A Distributed Denial-of-Service(DDoS) Attack using BitTorrent Peer-to-Peer(P2P) Network

Giovanni Branca
(nannibranca@libero.it)

Seminar "Internet Sicherheit"
Technische Universität Berlin

SS 2007/2008 (Version of May 30, 2008)

Abstract

Recently, many hackers attacks are done with a distributed denial of service (DDoS) strategy. Most famous centralized peer to peer (P2P) networks can be easily used for realize these attacks. File-sharing protocols such BitTorrent or an extension of Gnutella use centralized server for orchestrate 7-level connections between peers. This procedure create a point-of-failure because malicious centralized-server modifies can redirect peer connections toward a target machine on a specific port. Main attack's aim is exhausting machine resources, causing denial-of-service. However, are not generated big traffic peaks: are only setted big number of connections which are maintained alive until timeout. In this paper are analysed distributed denial-of-service attacks realized with a p2p file-sharing network by unconscious peers and possible methods of defence with their advantages and disadvantages. Popularity and constant growth of p2p networks mixed with serious difficulties to detect malevolent connections make these attacks very attackers attractive and so very worrying.

1 Introduction

The first peer to peer network developed is been Napster[1], in 1999. Original aim was share music content between users using the Internet. Napster's architecture is very simple: when a user want download a music file, such as a mp3, send a query request to a centralized server that return a list of users sharing content looked for. Napster became famous very fast but also is been fast closed because users shared copyrighted contents.

The next peer-to-peer system developed is Gnutella[2] protocol. Main difference from other P2P networks is that Gnutella doesn't need one or more centralized servers for give to peer control information to coordinate peer activities. The unique information useful to a Gnutella client is an address of a remote pc: it can discover other peers in the network asking to near peers which participants it knows. Query is so sent in flooding between all the network and client can have a network vision.

Successive evolutions of Gnutella[3], necessary for address problems of bottleneck implement a tiered system of ultrapeers and leaves. Nodes are not consider all equal: a node routing-capable is promoted to ultrapeer with features of accept leaf connections, route search and network maintenance messages. Developers's target is have a more

efficient and scalable use of network resources.Another similar protocol implementing idea of an architecture where all nodes aren't equal is FastTrack protocol[4]. A node with more intelligence is named supernode and can be compared to a directory server.

BitTorrent[5] is the most recent P2P protocol, created by Bram Cohen in 2001. It' s very efficient in transfer files because with its fairness mechanism force clients to participate actively. If only some file's pieces are already downloaded in correct manner, these can can be stand uplinked in the network.

Recent studies[6] confirm popularity of P2P networks, electing BitTorrent the most used P2P protocol. But BitTorrent has an internal vulnerability that can be used for malicious purposes. In BitTorrent architecture is present a central server knows as a tracker: it has important assignment of organize connections between peers that partecipate in an ad-hoc file sharing network, a swarm. The tracker of a swarm is indicate by original file distributor, and all peers in the swarm relies blindly of tracker. This is a crucial point for protocol security: a not honest file distributor can act with a modified tracker to redirecting peers connection traffic toward a target machine in the Internet, toward an arbitrary service or application port. A BT-driven attack don't require modifications in client software, is easy to be implemented and peers are unconsciously dragged. Effects of these attacks are conspicuous and worrying.

In this paper we analyse in the first place what is a DDoS attack and how can be realized take advantage of BitTorrent vulnerability or others centralized, and in a second place some defence methods. An hacker, or in this case an attacker, can easily use a popular file for make an application-level connection exhausting distributed denial-of-service (DDoS) attack using unconsciousness members of a BitTorrent swarm. This type of attack is easy to implement, efficient, and hard to defend. Redirecting a big number of peers to a target machine on a specific port has aim to exhausting network and computing resources of victim, making it out-of-service. In this manner is denied normal service to a legitimate host requires it.In literature there are only two researches done on overusing BitTorrent to attack an arbitrary target: a paper[7]studies vulnerability of BitTorrent and propose both a proactive and a passive solution, and another research project[8]focuses on exhausting target resources without generate a big amount of traffic.

2 DDoS attacks description

Definition 1 (DDoS attack) *A DoS (DDoS) attack is an attempt to prevent legitimate users of a service or network resource from accessing that service or resource. DoS attacks usually make use of software bugs to crash or freeze a service or resource, or bandwidth limits by making use of a flood attack to saturate all bandwidth. A distributed DoS is an attack concerning a big number of host coordinated by a central intelligence for hit an established target machine in the Internet.*

Today is very difficult have an exhaustive vision of all DDoS attacks. Given variety of knows attacks it seems that DDoS phenomenon is hard to explore and understand. In literature there is not a common accepted classification of DDoS attacks. Every researcher try to group attacks by some characteristics that he deems for his studies. We present a complex and a simple classification useful for have a general idea about all DDoS attacks and at some time a correct characterization of P2P DDoS attacks.

2.1 First classification

A recent study[9]tries to give a taxonomy of DDoS attacks grouping attacks for common characteristics. Most interesting groups are described below:

- Degree of automation
  Usually a DDoS attack is composed by 3 main phases: recruitment of a multiple machines, acquisition of a certain control level over these and attack launch. Each of these phases can be do in manual or in automatic way. Early attacks are in manual category, now all attacks are automatic or semi-automatic because they use a big number of hosts. An attacker can deploy automatic scanning and propagation techniques for recruit machines that take advantage of protocols or operating system security or applications bugs. Sometimes in hosts are located worms or Trojans that send a response to attacker when host receive a recruit message. Propagations mechanisms are also differentiated in central-source, back-chaining and autonomous modalities.
- Weaknesses
  This classification considers the weaknesses to deny service: are distinguished semantic and brute-force attacks. Semantic attacks want exploit some implementation bugs or specific features or applications running in victim machine with the aim of waste its resources. An example is TCP SYN attack, that we explain in next paper section. Brute-force attacks are realized by initiating many seemingly legitimate transactions. A big number of attack packet can exhaust victim's resources because core network can deliver an higher traffic volume than victim can handle.While a semantic attack can be repressed with a good protocol design, brute-force attack is more difficult to challenge because all packet appear belong to legitimate requests
- Attack rate dynamic
  During attack phase, each single participant send a certain amount of packet to target. Send-rate can be constant, usually as many as their resources permit. The effect is a rapid disrupt of victim's services. This method is very efficient for attackers: with a limited number of hosts can exhausts in few time victim resources. On the other hand, a simple server-side strategy could easily individuate IP addresses and reject ingoing traffic from these sources. Instead, a variable increasing rate want gradually degrade victim performances over a long period, delaying detection of the attack. At last, a fluctuating rate can follow victim's behaviour, avoiding detection. For example a pulsing attack is performed launching and aborting it periodically; if attacking machines are coordinated into groups so that one group is always active victim experiences continuous of denial-of-service, while are not notice prolonged anomalies into network.
- Victim type
  Victims can be of many types: the same host, by disabling its communication mechanism or overloading it or crashing (freezing, rebooting,ecc); could be also an application running on the target host. Not less important are attacks to network, infrastructures or resources. A typical network attack want consume incoming bandwidth of a targeted network with packets that have a destination address chosen from target network's address space. A infrastructure attack could have target some distributed services in the Internet such DNS, routing protocols; lastly, a resource attack can hit critical resources into victim's network such as router or a bottleneck link.
- Impact on the victim
  Consequences of a DDoS attack on victims are a complete denial of service, with some possibilities of recovery, or simply decrease victim's service performances. Irreversible attacks inflict permanent damages to hardware victims; paper's authors haven't found news on these type of attacks. If an attack is recoverable, victim can do it in automatic way (self-recoverable attacks) or by human intervention such as rebooting or reconfiguring machine. We must notice although degrading victim's

resources attack is less worrying, it can have economics consequences for victim: a certain percentage of clients unsatisfied for service could consequently change their service provider(actually the victim).

2.2 Second classification

Another simple classification[10] of DDoS attacks groups all in only four broad class:

1. attempts to "flood" a network, thereby preventing legitimate network traffic
2. attempts to disrupt connections between two machines, thereby preventing access to a service
3. attempts to prevent a particular individual from accessing a service
4. attempts to disrupt service to a specific system or person

Subject of this paper are attacks made using P2P network and can be well classified in the first class of last classification, whereas is difficult label them in the more complex taxonomy.

2.3 A real DDoS attack : the TCP SYN attack

An attack well-know to community is the SYN flooding method[11]. It can hit servers that runs TCP processes but has final target network's packet-processing capability and not TCP applications.

In many common applications like web-servers, a connection can be established without preliminary informations: server stay in listening-state on a specific port, waiting incoming clients requests. Initialization of TCP-connection process happens with three-way handshake protocol: client send a TCP segment with flag SYN setted, server respond with another segment with ACK and SYN flag setted and client in turn respond with a ACK-segment. So is created a full-complete connection identified with IP client and IP server address, client and server port application number named socket. In server-side or better in victim-side are allocated resources for manage every TCP SYN segment received.

A public server accept all incoming SYN segment and flooding attack relies on this behaviour, assuming that victim allocate state for every segment received. When server send SYN-ACK segment has built in its system memory a data structure for manage this not-confirmed connection. But a server has limited resources and so exist a maximum number of connections beyond which may exhaust memory, crash or be inoperative. In consequence will be rejected new incoming connection requestes. Normally a not-confirmed connection has a timeout associated for don't make use of resources for an infinite time. Besides connection will closed and so victim resources released. However attacking system may send persistently new SYN segments for maintain victim busy.

This method of attack is very common and popular. Also some vendors of networks components[12] advise defending strategies against this type of attacks.

2.4 A case study: Estonian attack

The Estonia, although small, is a remarkably web-dependent European country with many wirespread Internet access, an 80-percent usage rate for online banking, remote medical monitoring and so on[13]. Estonia parliament in 2000 declared Internet access a fundamental human right and Estonian are so proud of high technologies that call their country E-stonia.

On May 2006, Estonian country was victim of a big DDoS attack, began on foreign minister's website and in few time spread to all government institutions and businesses.

From a traffic measurement done during attack appear that most of attacks were Internet Control Message Protocol (ICMP), messages originated by ping request command. Maximum bandwidth used was 90 Mbps in 10 attacks lasting 10 hours or more. Usually DDoS attacks can keep busy greater bandwidth, but in this case was enough only 90 Mbps since that Estonia is a very small country and all systems are configured for little network and server loads.

The set of all these characteristics, mainly small network link bandwidth and services capacities together to Internet dependence lead up an entire country to paralyse. But must worrying fact is that Estonia was unable to counter the attack. The unique action was cut Internet connections to outside world so that people within Estonia could use newly services. Probably Estonia wasn't ready to count a similar attack, anyway DDoS attacks can have serious consequences not only for victim machines but also economical and social.

3 Use of BitTorrent for DDoS attack

BitTorrent is a famous peer-to-peer file sharing protocol. This protocols separates file searching function from P2P network, using network only for delivery contents. BT also differs from earlier P2P protocols because there isn't a single network but are created smaller ad-hoc networks for each file that is being transferred. Architecture of BitTorrent is composed from clients (or peers), trackers and seeders. Usually clients and seeders are home pc, instead trackers are big servers.

Now we analyse procedures for share own file contents in public BitTorrent network and procedures for download contents.

3.1 Share a file

When a client, a generic user, want share a file in the BitTorrent network must do some simple operations. First of all, from original content file it must generate a compliant .torrent file, which contains information about name of the file(s) to be shared, trackers to be used, hash code of file contents and so on. Then, this torrent file is published through public online channels: for example in a newsgroup, in a forum, in a chat or in specialized websites. But file is not still shared. For start shared session client informs tracker that it starts sharing file relative to .torrent file already generate and so it's named seeder. In this way, tracker server has a correspondence between a .torrent file and client where this files is stored.

3.2 Download a file

A generic user that want have a specific content in his pc in a first moment search .torrent files relatives to his desiderata content. Web-surfing in forum, chats, or in web sites can find, download a torrent file and add this to BitTorrent software client runs in own local pc. With informations contained in torrent file, client program can initialize a connection to tracker suggested, asking a list of other peers that are in this moment sharing file contents desired. In this list there are public IP address and port application number of peers : so client starts TCP connections to other peer and can request file pieces, named chunk, from different sources (peers). When download of a chunk is finished, it's hashed and checksum is compared to checksum given for the chunk in the torrent file. If checksum match, chunk download is considered complete and client can in turn send this piece to other peers will required.

Download process continue in a parallel way from many active members in a BT ad-hoc file sharing network, called swarm. Transfer speed rate increase as more peers

participate in the file swarm, so a popular file can be downloaded with an high rate. Now downlink and uplink bandwidth is fractioned between swarm participants, and this distribution system had demonstrate to be efficient and economical cost saving than a classical client-server model.

Until file download is in progress a client send to its tracker continuous informations: it announce itself, confirm participation in the swarm and obtain a refreshed lists of peers also participating in the same swarm. When a file is completely downloaded and all chunk checksums are verified, client announce itself to tracker as a seeder.

3.3 Communication paths

In the architecture of BT can be recognized two communication paths:a control path between every peer and its tracker, and a data path between peers. A peer it's free to create an independent number of connections with other peers, and each data connection is independent of all other data connections, so a peer is free to download chunks from any combination of swarm peers.

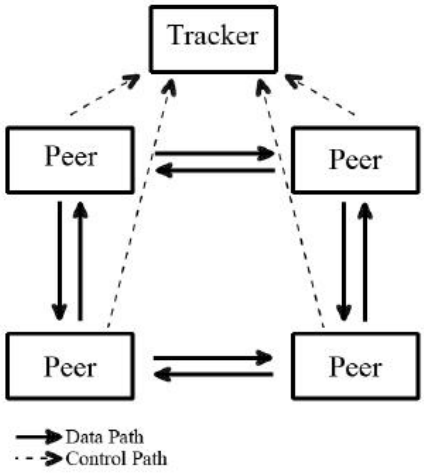

Figure 1

3.4 A BitTorrent-driven attack

A DDoS BitTorrent-driven attack can be realized in few step. First of all original file distributor, that is also the attacker, needs a server where is running tracker software. Peers rely on the response provided by tracker, so if an attacker has control of tracker server (or better tracker software) can alter response that tracker provide to peers in the swarm.

Tracker software must do two main tasks: maintain a local database of currently peers connected to swarm and create a peer list of these peers to a generic calling client. Alterations to tracker software can be done to return a new list which has both legitimate data and attacker-configurable addresses of target machines. These can be added on top, in tail or in a group but a more aggressive strategy can be insert these in a random way, so hardly a client take notice of anomalies. In fact a malicious connection don't transfer data content and it's normal that someone peer can't immediately send data requested, but is not much probable that a long subsequent list of contacted peers haven't data requested. With the ability of insert arbitrary IP address into peer list, a modified tracker

can redirect client connections to targets desired. A most astute attacker that want attack a machine with multiple IP addresses like a web farm, can set-up modified tracker with a set of IP addresses belong to same server.

In the second step, attacker would need to obtain a file(s) that is likely to generate high demand : a new film, a new videogame software. So it can generate a torrent file and register the torrent with modified tracker and subsequently upload it in a famous torrent directory, like a web-site.

From this moment the torrent file is available from anyone: a user can download it and join the swarm. Peers into the swarm will begin connecting both to true peers and to targets pointed by tracker.

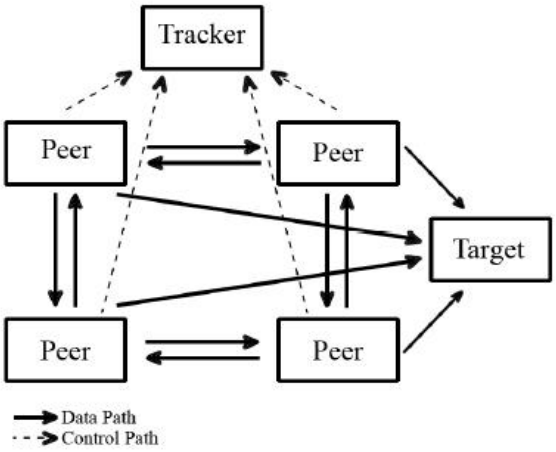

Figure 2

3.4.1 Attack evaluation and TCP stack saturation

In a small scale real world experiment[5] for study behaviour of peers in a swarm was used only a torrent file registered in a modified tracker. The torrent file was given to few people consciousness of experiment that was execute in 24 hour lifetime.

Results show a constant stream of new downloaders during first 6 hours, totally correlated to increasing number of attack connections. After this time, we can see a decreasing number of attack connections due to fact that peers that are joined early have completed their download and became seeding peers. So, those peers don't continue to create new connections to other peers in the swarm since they stopped seeking chunks.

From these results we can see that it's important that a swarm has a big number of peers remain constant during all time of attack. A successful attack can be done using a very popular content, and can be sufficient long if there is a constant number of peer requesting or if file take a long time for complete download.

It must be done also an other important consideration concerns effects of this BitTorrent-driven attack. When a peer connects to an other peer tries to open a new TCP connection. Target machine, usually a server, build in its memory a data structure for manage every incoming connection. A big number of incoming connections it means that target machine has to manage a big amount of data, consuming CPU and memory resources. If number of incoming connections increase we will see a reduction of target performances, until to limit that target will reject all new incoming TCP connections. These effect is very similar to classical TCP-SYN attack.

The most serious consequence it's that a BitTorrent-driven attack can exhausts not only a specific service on a specific application port in the target machine, but all services using TCP transport level. We can have a complete saturation of all TCP stack, with all consequences of this lockout on the target machine.

3.5 Another famous peer-to-peer network

In this section we would give some hints to another famous P2P network, Gnutella, that can be used for realize a DDoS attack. Unlike BitTorrent, Gnutella architecture is a very P2P system in sense that it has not a hierarchic architecture: all network members have same importance. When a peer want download a content, send a QUERY message that is forwarded on network. If another peer has contents requested, it respond with a QUERY-HIT message that contains its IP address and port number. So peer can contact and negotiate a TCP session using an HTTP/1.1 based mechanism, and all data transfer process is an HTTP transaction, identical to transaction between web-browser and web-servers.

As in BitTorrent peer blindly relies of informations given by tracker, in Gnutella there is not a central mechanism to verify trustworthiness of a query-hit. Malicious nodes can reply to any query message received with a query-hit which contains the IP address and the port number of any target machine connected to the Internet. The main effect is a server overload and if a big number of peers are requesting files, it can become a type of distributed DoS attack.

An evolution of this protocol, Gnutella2, has many features equal to BitTorrent. It introduce two different types of nodes, leaves and hubs: leaves are classical peers instead hubs are supernodes with an increased intelligent level. A supernode make same actions of tracker server and it's possible create a modified Gnutella supernode capable of reporting clients beyond an arbitrary machine that it's the target of an attack.

4 Possible methods of defence

In this section we analyse some useful and feasible defence methods against BitTorrent-driven attacks.

A first method [14]proposed use behavioural and probabilistic-based anomaly detection. To protect server is used a counter-mechanism that consist of a suspicion assignment mechanism and a DDoS-resilient scheduler. Suspicion mechanism assigns a continuous value to a session establishes on some session parameters like session inter-arrival time, request inter-arrival time or session workload-profile. On the other hand scheduler utilizes these values to determine if and when to schedule a session's requests according to some policies. This method is able to differentiate between legitimate and malicious session rapidly, intercepting requests belonging to malicious session before they overwhelm system resources.

Another method[5] of defence want improve security of BitTorrent protocol against this type of attack increasing intelligence of client software. For example, client behaviour could be modified in sense that each peer can connects once to target. If it receive an invalid response, client software can blacklist this target IP address and not attempt any further connections to the target. This method can be effective only if all or a big part of BitTorrent client software implements this feature. But BT protocol is an open protocol and today exists many client software implementations used by users. Besides, many user never will upgrade their software client if it works fine in file downloading.

5 Conclusion

In this paper we presented distributed denial-of-service(DDoS) attacks giving a brief description of general aspects and also describing in detail a possible BitTorrent-driven DDoS attack and analysing some methods of defence. This attack is easy to realize and have many attractive features for attackers:

- Don't require modifications of any BitTorrent client software
- Attack can have multiple target victims on a arbitrary service ports specified by the attacker
- Real compromised machine(the tracker) is not exposed to a victim that instead knows unaware peers
- Attackers can decided start and stop attack time controlling only the tracker
- Clients don't perceive attack's presence since they can download files normally and attack traffic is very limited

We can say that this type of attack(DDoS)in the last years is very popular. Also popularity of BitTorrent is growing and these two facts make us understand what can be serious and concrete threats of such attacks. To limit these facts, we must face the challenge of having security in a distributed system without a centralized control.

References

[1] `http://en.wikipedia.org/wiki/Napster`

[2] `http://en.wikipedia.org/wiki/Gnutella`

[3] `http://en.wikipedia.org/wiki/Gnutella2`

[4] `http://en.wikipedia.org/wiki/FastTrack`

[5] `http://en.wikipedia.org/wiki/Bittorrent`

[6] `http://www.ipoque.com/news_&_events/internet_studies/internet_study_2007`

[7] Ka Cheungh Sia: *DDoS Vulnerability Analisys of Bittorrent Protocol*; `http://oak.cs.ucla.edu/~sia/cs239spring06.pdf`June 2006.

[8] Jerome Harrington, Corey Kuwanoe, Cliff C.Zou: *A BitTorrent-Driven Distributed Denial-of-Service Attack*; Securecomm,2007.

[9] Jelena Mirkovic, Peter Reiher: *A Taxonomy of DDoS Attack and DDoS Defense Mechanisms*; `http://www.cis.udel.edu/~sunshine/publications/ccr.pdf` ,2004.

[10] `http://www.cert.org/tech_tips/denial_of_service.html`

[11] `http://tools.ietf.org/html/rfc4987`

[12] `http://www.cisco.com/en/US/tech/tk828/technologies_tech_note09186a00800f67d5.shtml`

[13] M.Lesk: *The New Front Line: Estonia under Cyberassault*; Security Privacy, IEEE Volume 5, Issue 4,Page(s):76 - 79,July-Aug. 2007

[14] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly: *DDoS-Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection*; Proc. INFOCOM, Barcelona, Spain, 2006