Peer-to-peer Networks: Security Analysis

J.Schäfer
Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
schafer@fit.vutbr.cz

K. Malinka
Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
malinka@fit.vutbr.cz

P. Hanáček
Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
hanacek@fit.vutbr.cz

Abstract

In this work we're dealing with security in highly distributed systems, specifically peer-to-peer networks. We are describing some known theoretical attacks and defenses in these kinds of networks and comparing them with real world data. Classification of attacks realizable in peer-to-peer networks is given. We also discuss influences of their combinations. This could be useful for creating models of peer-to-peer networks' defense and malware spreading. Also we are proposing our new system for automatic downloading and detection of new viruses in peer-to-peer networks, together with all possible extensions.

Key words: P2P, malware, behavior analysis, botnets, DoS.

1. Introduction

This work deals with security problems in decentralized peer-to-peer (P2P) networks, which are part of highly distributed systems. All pieces of knowledge which arises from this research are, sometimes in special manner, applicable to the other types of distributed systems. Discovered information can help us realize some security principles in larger scale, not only in terms of P2P networks.

P2P networks became very popular due to their contents. They contain wide variety of all possible data, including illegal stuff such as movies, MP3 songs etc. Altogether, you get highly dangerous network, which is used by millions of users, who are usually unaware of security and risks of using client software in peer-to-peer networks.

Clients are forced to use special protocols for communication and file downloading, because there is no central server in P2P networks. Smaller subnetworks emerge, in which clients are connected to each other.

Here we get very specific environment suitable for investigating security properties, e.g. specific spreading of malware, monitoring effects on common users or other misuse caused by different types of attacks.

Creating new viruses and worms in P2P networks is often simplified into finding error in specific communication protocol, but in general it is very similar to common environments. Direct misuse of communication protocols presents us with more interesting point of view (in terms of security). DDoS attack can serve us as an example – we will discuss this kind of attack based on impersonation later.

In this article, we define peer-to-peer networks; specify their usage and present basic security problems. The basic enumeration and analysis of attacks is given followed by attack examples from different groups.

The main intent is to verify properties of already known attacks on peer-to-peer networks. Most of these attacks are only theoretical, usually based on number of preconditions. We want to check their power on real world data, as well as implementation requirements.

As far as we know, there are no implementations of attacks in networks we are interested in (DC++). Thus, we choose few appropriate and interesting candidates for our own implementation and further analysis.

We decided to enlarge scope of this article to cover not only P2P networks security overview, but also problems of malware occurrence and spreading, because it is closely connected to other attacks, as we show later.

While primary objective of this article is categorization of malware and attack simulations, it is also shown that phase of gaining information about

infection spreading is very important. This information can be used to improve attacks implementation as well as design of defense against many types of attacks. Due to this fact, we present our system for automatic download and detection of new viruses in peer-to-peer networks, which helps us understand spreading of different types of malware, diffusion of different files and impact on users when infection appears.

In the first section, basic overview of P2P network types and their properties is presented. Individual threats are described in detail. In the second section, effectiveness and feasibility of some theoretical DDoS attacks are verified.

The third section is oriented to problems of viruses in P2P networks. Our system for automatic download and detection of new viruses in peer-to-peer networks is presented. The purpose of this system is getting precise information about state and behavior of P2P networks. Acquired data should show us structure of shared data from malware point of view, which will help us create empiric models of worm spreading.

Better understanding of new strategies of attackers, their methods and tools can be obtained, based on the results of security analysis of P2P networks. Next step of this process is developing an effective defense against these strategies.

2. State of the Art

Definition of P2P network is presented here. Also we describe differences and similarities of their types. Description of attacks on peer-to-peer networks is given.

2.1 Peer-to-peer networks

Peer-to-peer (P2P) [2], can be defined as sharing of computer resources and services among participants using direct exchange. P2P client can ensure direct information exchange, computing time and data sharing. Participant in P2P network acts as client and server simultaneously. For imagination how can be P2P network established see Figure 1.

We've divided some well-known P2P applications into these few groups (separated by usage):

- Cooperation: Geographically distributed teams use communication-based P2P services, e.g. Skype [3].
- Services: Can be moved to places where they are needed more. Distributed service architecture disburdens remote servers.
- Distributed computations: Idle computer resources can be used for greater benefit of whole P2P network, e.g. seti@home project [4].
- Agents: P2P networks enable dynamic merging of power of individual intelligent agents operating on nodes [5].

The most famous P2P networks due such, providing music and movie sharing. Main breakthrough was caused by centrally controlled Napster [6], later replaced by Gnutella and KaZaA, considered as fully decentralized and highly dynamic. There is no central authority in these "new" P2P networks. They are self-organized, with dynamically adjusted structure. Due to the lack of trusted managing authority, P2P represents a great security risk, especially during expansion [7].

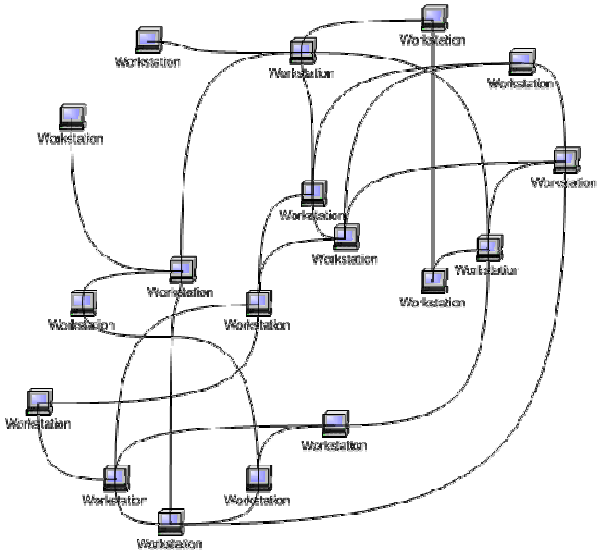

Figure 1 – Peer-to-peer network

Basics of the peer-to-peer file-sharing systems can be described like this: every connected user has its own folder, which contains files user wants to share. Anyone who wants to download file sends a request to all users in the network and then waits. Result contains list of results that matches search query. This list is generated in specific way for every type of network. After selecting one result, client sends request to download. The request is also specific for every peer-to-peer network.

During the file download phase, different approaches are used. One approach is to download from one specific source. Another approach could be downloading from more sources simultaneously. Here, P2P network must ensure content verification to prevent mixture of two different files with the same name. Files are typically downloaded to shared folder

for other users' disposal.

2.2 Attacks on P2P networks

Services are run by specific server or group of servers in standard client – server architecture. The attacker can take down, modify or counterfeit given service by successful attack on only one device. Ebay can serve us as an example: during successful DDoS attack on the main server, no visitor is able to use services of this internet auction centre. All services are closely connected to the server. Attacking the service is equal to attacking the server [7]. But not in P2P networks. Individual participants can be affected by the attack, but services are provided by more of them. So, there is no general effect on whole network. Successful attack on one supernode in Gnutella network does not affect accessibility of files. The only success can be obtained by shutting down the only client proposing specific file. Decentralized P2P networks spread services among all participants. This must be taken into account during security analysis of P2P networks.

Our classification of attacks connected to peer-to-peer networks can be found in Table 1. Selected attacks from table 1 will be discussed later.

Classification of these attacks seems to be a little bit inaccurate, because of their ambiguity. Some of them may belong to more groups than we mention. Nevertheless, the classification is based on the measurement of impact on the destination group (like peer-to-peer users or peer-to-peer network itself) – this means that attack is classified into the group where it can do most damage.

Type of attack	Attack Example
Attacks on Peer-to-peer network	• Listening queries • Filtering queries • P2P network disintegration
Attacks realized through peer-to-peer networks	• Malware spreading • DDoS attack • Setting up Botnets
Attacks on users of Peer-to-peer network	• Content Verification • Anonymity weakening • Stealing Identity

Table 1 - Classification of P2P network attacks

We can see attack summarization and attacks overview in Table 2. Some of these attacks are described later.

Attack name	Target
Leechers	Attack on networks reputation
Social attacks	Attack on users
Searching for sys. files	Attack on users-attack / on peer computer
Listening queries	Observation attack
DDoS attack	Attack on users/attack on peer/attack on other computer
Content verification	Attack on networks reputation
Attack using malware	Combined attack

Table 2 – Attacks overview

2.2.1 Content verification Genuineness verifying of downloaded file is mentioned here, despite it is not an attack at all. Every time we download a file, we must ensure that content of file corresponds to proposed file and doesn't include some unwanted part such as malware. In real world P2P networks, there is no mechanism to ensure this, with one exception – good will of users, which is usually missing [7].

In [8], Jian Liang determines number of fakes in KaZaA. He implemented mechanism for downloading of all music titles, which correspond to latest trends. During analysis of these musical files, he found out that 70% of ones that containing most widespread title "Naughty Song" was depreciated or were fakes.

2.2.2 Listening queries This attack utilizes open architecture of Gnutella network. Dependence on third parties makes it vulnerable to malicious behavior. But if we look at these problems from the perspective of attacking services (not attacking servers), we move to quite different area. In Gnutella network, interconnecting nodes are able to see significant part of queries from all servers in their local sub graph. How much damage can these nodes inflict if they behave badly? Each super node (node with broadband permanent connection to the internet dedicated to routing messages and keeping list of shared files of his sub nodes) can see crucial amount of communication taking place in its sub graph. Gnutella uses 7-jump searching protocol, so every query goes through the whole network via 6 super nodes. If each super node knows about four other super nodes, then it's possible for 1300 super nodes to see this query. In fact, Gnutella architecture is similar to Ethernet broadcast with more than 1300 nodes able to respond to any query. We don't know exactly how serious trouble can be caused by just one node, which is able to eavesdrop this amount of queries and respond to them accordingly to

its will, but it is obvious that in case of compromising such node, anonymity of Gnutella users would go down considerably [7]. Nevertheless, this kind of attack can be carried out in most types of peer-to-peer networks.

2.2.3 Leechers Leechers are P2P network users, who don't use service for file sharing and just downloads data. This type of users does not participate in network's data redundancy and therefore they're usually banned and kicked out of the network.
There are some techniques that could be used to produce enough data to fulfill sharing limits, fake sharing (sharing of non-existing files, modification of communication protocol and another techniques to use P2P network for free [1, 7].

2.2.4 Social attacks P2P networks are mostly used by users with limited knowledge of computer security. Their computers and accounts could be attacked by advanced users mostly by chat service of P2P clients. Attackers can misuse demands of these beginner users for a help to obtaining sensitive data. Typical attacks lead to sharing of whole system drive or to leakage of password and other sensitive data.

2.2.5 Searching for system files P2P users sometimes unintentional share all of his hard drive including operating, system files, application files, registers, private documents and other sensitive data. Some users share this data intentionally to extend amount of shared files to fulfill the rules of some P2P networks. This attack can be also joined with the social attacks – advances users could suggest victims to share their sensitive data.

2.2.6 Attacks using malware Many of described attacks can be combined with malware, for example Content verification with attack using malware.

2.2.7 DDoS attack This kind of attack is very well known, because no defense against this attack exists so far. P2P networks are not an exception. In flooded DDoS (Distributed Denial of Service) attacker abuses a lot of P2P network users, called zombies (see Figure 2-- description of Distributed Denial of Service attack).

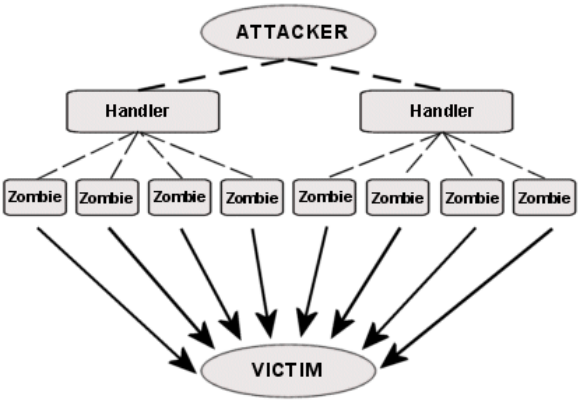

Figure 2 – DDoS Attack

These zombies can send bogus packets to specified target. The main goal of this attack is to exhaust victim's resources, so he is not able to provide or use some services anymore. The key resources are network bandwidth, network latency and TCP resources. From the attacker's point of view, successful attack is not just about exhausting victim's resources, but to use large amount of zombies too. Due to its properties is this attack difficult to detect as well as prevent. In this section we will discuss two main classes of flooding DDoS attack [9].

First one is TCP connection DDoS attack. Main goal of this attack is to exhaust victim's resources by many fully open TCP connections. When normal user tries to use the service, there are no TCP resources left [9].

Second type of attack is called bandwidth attack. In this kind of attack, attacker tries to generate huge amount of packets to overload target's bandwidth. In this kind of attack, UDP, TCP SYN or ICMP packets are used. This attack can be carried out by index poisoning or routing table poisoning [9].

In index poisoning attack attacker inserts fake records into P2P indexing system. These fake records say that target shares very popular and desirable files. The main idea is that the victim has not to be a participant of any P2P network, because each owner of shared files is addressable by his IP address, so the victim can be any mail server, web server or just user desktop. When normal P2P users start searching for these highly wanted files, faked indexes point them to the victim's computer. Then these deluded users try to negotiate download of these popular files, therefore they establish fully open TCP connection and exhaust victim's TCP resources or reach maximum of simultaneously opened TCP connections.

In routing table poisoning attack, attacker tries to insert fake records into routing tables of P2P network. Attacker tries to convince all users that the victim is their neighbor. When poisoned user tries to send a message (for maintaining connection or request query), he chooses a neighbor from his routing table. This is the point when he can choose the victim instead of real neighbor. If we imagine these networks can have about millions of users, even if only a part of it was infected, the communication routed through the victim could lead to the bandwidth DDoS attack [9].

3. Attack simulation

We have decided to implement and simulate only two attacks mentioned above. DDoS attack realized using native DC++ client and Listening queries attack [9], realized using modified client of DC++ networks. All simulations were run in laboratory conditions.

We are at the very beginning of a research on peer-to-peer network attacks; therefore the dimension of simulations is relatively small. First, we need to monitor targeted area, gain more experience with these kinds of attacks and finally, based on acquired data, choose more appropriate candidates for future research.

There is no confrontation of our results with different research groups, because we were unable to find any similar attack implementation.

3.1 DDoS Attack

The goal of this simulation was to implement DDoS attack in DC++ network and check the results of this attack on real data.

3.1.1 Attack resources Network of 20 virtual computers connected to P2P network DirectConnect++ was used for this attack simulation, together with DC++ hub. Hub acts as a supernode used by other nodes to forward their communication. Client programs, operating on nodes, were randomly selected from freely available clients for DirectConnect++, namely CZDC++, StrongDC, DC++. Opendchub version 0.7.15 served as the hub.

Ratio of active nodes to passive nodes was 20:80. By passive node we mean a node without public IP address, all of its communication is forwarded by the supernode. On the other hand, by active node we mean a node with public IP address. Its communication with other nodes is partly direct and partly forwarded by supernode.

3.1.2 Attack description We simulated hijack queries attack by automatic download of non-existing file provided by target client (the one we attack). This state was simply reached by deleting the shared file. Resulting scenario is similar to real attack. Attacker responds to queries. In reply, address of the desired file is substituted by address of target machine, which is obviously not possessing desired file (simulated by file deletion).

In next phase, the group of clients automatically sending requests for non-existing file download was created. The target machine responded by non-existing file error message. This kind of attack was unsuccessful due to the small number of attacking clients. The limited number of clients was caused by laboratory environment. We are working on creating a bigger network with complying parameters, which would allow us to simulate a successful attack.

Different approach to this attack brings us more interesting results. *Force attempt* technique (provided by overwhelming majority of DC++ clients) leads to exhaustion of client resources important for sharing with relatively small number (12) of attacking clients.

Result of successful attack is simple: no one can obtain any other files from target client. In extreme case, communication is affected mutually and target client cannot obtain any files from the rest of participants. This happens when target client is in passive mode – public IP address of this node is not allocated, all communication is forwarded by dedicated node. Taking into account all consequences of the successful attack, reaction of other participants must be expected. There is a high probability of target machine exclusion from P2P network due to constant failures resulting from other nodes attempting to acquire some data.

3.1.3 Results evaluation After detailed problem analysis, we have shown that only a few passive clients (12) were able to break client's functionality. If we want to fully exhaust victim's TCP resources, we should use active clients. It is because of active clients' connection type; active clients are connected via victim's socket server, so when lots of clients try to download non-existing files, victim must fully open this connection and answer that this file is not available. And even after victim sends this error message, previous established connection remains open. This is the easiest way to carry out DDoS attack. This attack was realized with real data and with minimal costs. We

have proved that this kind of attack is very easy to carry out.

3.2 Listening queries

Most users of P2P networks use pseudonyms and only a few of them are recognizable by their IP address. That is why we tried to determine users name or ID by listening to queries, search requests and other communication between P2P users. For this reason we have created a special tool, which is able to connect into DC++ P2P network [10], and act like a normal P2P user. Then we logged and analyzed the communication going through our program.

After analyzing these data, we were able to find out what specific users are searching, what they are downloading and who they are talking to. After detailed analysis, we were able to determine what kind of person it is, what are his hobbies and who is he talking to and which group of people he belongs to.

P2P network DirectConnect++ has slightly different structure than other P2P networks, so we were able to gather communication from only about 9500 users from one hub (hub is something like supernode in other P2P networks). So it is not the true attack on P2P network, but it is an attack which can lower the anonymity of users in this kind of P2P network. Though P2P networks do not guarantee anonymity, most of the users use some kind of pseudonyms and try to conceal their true identity. That is why the possibilities of user privacy compromise must be taken into account.

We have proved that even if user uses pseudonyms there are techniques that can help us to reveal his true identity.

4. Malware behavior analysis in P2P networks

This section deals with malware occurrence in P2P networks, especially with the possibility of compromising particular nodes and further exploitation. Attackers are trying to compromise more computers, which can be later use for further infection. They acquire control of these machines by using specific malware.

If we know malware true behavior in peer-to-peer network (propagation model, speed and impact on users), we can predict every possible consequence and lower the impact in case of real infection.

4.1 Botnets

Recently we are experiencing change in a way how attackers compromise some systems: wide spread worms, which infect hundreds or thousands of machines (similar to CodeRed [11] or Slammer [12]), are rather rare. This behavior can be caused by two main reasons. First, worms do not offer the attacker any means of remotely controlling attacked system – once the worm spreads, attacker cannot redirect the attack or even add some additional commands to worm. Second, attacker gains no financial benefit from releasing the worm. Ten years ago, most attackers were motivated by technical challenges or by effort of proving vulnerabilities, today most of attacks are motivated by money. Botnets are currently one of the serious internet problems [12].

Botnets can be defined as networks of compromised computers, which can be remotely controlled by the attacker. Every compromised machine (called bot) has a special program installed, which is remotely controlled by the attacker. Typical examples of these „remotely controlled networks“ are IRC networks and http servers. Few years ago, botnets based on P2P networks appeared. These botnets can be used to perform malicious activities, e.g. DDoS attack, sending spam, phishing, stealing important data or further spreading some malware [14][15].

4.2 Worm spreading in P2P networks

In order to develop the countermeasures, we are interested in model of malware and worms spreading. Studying worms in the phase of propagation is important for various reasons. First, warning systems capable of detecting worms can be created and (ideally) preliminary analysis of propagation can be given. No such system exists presently and it will take a while to deploy one. Second interesting aspect is threat analysis according to spread rate and number of hosts which can be infected by the worm. Last, but not least, we can stop quick establishing of large botnets by appropriate filtering out the worms.

Because P2P networks already have an established structure, these worms do not have to search for new victims by scanning (e.g. random scanning). Also these worms do not make large number of unsuccessful connections and their communication can be integrated into other ongoing communication. In [16], Hiestand showed that detection systems based on worm analysis

are not able to distinguish worm communication from communication of other subjects [17].

According to the way of propagation, we can divide worms into two categories:

- P2P worm using topological scanning: In this case worm takes advantage of information obtained from victim about his neighbors. This strategy can significantly speed up worms spreading, because he does not have to be bothered by scanning his possible victims. To improve this way of spreading, worms might use so called "hit-list" (list of victims). In case of collecting addresses before releasing, the worm gains more time in the early stage of attack. Once user is infected, he becomes involved in spreading the infection among users from hit-list, until the list is eventually empty. But no such worm appeared so far.
- P2P worm using passive scanning: These correspond to the current type of P2P worms. They do not actively search for victims, the just stay in shared folder on the infected machine. Being downloaded by another user, they begin to replicate and infect more shared files. As an example, we can mention worm Benjamin. Gunman worm works in slightly different way – he positively responds to all search queries by renaming infected file according to the query. When client downloads and runs this file, worm infects files of unsuspecting client [17] [18].

4.3 System for automatic file downloading from P2P networks

It is crucial to find out empiric model of malware behavior in P2P networks, as we proposed in previous section. It is important to find these models because precautionary measures and detections systems can be built on these empiric models. These precautionary measures and detection systems can help us with detecting of new malware spreading. These systems, which are able to analyze future propagation of spreading malware, do not exist, so far. The first part of creating on this system is to create empiric model of malware spreading. For creating such model we need a system, which allows us to access a large amount of data in P2P networks.

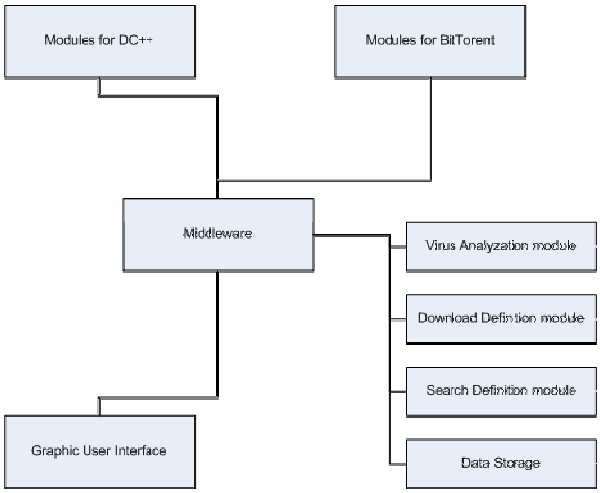

Figure 3 – System architecture

We need a system, which can tell us what each participant of P2P network is sharing, and system, which can help us analyze these data. That is why we designed a system, which can access a BitTorrent and DC++ P2P network, analyze traffic in these networks and can access the data shared in these networks. Most of this system is already implemented. System's design is described on Figure 3.

In this system, BitTorrent module cooperates with internet BitTorrent search engines, downloads the newest torrents, analyzes them and finally decides if it is desirable to download and analyze these files or not. This system also works in DC++, where it analyzes the search results and compares file hashes to decide if it is desirable to download this file or not.

The whole system is based on special architecture, which allows us to join two different P2P networks and gather search result from both of them. Then it can compare them (in case of positive infection in one network, we are able to search for similar files in other network). This system is based on modular approach, so virus analysis is similar for both types of networks and we are able to gather results based on the same metrics. Finally, these results can be afterwards discussed.

This system can be described as a group of cooperating independent modules, which are strictly specialized. System consists of a few groups of modules: module for communication with P2P networks, module for download and search definition, module for virus analysis, module for communication analysis and module for maintaining communication between other modules.

Individual modules are implemented independently.

They work on different platforms in different network due to proposed architecture. Modules communicate via own adaptive communication protocol transported by Middleware. Protocol enables special routing which allows duplicate work of modules belonging to the same functional group and mutual redundancy in case that one module disconnects from the network.

We have implemented only modules for two P2P networks so far, but this system is not limited only to these. System is highly scalable and these two networks were implemented first, because there are open source clients for them, but other networks will be implemented soon.

Thanks to this system we are able to download and analyze the newest files available in P2P networks. This kind of data is very valuable for building empiric models of malware spreading. For example, when we run into some infected file while randomly analyzing some new files, system notices it and starts a full analysis of this file, spreads this file metadata through all connected P2P networks and tries to find out the level of diffusion of the file across these networks. Then system does this diffuse analysis regularly. Thanks to this approach we can acquire specific information about the speed of file spreading, number of users involved and how long this file remains on infected users' computers. With this information we can adequately design and build an empiric model of file diffusion in peer-to-peer network. Obtaining empiric models for different kinds of P2P networks and different kinds of files is very difficult, because we need lots of data. That is why we can spread some fake files (with very popular names) by ourselves and then gather results by observing spreading of these files.

We present only basics of this system in this article, because we do not possess enough result data yet. But we have already gathered some data that can lead us to more promising results.

5. Conclusion

In this article, we dealt with relatively well-known questions of network security in relatively less common environment of distributed networks, particularly DC++. We described problems of P2P networks security with focus on particular attacks.

We have successfully proved that some attacks on peer-to-peer networks, more precisely on peer-to-peer users, can be carried out with minimal efforts and price. We were able to simulate some well-known attacks with very good results in real networks. It proved great vulnerability and low resistance of these networks. For example, it is possible to deny the access to service by DDoS attack even with small number of attacking machines. There is no need to use large botnet to carry out DDoS attack in peer-to-peer network DC++, all you need is just a few users participating in this network. We have proved that some of theoretically designed attacks can be realized very easily and that they are very effective against peer-to-peer network users.

New questions, related to the privacy of P2P network users, were opened during our research. We have pointed out that this issue has not been sufficiently discussed, for example previously proposed easily realizable DDoS attacks, or listening queries attacks, which leads to lowering the anonymity mainly there, where users from social networks, share data, but try to stay anonymous (hiding behind nicknames). We showed that it is quite easy to collect data about communicating entities and get enough data to be able to make judgments about user's identity and behavior.

We have proposed basic categorization of attack on peer-to-peer networks in this article and we have shown some basic attacks and their analysis and evaluation. We have separated attacks into a few groups, attack realizable thru peer-to-peer network, attacks on peer-to-peer network and attack on peer-to-peer users and we have detail described each kind of attacks.

In last section we presented our tool for automated file download. We showed its basic structure, its possibilities and its worth for creating and verifying empirical models of worm propagation in P2P networks. This area of research gives many (still open) questions, which we want to devote to in our future work.

This research was supported by the Research Plan No. MSM, 0021630528 -- Security-Oriented Research in Information Technology.

6. References

[1] Schafer Jiri, Malinka Kamil, Hanáček Petr: Peer-to-peer networks security, In: The Third International Conference on Internet Monitoring and Protection, Bucharest, RO, IEEE CS, 2008, s. 13-13, ISBN 978-0-7695-3189-2

[2] N. Minar, M. Hedlund, C. Shirky, and others. Peer-to-Peer: Harnessing the Power of Disruptive Technologies, O'Reilly, March 2001.

[3] Skype Limited. Skype. http://www.skype.com/, May 2008.

[4] Seti@home, http://setiathome.berkeley.edu/, May 2008.

[5] D. DeFigueiredo, A. Garcia, and B. Kramer. Analysis of peer-to-peer network security using gnutella. Technical report, University of California at Davis, University of California at Berkeley, National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, April 2002.

[6] LLC Napster. Napster. http://free.napster.com/, May 2008.

[7] V. lachos, S. Androutsellis-Theotokis, and D. Spinellis. Security applications of peer-to-peer networks. Technical Report 2, New York, NY, USA, 2004.

[8] Y. Xi K. Ross J. Liang, R. Kumar. Pollution in p2p file sharing systems, 2005.

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

[10] DCFORGE. DirectConnect++, http://www.dcforge.com/, May 2008.

[11] Steve Friedl. Analysis of the new "Code Red II" Variant. http://www.unixwiz.net/techtips/CodeRedII.html, May 2008.

[12] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver. Inside the Slammer Worm. IEEE Educational Activities Department, 2003.

[13] T. Cymru. The underground economy: Priceless. Technical report, ;Login: vol.31, no.6, December 2006.

[14] J. Goebel, T. Holz, and C. Willems. Measurement and analysis of autonomous spreading malware in a university environment. In Bernhard M. H´nmmerli and R. Sommer, editors, DIMVA, volume 4579 of Lecture Notes in Computer Science, pages 109–128. Springer, 2007.

[15] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-peer botnets: overview and case study. In HotBots’07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, pages 1–1, Berkeley, CA, USA, 2007. USENIX Association.

[16] C. Göldi and R. Hiestand. Scan detection based identification of worm-infected hosts. Technical report, Institut fur Technische Informatik und Kommunikationsnetze, April 2005.

[17] Wagner, T. Dübendorfer, B. Plattner, and R. Hiestand. Experiences with worm propagation simulations. In WORM ’03: Proceedings of the 2003 ACM workshop on Rapid malcode, pages 34–41, New York, NY, USA, 2003. ACM.

[18] N. Khiat, Y. Carlinet, and N. Agoulmine. The emerging threat of peer-to-peer worms. Technical report, University of Evry, France, September 2006.