Dragonfly. By using Dragonfly to listen file publishing events related to ten predetermined keywords, we found more than five thousand private files are shared by Kad users within two weeks, over half of whom come from Italy and Spain. What is more, most of such files are shared for more than 7 days, which gives potential attackers enough time to download them. Then making use of Dragonfly and Honeyfiles, we detected that 192 distinct potential attackers tried to download five honey files created by us through Kad within one month. And at least 45 attackers further abused the inner honey accounts to carry out identity-theft attacks for 125 times. Analytical result shows these attackers aim at spying on user privacy and seeking for illegal economic interests.

Finally, we design Numen, a plug-in for P2P sharing clients and implement it on eMule. Based on flexible policies of sharing, Numen can greatly protect users' private files from being inadvertently shared. We hope this work highlights the privacy threats faced by users in P2P sharing networks and promotes user awareness of privacy protection.

## Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions. This research was supported in part by National Natural Science Foundation of China (Grant No. 61003216), DARPA under award HR0011-12-2-005, the National Science Foundation under Grant No. 0842695, the Office of Naval Research under MURI Grant No. N000140911081, and Beijing Technological Innovation Project (Grant No. Z111101055311052).

# 9. REFERENCES

[1] aMule web site. `http://www.amule.org/`, September 2011.

[2] eMule web site. `http://www.emule-project.net`, September 2011.

[3] FTC's p2p alert. `http://www.ftc.gov/opa/2010/02/p2palert.shtm`, June 2012.

[4] An industry-leading provider of ip intelligence and online fraud detection tools. `http://www.maxmind.com`, March 2012.

[5] ipoque web site. `http://www.ipoque.com`, June 2012.

[6] Secure federal file sharing act (h.r. 4098). `http://www.govtrack.us/congress/bills/111/hr4098`, June 2012.

[7] K. Bennett, C. Grothoff, T. Horozov, I. Patrascu, and T. Stef. Gnunet - a truly anonymous networking

infrastructure. In *Proc. Privacy Enhancing Technologies Workshop (PET*. Citeseer, 2002.

[8] D. Bickson and D. Malkh. A study of privacy in file sharing networks. 2004.

[9] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[10] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1):65–75, 1988.

[11] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing Privacy Enhancing Technologies*, pages 46–66. Springer, 2001.

[12] A. Davidson. Peer-to-peer file sharing privacy and security. *Center for Democracy and technology*, pages 1–16, 2003.

[13] R. Dingledine, M. Freedman, and D. Molnar. The free haven project: Distributed anonymous storage service. In *Designing Privacy Enhancing Technologies*, pages 67–95. Springer, 2001.

[14] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *13th USENIX Security Symposium*, 2004.

[15] N. Good and A. Krekelberg. Usability and privacy: a study of kazaa p2p file-sharing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 137–144. ACM, 2003.

[16] T. Isdal, M. Piatek, A. Krishnamurthy, and T. Anderson. Privacy-preserving p2p data sharing with oneswarm. *ACM SIGCOMM Computer Communication Review*, 40(4):111–122, 2010.

[17] H. Kang, E. Chan-Tin, N. Hopper, and Y. Kim. Why kad lookup fails. In *IEEE Ninth International Conference on Peer-to-Peer Computing (P2P'09).*, pages 121–130. IEEE, 2009.

[18] M. Kohnen, M. Leske, and E. Rathgeb. Conducting and optimizing eclipse attacks in the kad peer-to-peer network. *NETWORKING 2009*, pages 104–116, 2009.

[19] B. Liu, T. Wei, J. Zhang, J. Li, W. Zou, and M. Zhou. Revisiting why kad lookup fails. In *12th IEEE International Conference on Peer-to-Peer Computing (P2P12)*, 2012.

[20] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *1st International Workshop on Peer-to-peer Systems (IPTPS'02)*, 2002.

[21] G. Memon, R. Rejaie, Y. Guo, and D. Stutzbach. Large-scale monitoring of dht traffic. In *Proceedings of the 8th international conference on Peer-to-peer systems*, pages 11–11. USENIX Association, 2009.

[22] N. Nikiforakis, M. Balduzzi, S. Van Acker, W. Joosen, and D. Balzarotti. Exposing the lack of privacy in file hosting services. In *Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats*, pages 1–1. USENIX Association, 2011.

[23] V. Scarlata, B. Levine, and C. Shields. Responder anonymity and anonymous peer-to-peer file sharing. In *Network Protocols Ninth International Conference on ICNP 2001*, pages 272–280. IEEE, 2001.

[24] M. Steiner, E. Biersack, and T. Ennajjary. Actively monitoring peers in kad. In *Proceedings of the 6th International Workshop on Peer-to-Peer Systems (IPTPS'07)*. Citeseer, 2007.

[25] M. Steiner, T. En-Najjary, and E. Biersack. Long term study of peer behavior in the kad dht. *IEEE/ACM Transactions on Networking*, 17(5):1371–1384, 2009.

[26] J. Yu and Z. Li. Active measurement of routing table in kad. In *6th IEEE Consumer Communications and Networking Conference (CCNC'09).*

[27] J. Yu, P. Xiao, Z. Li, and Y. Zhou. Toward an accurate snapshot of dht networks. *Communications Letters, IEEE*, 15(1):97–99, 2011.

[28] M. YUILL, J.and ZAPPE, D. DENNING, and F. ANDFEER. Honey-files: deceptive files for intrusion detection. In *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop*, pages 116–122. IEEE, 2004.