ULBS
Universitatea “Lucian Blaga” din Sibiu

Facultatea de Inginerie Hermann Oberth
Master-Program “Embedded Systems“
Advanced Computer Communications
Summer Semester 2014

Security in peer-to-peer networks

Professor: Valentin Lup

Masterand: Stefan Feilmeier

TABLE OF CONTENTS

1 Introduction 3
2 Finding peers 3
2.1 Central Server/Directory 3
2.2 Query Flooding 4
2.3 Distributed Hash Table (DHT) 4
2.4 DNS Seeding 4
3 Vulnerabilities and possible Solutions 4
3.1 (Distributed) Denial of Service 5
3.1.1 Attack 5
3.1.2 Solution 5
3.2 Man in the Middle 5
3.2.1 Attack 5
3.2.2 Solution 6
3.3 Sybil Attack 6
3.3.1 Attack 6
3.3.2 Solution 6
3.4 P2P Worm 6
3.4.1 Attack 7
3.4.2 Solution 7
3.5 Rational Attack 7
3.5.1 Attack 7
3.5.2 Solution 7
4 Conclusions 8
Sources 8

1 Introduction

As the name suggests, in a “peer-to-peer network” two or more computers, named peers, are interconnected with each other over a network like the Internet. In contrast to a traditional client-server network approach, those peers are treated equally and are able to spontaneously collaborate directly, without the need for central coordination.

The “peer-to-peer” (P2P) network structure promises to improve scalability, to lower the cost of ownership, offers self-organized and decentralized coordination of previously underused or limited resources, greater fault tolerance and better support for building ad-hoc networks. In addition, P2P networks also allow totally new usage scenarios compared to traditional networks (Schoder, Fischbach und Schmitt 2005).

The ideas of P2P started to be used already in the 1960s with ARPANET and continued later with USENET, but only grew to be really widely used by the year 1999, when millions of people on the Internet were using the file-sharing application Napster. Since then a variety of applications and protocols adopted the P2P concept for content delivery, file-sharing, multimedia streaming with distributed load balancing, distributed backup systems and more. Among the best known is the BitTorrent file-sharing protocol, but also new developments like the Bitcoin digital currency are based on P2P principles (Peer-to-peer 2014).

Because of its decentralization and the fact that peers are unregulated and operated by strangers, P2P networks are also susceptible to vulnerabilities and security frauds. In other words: “In a large and open domain, such as the internet, it is almost a certainty that malicious nodes will be joining the network” (Engle und Khan 2006).

Vulnerabilities of peer-to-peer networks that might be used for attacks and possible solutions shall be discussed in this paper.

2 Finding peers

To understand the vulnerabilities of peer-to-peer networks, it is important to understand the internal functionality. As P2P systems try to avoid any fixed server structures, one of the major issues for any system is the discovery of peers and resources in the network (Li 2007).

2.1 Central Server/Directory

The simplest solution to this problem is a centralized directory server which is indexing all resources and is queried by peers that are searching for those resources.

While the process of looking up other peers is fulfilled using this client-server architecture, the actual transmission of resources is done directly between the peers. The problem of the fixed servers in this concept is, that they are the central points of failure in the network.

Because of its simple and straight forward implementation, this approach was used by Napster and is still used – together with other approaches – for the BitTorrent protocol.

2.2 Query Flooding

To overcome the dependency on central servers, “query flooding” was developed. The theory is, that instead of looking up resources on a directory server, the query is being sent directly to the network as a broadcast. Peers which have the desired resource are asked to respond accordingly.

While there are optimized variants of this approach that select certain peers with high availability and high capacity as “supernodes”, the main problem of query flooding is a very high bandwidth usage which in itself may lead to an unintended self-distributed denial-of-service attack.

Query flooding was used by early versions of the popular Gnutella P2P network, but was replaced in the meantime by a variant close to distributed hash tables.

2.3 Distributed Hash Table (DHT)

To eliminate central points of failure in a network and to avoid excessive bandwidth usage, the “distributed hash table” approach was developed as a way to decentralize a central directory. The availability of resources is stored in hash tables, which are distributed among clusters of peers in the network.

The DHT approach is used in the trackerless BitTorrent protocol.

2.4 DNS Seeding

To find other peers on the Internet, a common method is “DNS seeding”. The idea is to maintain a list of host names like “dnsseed.bluematt.me” which, following to a DNS query, resolves to a list of IP addresses known to be available peers (BitcoinStats 2013). While the approach is still relying on a number of central DNS servers, the benefit is, that DNS was built to handle tens of thousands of connections and as such is easily scalable.

DNS seeding is used by Bitcoin.

3 Vulnerabilities and possible Solutions

“The peer-to-peer network design emerged from the motivation to realize a computing architecture which cannot be taken down by attacking any single point” (Engle und Khan 2006). While P2P clearly solved some vulnerabilities present in client-server environments, it also intensified old and introduced a number of new vulnerabilities.

3.1 (Distributed) Denial of Service

A “denial of service” (DoS) attack is an attempt to interrupt availability of a service or a network resource, making it unavailable to its intent users.

3.1.1 Attack

The most common form of an attack to cause DoS is to flood a network with invalid packets, eventually causing overload on the targeted peer and as such preventing valid communication requests to the peer from being handled. In P2P networks an attacker can make use of the way, the network is forwarding queries. For example in the case of a “query flooding” network, a straightforward attack is to simply send a massive number of queries to peers, resulting in a broadcast storm that will render portions of the network inoperable.

As a flooding attack by a single attacker is in general highly limited by the attackers own bandwidth and also easily detected and obviated, the more common approach is a “distributed denial of service” (DDoS) attack. A DDoS attack is classified by having many participating nodes which are often controlled by one attacker. This structure makes it difficult to be detected, as the attack is not coming from a single peer, but is simply looking rather similar to a high network utilization.

P2P networks can not only be victims of DDoS attacks, but more often are also used themselves to start DDoS attacks against other targets. This is done by poisoning the network with invalid data, which is followed by a high number of mislead requests against the target.

3.1.2 Solution

Due to the fact, that DDoS-attacks are not easily distinguishable from normal network usage, it is impossible to block them all. Still, there is a widely used method called “pricing”, which limits the speed at which nodes can make requests in a network. In order to be allowed to query a server-peer, a computational puzzle needs to be solved by the client-peer. These puzzles are designed to not interfere much with normal network usage, but to significantly slow down DDoS attempts.

3.2 Man in the Middle

As a “man in the middle” (MitM), an attacker is placing himself between two other peers in the network, forwarding, altering or reading the communications between them.

3.2.1 Attack

To start a MitM attack, the attacker needs to place himself in the route between two nodes, which is a complex task in traditional client-server architecture. However, this task can be fairly easy in most of nowadays P2P systems. Those usually do not have any control of the placement of the peers in their logical space, making them very vulnerable to this attack and in fact allowing an attacker to place himself in a very deterministic manner anywhere in the network he wishes.

Once there, the MitM can control all communications between the attacked peers. He may listen to messages or alter them and even create fake messages, assuming the identity of the peers. The attack itself is likely to stay undetected, as long as the attacker remains passive.

3.2.2 Solution

While "man in the middle" attacks cannot be avoided generally in a P2P structure, the main answer to the problem is, making an attack as worthless as possible. The most widely accepted solution is based on public and private key cryptography.

To prove authenticity of a message, the sender is creating a message signature with his private key. Any receiver of the message may then use the sender's public key to approve the fact that the message was in fact sent by the according sender and was not changed since.

To furthermore prevent a MitM from reading the content of the message, the sender may also encrypt the message with the receiver's public key, making it improbable for any other peer than the destination to read the contents.

3.3 Sybil Attack

In a "Sybil attack" a single user or malicious entity represents a large number of peers in the network, allowing him to exploit it similarly to a "Man in the Middle" attack, but on a P2P network level.

3.3.1 Attack

As previously discussed in the topic "Man in the Middle" (3.2), in a lot of recent P2P network structures, a peer can influence its own position within the network. A malicious entity can use this technique to create a certain number of peers and place them on strategically selected positions, taking over important routing tasks within the network. Using only a minimal number of peers can already inflict a large amount of damage to the network. Similarly to the MitM attack, it may then abuse its position to control all messages that pass through a segment of the P2P network.

The attacker may even extend to a larger scaled "Eclipse" attack, which is targeting to separate a P2P network into two or more partitions or even bringing down the P2P network in total.

3.3.2 Solution

Without a central server architecture, featuring a trusted authority, the possibility of a "Sybil attack" cannot be entirely eliminated. One way to decrease the vulnerability for this kind of attack, is the introduction of "pricing" similar to the one suggested for " (Distributed) Denial of Service" (3.1) attacks. To be allowed to join the P2P network, a certain puzzle would have to be solved, making it more difficult to create a lot of peers in a short amount of time.

3.4 P2P Worm

A "Worm" is a self-replacing computer program, designed to infect computers in order to collect all sorts of information from them or to take control of them.

3.4.1 Attack

While worms and other viruses or malware already cause big threats to traditional computer networks, their impact on P2P networks can be even worse. Most peers in the network are running the same software, making any vulnerability or bug in the software a threat for all connected peers. Where in traditional networks a worm had to scan the internet for vulnerable hosts, a P2P worm only needs to look at the list of connected peers to find new victims. For this reason, the worm spreads exponentially within the network.

The fact that peers usually have large network transfer capacities and are usually used as personal computers – housing valuable information like credit card data, account passwords, and so on – makes developing specifically for P2P networks even more popular among malware developers.

Furthermore a P2P worm can also be used to gather control of a peer and to use the P2P network itself as a tool. This method can be used to start a “distributed denial of service”, as previously discussed in section 3.1.

3.4.2 Solution

The preferred solution to the problem of P2P worms is the usage of perfect software without vulnerabilities and bugs. While this is obviously not possible, a lot can be done to improve software quality to an acceptable level, using well established software development methods like in-depth software testing, peer-review, strongly typed programming languages and so on.

An appropriate way to decrease vulnerability of a network to P2P worms is to enforce free and open standards and to publish the network protocol as well as the source code of the used software. Popular peer-to-peer networks prove, that in such an environment alternative software products are likely to come up, diversifying the network and making it less vulnerable to this kind of attack.

3.5 Rational Attack

The whole idea of P2P networks is based on peers generally cooperating with each other in a fair and efficient manner. If this cooperation is not enforced, humans “rational” nature may harm the network.

3.5.1 Attack

It is likely, that a user may change certain parameters in his peer’s software. This may be done to save bandwidth, especially upload bandwidth, which is heavily regulated by most internet service providers. Also legal issues may come into account, as in most networks it is easy to track peers that are sharing copyrighted material. For these and other reasons, a user may decide to restrict access to his content or not to contribute his resources to the network, and as such harm the network as whole.

3.5.2 Solution

To convince the human rational to allow access to content and resources and as such enforce the availability of the P2P network as a whole, several concepts have been applied already. One of those, initially introduced at Napster, is a reward system. Users were given a higher “rank” (connected with a “title”) in the community depending on the amount of content and resources they were offering to the network.

Samsara, a P2P backup system, enforces network fairness using an algorithm that ensures, that each peer may only use as much space on another peer, as it is giving to the network. This is similar to the idea behind BitTorrent, where the amount of data a node is willing to upload, affects the speed of the download.

4 Conclusions

After analysing the basic functionality and the main vulnerabilities of peer-to-peer networks it is clear, that a key problem for securing them is, that because of the inherent decentralized nature, they lack a central administration and as thus control, required to combat security attacks. Still, by using cryptography and other measures mentioned in the solutions parts of each topic, many frauds can be effectively eliminated.

It is clear, that P2P networks can be the solution for a lot of upcoming use-cases, but also the attacks will without any doubt become increasingly sophisticated. Therefor it is important, that designers and developers of P2P networks keep in mind that peers are unregulated and operated by strangers and that network security deserves to have a high prioritization.

Sources

BitcoinStats. *DNS Bootstrap servers.* 2013. http://bitcoinstats.com/network/dns-servers/ (Zugriff am 29. May 2014).

Engle, Marling, und Javed I. Khan. „Vulnerabilities of P2P Systems and a Critical Look at their Solutions.“ 01. 11 2006. http://www.medianet.kent.edu/techreports/TR2006-11-01-p2pvuln-EK.pdf (Zugriff am 28. May 2014).

Li, James. *A Survey of Peer-to-Peer Network Security Issues.* Dec 2007. http://grothoff.org/christian/teaching/2013/2194/li_security_survey.html (Zugriff am 28. May 2014).

Schoder, Detlev, Kai Fischbach, und Christian Schmitt. „Core Concepts in Peer-to-Peer Networking.“ 2005. http://www.econbiz.de/archiv1/2008/42151_concepts_peer-to-peer_networking.pdf (Zugriff am 28. May 2014).

Wikipedia contributors. *Peer-to-peer.* Wikipedia. 26. May 2014. http://de.wikipedia.org/w/index.php?title=Peer-to-Peer&oldid=130745393 (Zugriff am 26. May 2014).