Security Analysis of the Foxy Peer-to-Peer File Sharing Tool

K. P. Chow, Ricci S. C. Ieong, Michael Y. K. Kwan,
Pierre K. Y. Lai, Frank Y. W. Law, Hayson K. S. Tse, Kenneth W. H. Tse

Computer Forensics Research Group
Center for Information Security and Cryptography
The University of Hong Kong

1 Introduction

With the expanding capabilities of personal computers and the ever-increasing bandwidth, peer-to-peer (P2P) applications have notably become the most popular means for data transfer over the Internet. One of the very hot applications spanning over Hong Kong, Mainland China and Taiwan is Foxy[1], which is a peer-to-peer file transfer program with Traditional Chinese user interface [7].

Due to the cost-free client program available on the Internet and its simple user interface, Foxy gained its popularity a few years ago when students in upper primary schools and secondary schools started using it to share music and video files. Since the beginning of this year, Foxy has attracted lots of public attention, which include sharing of a pop icon's scandal photos [9] and more recently, the unintended disclosure of sensitive documents by government officers [10, 11, 12, 14].

Beyond question, Foxy is a convenient tool for sharing resources online. However, if files containing sensitive information, like personal data or classified documents, are shared unintentionally, it can be a real disaster to individuals, companies, or even the entire society. It is therefore important for users to observe its working manners and take note of some traps coming along with it. This is the aim of this report to unveil the Foxy working protocol together with its security analysis. Also, it will shed some lights on a few local cases in Hong Kong and outline some

[1] *Foxy* refers to the Foxy Peer-to-Peer File Sharing Tool in the rest of this report.

handy recommendations for general Foxy users or the ones who share computers with them.

2 The Foxy Working Protocol

With the Foxy client installed, one is just a moment away from tons of online resources. Apparently, a user can enjoy the unlimited download capabilities brought by Foxy through three simple steps: *connect*, *search* and *download*. In the following, we will look into each of them in more detail.

2.1 Connect to the Foxy network

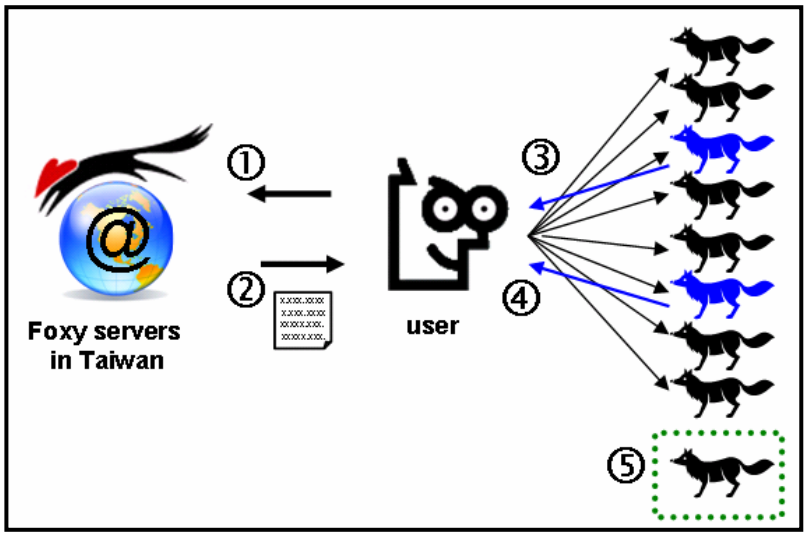

Figure 1: Connecting to the Foxy network (1) connect to Foxy servers (2) peer list returned (3) send PING requests to peers on the list (4) PONG requests sent back from Ultrapeers (5) become part of the Foxy network

There are two types of peers in the Foxy network: Ultrapeers and Leaf Nodes. Ultrapeers are active nodes which are responsible for coordinating surrounding nodes, filtering and directing query traffic in the Foxy network. Leaf Nodes are most common nodes which do nothing except sending out search queries, uploading and downloading files. Through Ultrapeers, communications from Leaf Nodes are relayed from Leaf Nodes to other nodes without flooding the entire Foxy network [5].

To start off, a user (which also refers to the Foxy client program) needs to connect to some Foxy servers to get a peer list. By capturing the network packet sent from a Foxy client, it is found that the Foxy client attempts to connect to servers at *iblinx.com*. Through a preliminary analysis, it is believed that the Foxy protocol is constructed based on the Gnutella protocol [4, 5]. When the peer list arrives, a PING request is sent to each peer on the list to determine if a particular peer is active. PONG requests are sent back from all active peers to the user as an acknowledgement. The user then joins the Foxy network as a Leaf Node of the Ultrapeers and is regarded as a part of the network. Figure 1 summarizes the steps in initiating a connection to the Foxy network.

2.2 Search files on the Foxy network

On the Foxy network, each file to be shared is specified by its name. When a user wants to look for a file, he may enter the file name (or just part of it) as a search query. The *Query* message is sent to the peers, including Ultrapeers, and then passed on to other Leaf Nodes and neighbouring peers. When a peer possesses a file which matches with the *Query* string, it replies with a *QueryHit* message to the requesting user. The *QueryHit* message contains information, like IP address and port number, about the peer sharing the file and the file itself. This enables the user to establish a connection to that peer and initiate the download with the HTTP/1.1 protocol. Figure 2 summarizes the propagation of a *Query* and *QueryHits*.

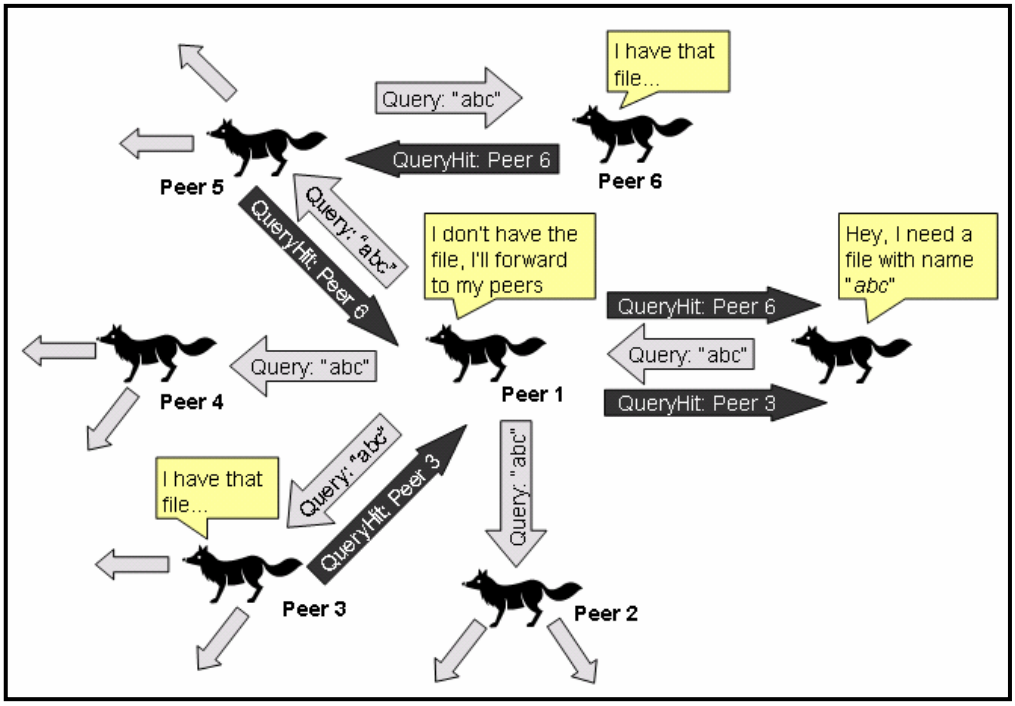

Figure 2: Propagation of a query and *QueryHits*

2.3 Download files through the Foxy network

Upon receiving the *QueryHit* messages along with the necessary connection information, the user chooses a file and requests a download by sending a TCP HTTP GET request, together with his own IP address, to the peer that is hosting the file [15]. The download request can be made by the index number (generated for the purpose of sharing) and file name (e.g., *GET /get/<file index>/<file name>/*), or by a URN SHA-1 value stored in the *QueryHit* message. It is transmitted in the form *GET /get/<file index>/<file name>/* and *GET /uri-res/N2R?<urn value>*, respectively. The hosting peer then responds back and starts sending the requested data. It is worthwhile to note that, a user may send out multiple requests for different portions of a file over the same HTTP connection or to multiple source addresses over multiple HTTP connections. In other words, a user can download a file from one or multiple single sources depending on the Foxy client request message. Figure 3 illustrates a file download on the Foxy network.

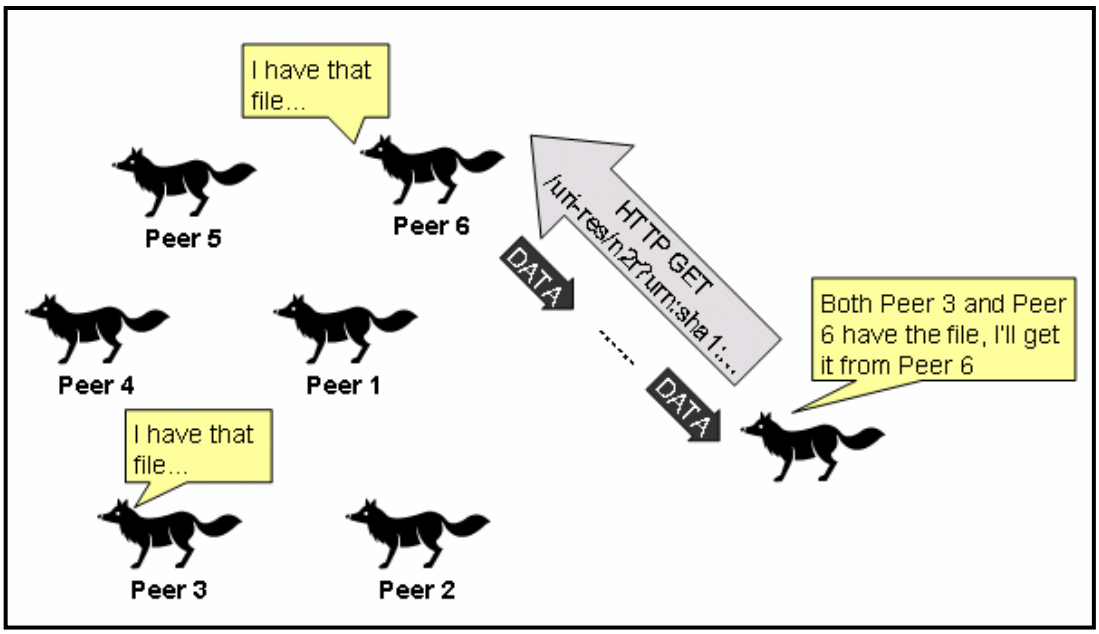

Figure 3: File download on the Foxy network

3 Foxy Cases in Hong Kong

When the Foxy client software is up and running, it is able to share files from the computer without the user's intervention. The scope of sharing is determined by the configuration of the software as well as the settings done by the user. When the pre-settings of the software have not come to users' notice or the users actually make some incorrect settings, files stored at the computers may become virtually available to all Foxy users.

Indeed, there are a few cases involving the unintentional sharing of sensitive files by law enforcement officers in Taiwan and Hong Kong, hitting the recent news headlines. In April 2007, a Foxy user in Taiwan discovered a list of police documents, including the caution statements, telephone directory, police proforma, etc, being shared by someone on the Foxy network [1], causing the government to prohibit their officers from using Foxy at work. The incident caused a great impact to Taiwan police and raised an alarm to the public regarding the significance of information security. Notwithstanding, similar incidents happened in Hong Kong in May 2008 when a number of sensitive departmental documents of the Civil Aviation Department [10], the Immigration Department [14], the Hong Kong Police Force [11] and the

Customs and Excise Department [12] became available to download with Foxy. Though the related departments intended to remove the leaked data at once, Foxy's scattered design and the extensively distributed network posed a great hindrance in the removal of the subject documents. It is obvious that the installation of the Foxy client software at the computers where those documents stored is the cause of the leakages. But more importantly, the deficient awareness of information security of the involving parties is the major concern in these incidents.

What made a greater furor is the overwhelming sharing of a pop icon's scandal photos in February 2008 [9]. Most local newspapers headlined the story in their front page for weeks. Since the early stage of the event, the Hong Kong Police has announced with a lofty tone that publishing any of the relevant photos on the Internet is against laws. However, the photos have been handed around wildly over the Foxy network for months. It is believed that the photos were initially distributed with Foxy using the code "hurry on bit the fox" and using the keyword "新閃卡".

Being an Asia World City, the household broadband penetration rate in Hong Kong as in March 2008 is 77% and the total number of public Wi-Fi access points as in May 2008 is 6,841 [13]. These are signs of Hong Kong becoming a real High-tech city. But it is sad to admit that, the high Internet penetration rate and the well-built telecommunication infrastructures are now exploited by P2P networks. According to Velocix, it is estimated that between 60% and 80% of capacity on consumer ISP networks is consumed by P2P [3]! As more and more people join the P2P community, it is envisaged that the trend will keep upsurging in the forthcoming years. In particular, Foxy is one of the emerging tools generally adored by local school kids and teenagers. Apparently, it offers the convenience of sharing information online very easily. It is also clear that Foxy can harm a user badly if the tool is not properly configured or the user is not aware of the security settings of their computers. To help users to enjoy the advantages of the technology and at the same time balance the security needs, we will further analyze the problem and propose better ways to close the existing loopholes in the following section.

4 Foxy Security Analysis

P2P file sharing software has become popular since the advent of the very famous *Napster* in 1999. One may question, apart from Foxy, why there are no serious security incidents in other implementations, like the BitTorrent (BT) network [2]? Also, what are the potential causes of the leakage incidents happened in the Foxy network? This section reveals some facts that may help to answer these questions.

4.1 Special Features of Foxy

In the Foxy configurations, a share folder list is used to define the folders and the files to be shared with other Foxy users. By default, the list only contains the folder "*C:\Program Files\Foxy\Download*", where downloaded contents from Foxy are stored. In other words, if a user adopts the default settings, all files downloaded from other peers are automatically shared, until the user removes them from the share folder list or the share folder explicitly. Users are allowed to include any other folders in their file systems into the shared folder list by appending the paths of those folders to the list. All files reside in these shared folders are searchable by all Foxy peers.

Like many other desktop applications, the Foxy client software, by default, is launched whenever a user logs on to Windows. It is then minimized to the system tray and only a tiny icon is displayed. With the "*Hide inactive icons*" feature introduced in Windows XP (as shown in Figure 4), users may not even see the Foxy icon until they press the arrow button in the taskbar. This enables the Foxy client software to execute and share files silently in the background.

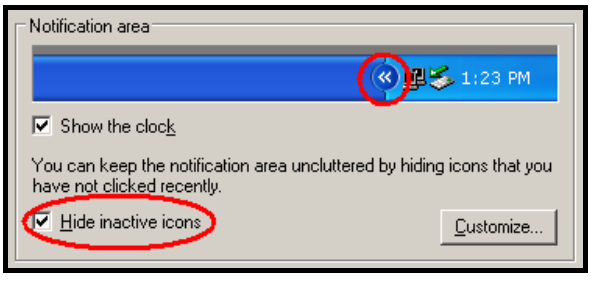

Figure 4: The "Hide inactive icons" feature in Windows XP

However, all these are actually common for P2P file sharing software. What makes Foxy really different from the others is its distributed topology. The Foxy network relies on no tracker server to store file sharing information, instead the information is distributed among thousands of Foxy peers. Users can use simple search queries to look for files available in the network. This may result in a larger exposure of the shared files comparing to other P2P file sharing networks, where additional information (e.g. torrent files in BT network [2]) is needed to download a file. The distributed topology also makes the revocation of a shared file in the Foxy network technically unfeasible. If some sensitive information has been shared in the Foxy network, it is impossible for the authority to technically stop it from further spreading.

4.2 What may have happened in the data leakage incidents?

The special features of Foxy mentioned above are nothing bad: fewer user interventions are required, file searching is simple but effective, and shared files exist in the network just about forever even without the initial uploader staying online. These enable Foxy to become an optimal file sharing channel. The main concern here is, why and how those sensitive documents become available on the Foxy network. After analyzing the behaviors of Foxy, we list out five possible reasons regarding the leakage incidents, namely:

1. Human faults
2. Misconfigurations
3. Program bugs or vulnerabilities in the Foxy client software
4. Modified versions of Foxy client software by hackers
5. Inherited configuration from previous users

In the following, we are going to discuss each of them in more detail.

4.2.1 Human faults

The sensitive documents may be copied by the users to the Foxy shared folder, either with intention or by accident. However, this is unlikely to be the sole cause for all the revealed incidents which involved a number of different parties.

4.2.2 Misconfigurations

The folder containing the sensitive documents may have been selected in the share folder list. This may not be the cause since the users need to explicitly select the folder in the user interface of the Foxy client software (Figure 5). If it is the case, that would be an intentional sharing instead of an unintentional one. In Figure 5, the folder "*Windows*" under drive *C:* is selected to be shared, thus all documents in "*C:\Windows*" are shared to all other Foxy peers.

Figure 5: Selecting a folder to share in the Foxy client software

4.2.3 Program bugs or vulnerabilities in the Foxy client software

There may be some program bugs in the Foxy client software which makes folders and files outside the share list searchable by other Foxy users. There may also be vulnerabilities in the client software, allowing remote attackers to retrieve arbitrary files from target machines. However, these claims have been officially denied by the Foxy official website [6].

4.2.4 Modified versions of Foxy client software by hackers

There are some unofficial versions of Foxy client software available on the Internet. These versions have been modified by third-parties and claimed to have additional features than the official ones. Hackers may include malicious codes or configuration settings in these modified versions. Users attracted to its claimed features may have their computer compromised.

We have identified some *green versions* of Foxy client software available on the Internet [8]. They have all the needed binaries and configurations included in a package so that no installation is required to execute the software. Configurations are stored in configuration files under the folder "*Conf*" within the directory that contains the Foxy executable. The advantage of using the green version is that Foxy can be started on a removable device without installing Foxy onto the personal computer. One of the Foxy configuration files named "*Settings.cfg*" is found to store the share folder list. If a hacker creates one "*Settings.cfg*", adding the folder "*C:\Documents and Settings*" into the share folder list and then include this compromised "*Settings.cfg*" in a green version package, all users running this green version software will have their "C:\Document and Settings" folders shared by default. We called this hacking technique *poisoned configuration file.*

Up to the moment when this report is drafted, we cannot find any other hacked versions of Foxy client software except the above discussed green versions with poisoned configuration files.

4.2.5 Inherited configuration from pervious users

On a shared computer, a user may not always be aware of the configuration set by other users. Consider the following scenario that a personal computer (PC) is shared by *the kid* and *the father*:

> *The kid wants to share a MP3 song in his USB thumbdrive by Foxy. The kid plugs the thumbdrive into the computer and the OS automatically assign a drive letter F: to the thumbdrive. He then adds F: in the share folder list through the Foxy user interface. After some time the file is shared, the kid removes the thumbdrive from the computer. The drive letter F: is released by the OS. He keeps the Foxy client running in order to download a long movie found in the Foxy network. After a while, the father then takes control of the computer. He plugs his own USB thumbdrive, containing confidential documents from his working place, into the computer. The OS automatically assign the drive letter F: to the thumbdrive. And then he opens some of his document on the thumbdrive for working. Since the drive F: is in the share folder list, all files located in the current drive F: (i.e. the father's USB thumbdrive) are searchable by all Foxy peers.*

The case can also be applied when a CD/DVD drive is added to the share folder list of Foxy – files in different CD/DVD can be searchable without changing any of the Foxy configurations. Similar scenarios appear when a *foreign* user uses another person's PC, which has Foxy installed and running on it. The owner of the PC may have shared everything from his PC while the foreign user is not aware of. When the foreign user reviews his confidential documents on an external USB disk, all his documents may then be shared by Foxy without his knowledge.

4.3 Security Analysis

In Hong Kong, sharing computers is rather common in households or even workplaces. The above scenario or similar ones could be happening from time to time. As a result, among the five reasons suggested above, we believe the last two reasons are likely to occur and the analysis is summarized in the Table 1.

	REASON	ASSESSMENT
1	Human faults: user copies confidential documents to the shared folder	Very unlikely since so many incidents
2	Misconfigurations: user shares a folder that contains confidential documents	Possible but unlikely
3	Program bugs or vulnerabilities in the Foxy client software	Foxy Media officially denied
4	Hacked versions of Foxy client software by hackers	*Poisoned configuration file* in the Foxy green version from some websites were found
5	Inherited configuration from previous users	Likely if the PC is shared by more than one user with single user account

Table 1: Security analysis summary

5 From Foxy Saga to Security Principles of Personal Computers

The swelling of malicious attacks and security vulnerabilities in information infrastructures has caused organizations to set out a number of security principles or strategies to minimize their damages. On the other hand, most ordinary PC users are found not observing these principles at all. Some of them may have the naive impression that data can be perfectly safeguarded through the installation of anti-virus applications on their PCs. However, the signature-based detection technique can only deal with known viruses and sometimes can be evaded rather easily. Security principles or strategies have to be used to reduce the damages to data owners. The recent saga by Foxy users has further shown that information could have been compromised even when the defending applications are functioning properly. In fact, PC users have to beware the importance of security principles in order to protect their information, especially if they are using P2P tools for file sharing.

While the whole breadth of security principles is not to be discussed, this section focuses on two fundamental security principles that are considered vital for PC users who are engaging in file sharing activities using Foxy. They are the *Principle of Least Privilege* and the *Principle of Compartmentalization*.

5.1 Principle of Least Privilege

The principle of least privilege was first published in 1975 [16]. It states that every user or program should only be granted with the minimum set of permissions to perform the designated tasks. The principle is based on the assumption that when a system has multiple users who are assigned with excessive rights, the vulnerabilities of misuse of the system or unauthorized data access can occur. Through the establishment of user accounts with different levels of privileges, the system can be protected from intended or accidental misuses.

Although the least privilege principle can avoid many security breaches, most system administrators violate this principle when assigning privileges to users [17]. This is simply because granting full privilege to all users is much easier than determining the tasks of each user and defining the appropriate permission levels of them. For ordinary PC users, the situation can be more confounding. Firstly, many users may not be computer literate and hence do not notice the least privilege principle, let alone its importance to the whole computer system. Even if they learn about the principle, they may not be competent enough to configure their PCs properly. Secondly, many users are using Windows systems and use the account *Administrator*, which is the default account when Windows systems are installed. Thirdly, there are many Windows applications and tasks that require users to have administrator privilege. For the sake of convenience and simplicity, many users therefore run with administrator privilege and this escalates the security flaws.

In brief, when Foxy is running with administrator privilege, all documents and data may be shared and the scope of sharing depends on the configuration settings. If Foxy is running using the least privilege user account, sensitive data protected by a more privileged account are not accessible by Foxy and therefore will not be shared unintentionally even there are misconfiguration settings in Foxy.

5.2 Principle of Compartmentalization

The principle of compartmentalization is similar to the principle of least privilege, which also aims to minimize damages from vulnerabilities. Instead of creating different least privilege user accounts, compartmentalization refers to the segmentation of a system into several isolated components or compartments. As such, any security breach in a compartment would not affect other compartments of the system [17].

Compartmentalization is usually implemented together with the least privilege principle. It is believed that compartmentalization is able to provide a finer granularity of control, which can render a more restrictive access control on least privilege principle. For example, if a user has granted privilege to access restricted information together with shipping and marketing compartments, he can access data at the restricted level with no compartments. Although the right to access restricted data is granted, he is not allowed to access restricted data inside the personnel compartment. In other words, if a user wants to access a document in a compartment, he needs to have equal or higher access privilege and also be a member to that compartment.

In Windows XP and Windows Vista, each user has his own user directory and is not accessible to any other user except users with administrator privilege. If all user accounts are not granted with administrator privilege, each user account will have its own compartment to store data and documents. Apart from the administrator account, any users using Foxy can only share data and documents in his own compartment.

Referring to *the father and the kid* case under Section 4.2.5, if they are using different accounts and are assigned with different compartments, *the father*'s documents, being in another separate compartment, would not be shared even there are some erroneous Foxy settings.

6 Recommendations

Network security is only as strong as its weakest link. Accordingly, our recommendations are:

1. Apply the least-privileged user account approach. That approach ensures that users follow the principle of least privilege and always logon with limited user accounts. That approach also limits the use of administrative credentials only to administrators. Further, accounts should not be shared. Consequently, other users are unable to share, intentionally or accidentally, files belonging to another user.

2. Sensitive data should not be transferred via Foxy.

3. When using other people's PC, always check whether or not Foxy is running there. If it is running, turn it off at once.

4. Before connecting a USB drive to a computer, always check whether or not Foxy is running on that computer and whether or not that drive is a shared drive.

5. Obtain a copy of a stable version of Foxy only from the official website. This avoids installation of a copy which may contain bugs or malicious code like Trojan horse, back door or worm.

6. Do not configure Foxy to start at startup of the computer. This is because once the computer is connected to the World Wide Web, other Foxy users may obtain shared files from the computer even the user is not intended to let Foxy run.

7. Do not select an entire disk drive as shared drive. A user tends to select a larger disk drive to store downloaded files from Foxy. That allows other files stored on that disk drive to be available to other Foxy users.

8. Folders such as "*Desktop*", "*My Documents*" should not be used as a download folder or sharing folder. This is because those folders usually contain files not intended for sharing. Also, those folders are common locations for placing files

currently in use for just a while, the files available there can be rather unanticipated.

9. Always scan a downloaded file for virus, Trojan horse, back door or worm. Files downloaded could be just what a downloader expects, but they might also be a vehicle for malicious intent.

10. Always turn off Foxy after use.

In recent years, a number of peer-to-peer file sharing tools have been developed and many others are expected to emerge in the future. In general, the above recommendations can be applied to common P2P file sharing tools (except BitTorrent which requires a torrent file to initiate an upload). These recommendations not only address the security problems at present, but also be applied to future P2P tools which adopt similar working principles.

References

[1] Baidu 百科: Foxy, http://baike.baidu.com/view/1297167.htm; accessed on June 14, 2008.

[2] B. Cohen, The BitTorrent Protocol Specification, http://www.bittorrent.org/beps/bep_0003.html; accessed on June 14, 2008.

[3] J. Delahunty, Global bandwidth use rises due to P2P, http://www.afterdawn.com/news/archive/6298.cfm; accessed on June 14, 2008.

[4] Gnutella Protocol Specification, http://gnet-specs.gnufu.net/; accessed on June 14, 2008.

[5] Gnutella2 Developer Network, http://g2.trillinux.org/; accessed on June 14, 2008.

[6] *FOXY公告*. http://forum.mymaji.com/announcement.php?id=19#19; accessed on June 14, 2008.

[7] Foxy官方網址, http://www.gofoxy.net/; accessed on June 14, 2008.

[8] Foxy綠色版下載, http://www.downbai.com/wdown/5136.html; accessed on

June 14, 2008.

[9] 網上流傳藝人裸照英皇報警, Ming Pao, January 28, 2008, http://hk.news.yahoo.com/080128/12/2nzrv.html; accessed on June 14, 2008.

[10] FOXY軟件泄政府機密 警內部手冊 高官出生日期 一覽無遺, Ming Pao, April 5, 2008, http://hk.news.yahoo.com/080404/12/2rtmw.html; accessed on June 14, 2008.

[11] 警方泄機密性質嚴重 應禁用Foxy處理機密, Ming Pao, May 27, 2008, http://hk.news.yahoo.com/080526/12/2unwg.html; accessed on June 14, 2008.

[12] 病歷海關口供女警自薦信 Foxy又泄政府機密, Ming Pao, June 14, 2008, http://hk.news.yahoo.com/080613/12/2vp8f.html; accessed on June 14, 2008.

[13] Key Telecommunications Statistics, Office of the Telecommunications Authority. http://www.ofta.gov.hk/en/datastat/key_stat.html; accessed on June 14, 2008.

[14] 入境處機密文件外泄 網上任睇, Sing Tao, May 8, 2008, http://hk.news.yahoo.com/080507/60/2tnkf.html; accessed on June 14, 2008.

[15] P. L. Piccard, B. Baskin, C. Edwards, G. Spillman, M. H. Sachs, *Securing IM and P2P Applications for the Enterprise*, Rockland, MA : Syngress, 2006.

[16] Jerome H. Saltzer and Michael D. Schroeder. The protection of information in computer systems. In *Proceedings of the IEEE*, volume 63(9), pages 1278-1308. IEEE, September 1975.

[17] John Viega and Gary McGraw. *Building Secure Software - How to Avoid Security* Problems *the Right Way*. Addison-Wesley, September 2002.