Title: Monitoring Pedophile Activity in a P2P Network.

Authors: Philippe Jarlov[1], Matthieu Latapy[2], Frédéric Aidouni[2], Clémence Magnien[2], Christophe Berger[2], Frank Crispino[1*]
(1) Ministere de l'intérieur, de l'outre-mer et des collectivites territoriales, Gendarmerie nationale, Région de gendarmerie d'Aquitaine, Section de recherches de Bordeaux, Quartier Beteille, 33270 Bouliac. Tel : 05 56 68 82 00.
(2) LIP6 CNRS (Laboratoire d'Informatique de Paris), Universite Pierre et Marie Curie, 104 avenue du President Kennedy 75016 PARIS - www.lip6.fr
* Corresponding author : frank.crispino@gendarmerie.interieur.gouv.fr

Introduction.

Internet could easily be considered as a world apart, governed (or not) by its own laws and rules, in which « things » take place independently of real world. However, the first role of the Internet is to help communication between real people, making it easier for people to exchange real objects (although they are often digital objects), and, consecutively, create real relations between individuals.
The point is reached when these real objects are photographs or videos of abused children, obviously being real victims [1].
Hence, Internet appears as a very efficient and empowering mean of communication between pedophiles, who deserve no different laws than the ones applying to the real world. The pedophile nature of their activity has to be taken into account [2], whatever it is expressed on the Internet, outside of it, or both.
However, Internet is not a coherent and simple medium, like the phone for instance [3]. Instead, it is composed of many application layers allowing criminal exchanges in various ways. Investigating the communication tools used by pedophiles is then a key resource for understanding their way of thinking, anticipating their activity, and identifying involved individuals.

IRC (Internet Relay Chat) [4] is among the oldest Internet tools, providing chat facilities in public or private rooms, and exchanges of files (including pictures, video and text). Pedophiles are classically using this channel for mutual information (like pedomom®), or files exchanges (like preteengirlsexpics®).
The web [5] is obviously the most widely known tool for paedophile activity, making it possible to create web sites with pedopornographic content. The emerging web 2.0® develops more and more private and semi-private spaces, with chat, email, or blog facilities, in which pedophiles are also involved.
These networks are not only used by pedophiles as means of communication between each other; they are also used to identify possible victims. For instance, involvement of minors in web 2.0® networks (like online public personal diaries) are unending source of targets for pedophiles. Once the target is identified and contact is established, the predator tries to start online chat (often via MSN™, which young people are found of). They may also exchange files and use webcams through such tools.
Finally, pedophiles are present in all Internet layers, from the oldest ones like IRC, BBS or Newsgroups, to the most recent ones like virtual worlds. For example, on Second Life® , a virtual world where figures (avatars) live and are controlled via the Internet by their creators, pedophiles invented their own subspace called Wonderland® where figures have sex with young minors and fantasize about them.
P2P files exchange networks, which play a constantly increasing role on the Internet are not an exception. A significant portion of the millions of files exchanges daily in such networks consists of pedopornographic pictures, movies or text files. Having a better knowledge of these networks and pedophile activity occurring through this channel would help much in identifying victims and suspects.This paper presents a work conducted recently on ®, which is one of the most widely used

P2P networks [6].

1. Monitoring P2P exchanges.

Observing activity on the Internet is a challenge in itself (independently of P2P activity or not), the key problem for law enforcement being user identification.
Indeed, a user may be identified through the IP address of his/her computer connected to the Internet. This identification may in principle be provided to law enforcement institutions by the Internet access provider of the user, generally through a judicial warrant. IP addresses of users however change during time : a user may have several addresses, and the same address may be used by several individuals. Moreover, public spaces provide Internet access to users without necessarily identify them (correctly), and a hacker may use the address of someone else.
Therefore, while obtaining the identity of the user associated to an address is uneasy, checking that this user is indeed the one that conducted the action under concern is even more difficult.
In addition to these difficulties relative to the monitoring of Internet activity in general, the specific features of P2P networks make it even more challenging to monitor its specific activity.
The key principle underlying P2P networks is their distributed nature: there is no central server with a global knowledge of the activity in a P2P network as a whole. Instead, peers (*i.e.* users) self-organize into a distributed network. The exchanges occur directly between peers (peer-to-peer), without the involvement of a central service.
As a consequence, the information on the activity of such networks itself is disseminated in many places, which all have a very partial view of the global activity. One cannot monitor activity in such networks by contacting a central service, like phone operators for instance.
Some P2P networks, though, have a *semi*-distributed structure only : they rely on a few hundreds of servers, which manage user queries and directories of available files. Still, they have no knowledge of the exchanges actually performed between users, and the servers are in general ran by users themselves.
The eDonkey® network (also called Emule®) [7] is a semi-distributed P2P system. The basic running principles of this system are as follows.

- When a user enters into the system, it connects to one server and announces the list of files it provides. The server then adds this information to its directory of available files and adds this user as a possible provider for these files. The user periodically sends an update of the list of files it provides.
- When a user is looking for some content, it first sends a keyword-based query to the server to which it is connected. The server then looks for files that it knows which fit this query (in general, this simply means that the file name contains the entered keywords). It sends a list of appropriate files to the user, who may choose one or several files in it.
- For each file selected by the user, the server sends a list of possible providers (other users). The user may then contact these provider directly in order to obtain (parts of) the file(s).
- Finally, eDonkey® servers operate like file/provider directories and search engines in these directories. They store none of the exchanged files, and are not involved in actual exchanges themselves.

Other features of P2P exchanges make it particularly challenging to observe them globally: their sheer size and their poorly structured nature.
Indeed, dozens of millions of users are involved in P2P systems on a daily basis, in various systems and in many countries. They exchange dozens of millions of files, with new files added at a high frequency .
For instance, a ten week measurement of a medium-sized eDonkey® server [8] led to the observation of approximately one billion queries, from 80 million users exchanging 275 millions files. Observing such amounts of exchanges, and analyzing them, is a challenge in itself.
Moreover, the dynamics of users is extremely high: they connect and disconnect freely. They may have changing addresses, and changing roles (a same computer and software may be used by the

different individuals composing a family). Likewise, many variations of a given file are available (with filename in different languages, for instance, or various file formats or quality). Finally, there are many fakes in such systems (files with a name different from their actual content), as well as malicious users (for instance, users who do not follow the rules of the system in order to improve their own performances).
P2P activity is therefore huge, and extremely noisy. Information collected is complex, poorly structured, and subject to errors. It is however very rich, as it makes it possible to observe users, and in particular the ones involved in pedophile activity, at an unprecedented scale.
In face of this situation, the key goals of a law enforcement investigator fighting pedophile exchanges are:

- to identify files of interest, generally using keyword queries or files with identified pedophile content.
- to get possible providers for these files by sending queries in the monitored P2P system.
- to confirm that these providers are providing the file, generally by downloading it from them.
- To identify the real user behind the computer, crosschecking the different collected data with the time of connection (content, IP address, identity of the user associated, different addresses, etc.).
- To confirm the hypothesis through a computer search for relevant traces.

Obviously, such investigation does not look easy . It may lead to far too much information which is not precise enough. For instance, one may identify people who provide pedophile content by error (sometimes without even knowing it). Although this is illegal in most countries, these users are certainly not key targets. On the other hand, identifying users who introduce new pedophile content is of prime interest.
In this context, a deep technical knowledge of P2P networks and their underlying principles, as well as rigorous and powerful inference methods are needed. Much work has been done on eDonkey® and Gnutella® [9], as well as on other protocols.
However, current knowledge of pedophile activity remains very limited, even on these networks. Likewise, tools for investigation and law enforcement remain insufficient. We describe below an effort in this direction, in the case of eDonkey®.

2. A monitoring tool for eDonkey.

We present in this section a tool developed during a collaboration between researchers and law enforcement personnel. It consists in a library of eDonkey primitives written in Python®. With this library, several measurements may be conducted.
eDonkey® is a poorly documented protocol: the server code is not publicly available, and there is no official documentation. Many clients, though, are open-source. One may therefore read the source code of these clients to get insight on how the protocol works, and may then develop new clients; this has been done in various occasions [7].
Still, available codes are difficult to read and understand in general: the protocol in itself is complex, and programmers often implement intricate tricks aimed at improving the performance of their client. In addition, they may contain bugs and various flaws.
Moreover, we need something slightly different from an actual client for our task : a library which makes it possible (and easy) to write measurement applications.
To this regard, a graphical user interface has little interest, for instance; instead, one may expect an interactive tool able to execute commands and batch files. For this purpose, we used the Python® programming language. In addition, as we want to conduct measurements that run automatically during long periods of time, the stability of our software is a key point. Likewise, we want to use the collected data for both scientific and law enforcement purposes. It must therefore be highly reliable and rigorously collected.
The key operations in eDonkey® are divided in three categories: identification and house keeping, file searches and source searches. We have implemented most of these operations in our library.

- *Connection.* This is an handshake sequence bound to the TCP (Transfert Control Protocol) part of the eDonkey® protocol at the end of which the client is identified on the server with

a clientID, attributed by the server.

- *Server list.* Each eDonkey server is started with a list of friend servers as a parameter. We can query this list to extend the list of servers we know. As explained below, our software starts with an intensive use of this facility.
- *Keep alive.* eDonkey® servers probe their clients to detect faulty communication links; we must answer these probes.
- *Statistics.* We can query a server to get its client count, or indexed file count (although these claimed numbers should be trusted with care).
- *File searches.* These are metadata-driven file search queries. Typical search criteria are words which appear in filenames, size of files, or their types (audio or video for instance). The server answers with a list of files, with their known metadata and one provider for the file. One may therefore query a server with a keyword and obtain a list of filenames and hashes fitting this keyword.
- *Sources searches.* With a list of file hashes, one may query a server for a list of providers for these files. It is the last query type that is used by clients before contacting peers to initiate downloads.

We implemented in Python® language a procedure for each of these operations.
In addition, we implemented a powerful connection procedure aimed at connecting to as many servers as possible: it first opens a connection with a list of known servers (*Connection* operation above), asks to all these servers the other servers they may know (*Server list* operation above), then opens a connection with them, and iterates this process until no new server is discovered. We typically reach this way between 100 and 200 servers (with a classical initial list).
As a consequence, our measurement tool is multi-servers. Queries sent for our measurement are sent to all these servers, and all their answers are recorded.

Using this toolbox, we conducted a measurement from a singe machine which sent approximately every 12 hours and during 210 days (7 months) in continuous, a set of keyword-based queries. It recorded the answers to these queries (basically, lists of filenames and hash codes) and then asked for providers for each of these files. Again, the obtained lists of providers were recorded. For privacy protection, all the data were anonymized, but we performed a geolocation operation on each observed IP address, thus obtaining the country in which it is supposed to be.
The goal of this measurement was appreciate the relevance of our tool, and to provide some insight on observed pedophile activity (see Section 3 below). Therefore, the keyword-based queries we used were known typical pedophiles ones like *qqaazz, aabbccddee, babyshivid, hussyfan, pthc, ptsc, r@ygold, and kingpass* to be compared with non ones as *porn, madonna, linux, batman, cnrs, mickael jackson, sex.*

3. Application to cyberpedopornography.

a) Statistical analysis for improved knowledge of pedophile activity
During the experiment, we observed 3,229,715 distinct files, among which 790,505 (24.5 %) contained at least one clear pedophile keyword in our list in their name. We will call these file pedophile files. We observed also 3,599,451 distinct providers (IP addresses), among which 1,391,718 (38.7 %) provided at least one pedophile file.

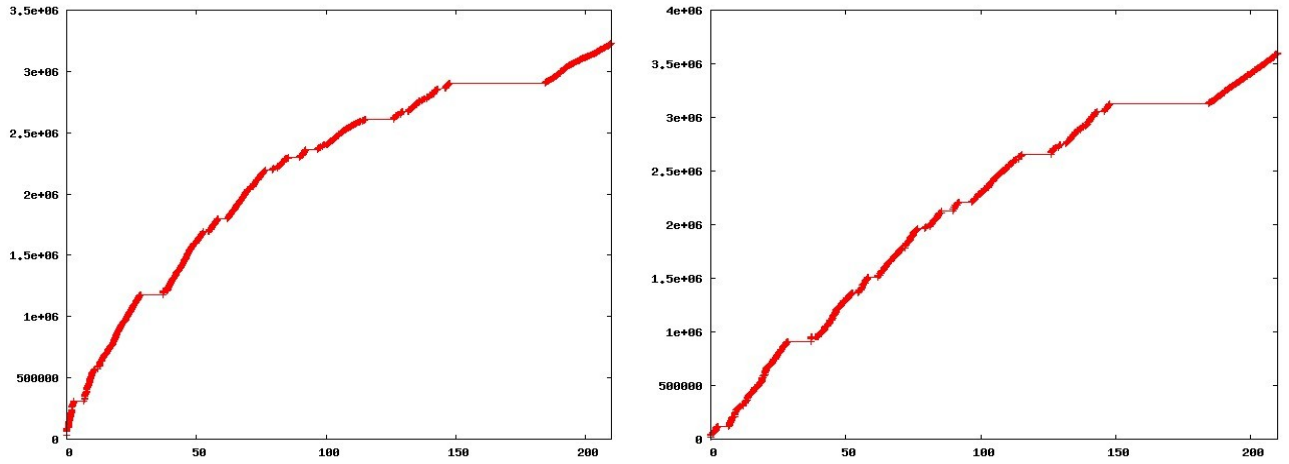

Figure 1

Left : Number of files discovered in our measurement, as a function of time (horizontal axis: days of measurement; vertical axis: number of files observed since the beginning).
Right: number of peers (IP addresses) discovered in our measurement, as a function of time.

The number of files discovered during this time is given in Figure 1, as well as the number of providers. It appears clearly that the growth of these numbers is significant, event after a very long period of time. It shows that continuous measurement is certainly relevant.
During such long measurements, one necessarily experiences network shutdowns, though, as is visible in Figure 1 (each interruption induces a plateau in the plot).

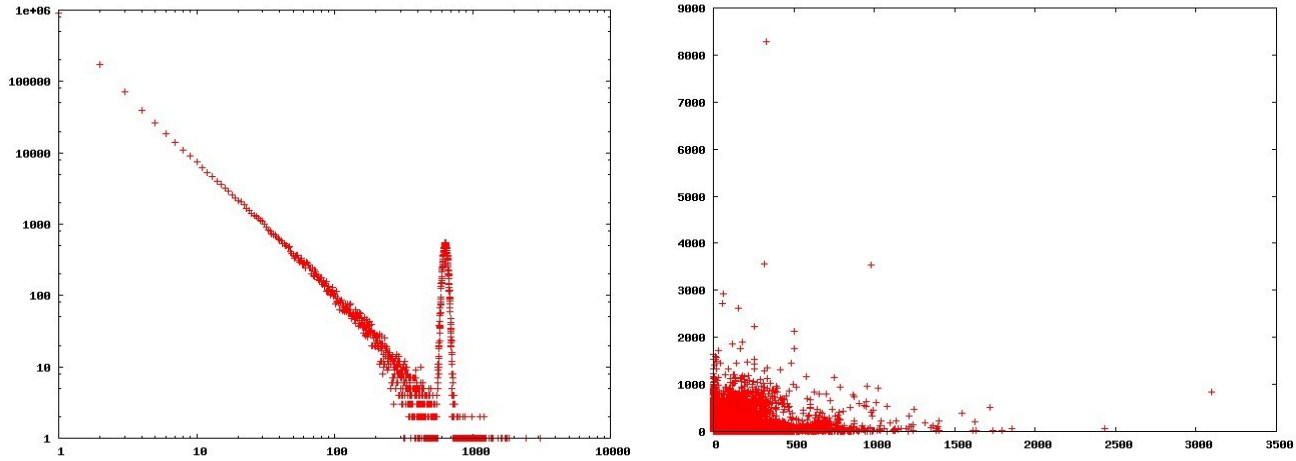

Figure 2

Left : Distribution of the number of pedophile files provided by each user, *i.e.* for each number *x* on the horizontal axis, the number of users (vertical axis) which provide *x* pedophile files. For instance, 7,388 users provide 10 pedophile files, which is visible by the fact that the plot goes through point x=10, y= 7,388. (This plot has a singular shape around *x*=600, but this is due to measurement artifacts which we do not detail here.)
Right: correlations between the number of pedophile files and non-pedophile files provided by each observed user: on the horizontal axis, the number of pedophile files; on the vertical one, the number of other files; for each observed peer, we draw a point at the coordinate given by the number of pedophile files he/she provides and the number of other files he/she provides.

We observe that the amount of pedophile files and of providers of such files is important. We plot in Figure 2 the number of pedophile files provided by each user; it appears clearly that, although we observe only a few files for most users, some users provide many pedophile files, up to 3,000, with

33,118 users providing at least 500 pedophile files.
If we observe the correlations of the number of pedophile files vs other files, see Figure 2, it appears clearly that some users have a strong pedophile bias: they provide many pedophile files, but only few other files. Of course, the converse is also true: some users provide mostly non-pedophile files. Interestingly, we also observe users who provide both types of files similarly, indicating that a user may provide pedophile files without having a strong focus on this kind of content.
It must be clear, though, that this measurement is limited and that the results presented here are preliminary ones : confirming them and making them more precise are our future perspectives.

b) Identification of targets for law enforcement
The identification of infraction suspects is now possible by using the tool "EdkExplorer©" (developed during this project).

Steps are followings:

- Connect "EdkExplorer©" to eDonkey® network: "EdkExplorer©" is a software programmed in Python® language ; its installation is possible on any operating system, as Python® is known for is portability. The software processing is managed on line, no graphical interface has been programmed (to provide simplicity and portability of the system). The start-command connects "EdkExplorer©" to a variable number of servers, from 60 to 150 on average, and assures this way a good overview on the protocol. We can surely say eDonkey® network is composed of about 150 to 200 servers. Note that a client (eMule®) is usually connected to only one server.
- Question the network, from keywords, hash, eDonkey® links on.
  - There are two kinds typically used keywords:
    - terms which leave no doubt about the file nature: for example, "pedo – pedoteen – preteen – littlepussy..."
    - "secret" terms which doesn't mean anything at first sight: for example, "aabbccddee" or "qqaazz" relative to pedopornography presenting very young children (less than five years old)
- Obtain replies and sort out results: getting replies in text file form and sorting on a visual way:

Example :
86.218.146.xxx:5551 (IP address)
Anancy-157-1-75-235.w86-218.abo.wanadoo.fr (Network informations and ISP Internet Service Provider)
Épinal, Lorraine, France (Town, Region, Country)
2009-05-07 16:47:27.044550+02:00 4c2260e1c6fa89f7efab48b253a0d273
my 14 yr old sister bathing lolita qwerty ddoggprn reelkiddymov preteen tits nipples pussy.jpg
(Date and Time – hash– file name)
ed2k://|file|my 14 yr old sister bathing lolita qwerty ddoggprn reelkiddymov preteen tits nipples pussy.jpg|46028|4c2260e1c6fa89f7efab48b253a0d273|/|sources,86.218.146.235:5551|/ (eDonkey® link, This data can be used to download the file with a client to be sure the file is illegal)
ed2k://|server|83.233.30.126|4500|/ (The eDonkey® server name where the information was obtained from)

- Identify the Internet user and launch further forensic investigations : the identification of the net-user is made on a classical way: question the Internet provider (ISP) which give to the investigator the client's name and address from IP address and date/time of connection. It must be then checked to the client's if the facts are real. This forensic operation consist in searching for files in the computer which was connected at the transfer time, and in the various computing media (USB Key, external Hard Disks, Removable media, other systems,...) that can be found at his place.

This technical processing is generally accompanied with a forensic analysis of the confiscated computing objects in order to restore deleted data or files.
The investigation aims to define the reality of the facts, the motives of the suspect, eventually

correlations with identical facts, child abuses committed around him.
This processing leaded on several occasions to identifications of child abuser suspects with a picture on a P2P network.
Using identification of pedophiles on Internet in this kind of protocols demonstrates the relation between activities on Internet and in real life.

Conclusion
We have presented a work aimed at developing a tool to monitor pedophile activity in eDonkey®, a typical and important P2P system. This tool makes it possible to gain deep insight on this activity, and is of great help for law enforcement investigation, as illustrated.
Of course, similar work may be conducted on other P2P networks, each with its specific technical features. The ones on which pedophile activity has been evidenced are of prime interest for us. The approach we have developed here may be reused in these contexts, as well as other contexts like IRC and web measurements.
It must be clear however that the investigator personal work is of prime importance, in particular when there is some kind of interaction between the pedophile and his/her victims. That is true in particular for chat systems and anonymous networks. On chats, the point is to make a convicting contact with the predator in order to succeed in identifying him.
Still, better knowledge and mastering of the underlying technical difficulties is extremely important. Software tools are a key resource for investigators in this context, and should be developed at a much wider scale.
Using such tools releases the investigator from the technical part and keeps him free in his/her investigation work. This is true for IRC investigations too, as a software may perform a preliminary search for keywords in thousands of chat rooms and identify rooms of special interest. More subtle language analysis is also possible.
The next key challenge for fighting pedophile activity on the Internet is certainly the emergence of anonymous networks. Anonymization technologies do exist and become more and more widely accessible. It makes no doubt that it will be more and more present P2P and chat networks.
In this context, uncovering the identity of predators is an extremely challenging task for investigators. Handling this will probably require deep technological skills and using appropriate undercover techniques. In other words, investigators will have to show their credential in order to locate the predator.
More globally, collaboration between researchers and law enforcement investigators is extremely important, promising and has to be encouraged. It provides interesting, challenging, and motivating questions to researchers, with a deep societal impact. It enhances technical skills of investigators, and provide them with advanced tools for their work.

Acknowledgements. This work is supported in part by the European MAPAP SIP-2006-PP-221003 project. See http://antipaedo.lip6.fr

Bibliography :
[1] Lauritsen A.K., Meldgaard K., Charles A.V., Medical Examination of Sexually Abused Children : Medico-Legal Value, Journal of Forensic Sciences, JFSCA 45(1) : 115-117, 2000.
[2] Grafeille J.-M. Et N., La pédophilie ou les maux d'enfants, Collection Vivre et Comprendre, Ellipses, ISBN 2-7298-5938-1, 1999.
[3] http://www.protocols.com/pbook/tcpip1.htm (TCP/IP Reference page)
[4] http://www.irc.org/ (Internet Relay Chat Web Site)
[5] http://www.w3.org/ (World Wide Web Consortium Website)
[6] Oliver Heckmann, Axel Bock, Andreas Mauthe, Ralf Steinmetz : The Edonkey File Sharing Network (http://subs.emis.de/LNI/Proceedings/Proceedings51/GI-Proceedings.51-50.pdf)
[7] Yoram Kulbak and Danny Bickson : The Emule Protocol Specification (http://www.cs.huji.ac.il/labs/danss/presentations/emule.pdf)

[8] Oussama Allali, Matthieu Latapy and Cl´emence Magnien : Measurement of *eDonkey* Activity with Distributed Honeypots (http://antipaedo.lip6.fr/Honeypots.pdf)
[9] Igor Ivkovic Software Architecture Group (SWAG) Department of Computer Science University of Waterloo Waterloo, Ontario N2L 3G1 Canada : Protocol Analysis And Research Proposals (http://www.cs.cornell.edu/people/egs/615/gnutella.pdf)