Misusing Kademlia Protocol to Perform DDoS Attacks

Jie Yu
School of Computer
National University of Defense Technology
Changsha 410073, China
yj@nudt.edu.cn

Zhoujun Li , Xiaoming Chen
School of Computer Science & Engineering
Beihang University
Beijing 100083, China
lizj@buaa.edu.cn, chenxm@buaa.edu.cn

***Abstract*—Kademlia-based DHT has been deployed in many P2P applications and it is reported that there are millions of simultaneous users in Kad network. For such a protocol that significantly involves so many peers, its robustness and security must be evaluated carefully. In this paper, we analyze the Kademlia protocol and identify several potential vulnerabilities. We classify potential attacks as three types: asymmetric attack, routing table reflection attack and index reflection attack. A limited real-world experiment was run on eMule and the results show that these attacks tie up bandwidth and TCP connection resources of victim. We analyze the results of our experiment in three aspects: the effect of DDoS attacks by misusing Kad in eMule, the comparison between asymmetric attack and routing table reflection attack, and the distribution of attacks. More large-scale DDoS attack can be performed by means of a little more effort. We introduce some methods to amplify the performance of attack and some strategies to evade detection. Finally, we further discuss several solutions for these DDoS attacks.**

Keywords- Kademlia; DDoS; P2P; Security

I. INTRODUCTION

Distributed denial of service (DDoS) attack is a technique that uses client/server model, combines lots of computers as an attack platform and launches at one or more victims (machines or networks)[1]. Traditional DDoS attacks involve two steps: first, breaking into a large number of computers using techniques such as virus, Trojan, buffer overflow, etc. and gaining a zombie network; second, sending a great deal of traffic to victims using zombie network and preventing them from offering service to their legitimate users. In most cases, the first step is the key to restrict the scale and performance of DDoS attack as more and more Internet users recognize the security of computer system and network.

Nowadays, P2P applications have been more and more popular and have lots of users. It is reported that P2P file sharing contributes more than 70% of the traffic in some areas [2]. P2P is characterized by peers in self-organized overlay network, which overlay the Internet Protocol (IP). It's possible to perform large scale DDoS attack by exploiting P2P protocol and application vulnerabilities, which doesn't need any compromised computers. Due to its easy operation, low cost, great performance and difficulty in defense, misusing P2P systems for DDoS attacks has been a new research hot in network security [3, 4, 5, 6, 7, 11, 15, and 16].

This paper is supported by National Natural Science Foundation of China (No.90604007, 90718017)

One of P2P's important characteristics is security. However, until now it is not covered in the protocols themselves but is just covered at the application level. Finally there is a low level of security [8], especially the DHT-based overlay protocols suffering from man-in-middle and Trojan attacks. Petar M. and David M. proposed a new DHT protocol in 2002, called as Kademlia protocol [9]. Comparing with other DHT technique such as Chord, CAN and Pastry, Kademlia-based DHT, called *Kad* for short, improves the performance of routing and searching. It mostly applies to P2P open source projects and has been deployed in Overnet, eMule, aMule, Bittorrent, etc. eMule is an open source P2P file sharing application and has large mount of users all over the world, especially in China. Nowadays, The Kad network in eMule connects over a million of simultaneous users [8]. aMule is the UNIX version of eMule. Bittorrent is one of the most popular P2P file-sharing applications. Kademlia protocol has been one of the most popular DHT protocols.

This paper focuses on how to exploit the vulnerabilities of Kademlia protocol in its design and implementation to launch DDoS attacks. We propose three attack methods: (1) *asymmetric attack* that misuses the difference in size or number of messages between request and response; (2) *routing table reflection attack* that sends spoofed routing messages to reflect traffic to victim; and (3) *index reflection attack* that sends spoofed index messages to reflect TCP connection requests to victim. We then identify them by real-world experiment on eMule. Furthermore, we discuss some possible approaches to improve the performance of attack and solutions to mitigate or defend against these attacks.

To the best of our knowledge, this is the first extensive study of Kademlia-based DDoS attacks against any victim host on Internet. The work in [4] is partly similar to ours, while it just experiments on Overnet and isn't from the view of the whole Kademlia protocol that is implemented in many P2P applications. Furthermore, we propose a new attack method – asymmetric attack, which is effective and easy to implement. [14] analyses the implementation of Kad in eMule and declares that it couldn't be exploited to launch active routing table reflection attack. Therefore, they perform DDoS attack by passively waiting for routing requests and responding with spoofed response messages. However, we discover another vulnerability in the implementation of Kad in eMule and verify that it indeed could be misused to launch active routing table reflection attack.

We find following results by experimenting on eMule:

- Above mentioned three attacks can be certainly performed and they tie up bandwidth and TCP connection resources of victim.
- Under same conditions, the effect of attacks is influenced mostly by the time the attacker starts. The effect at afternoon is best, and then is at evening, and the last is at wee hours. We have measured that the performance of attacks start at 13:00 is about 60% more than that start at 05:00.
- In eMule, file location information and routing information will live certain period. A single routing table reflection attack and a single index reflection attack will keep 1 hour and 5 hours respectively.
- Kad network has a kind of "memory" about the file location information and routing information as there is some attack traffic on victim even 6 hours after attacks are stopped.
- The effect of asymmetric attack ramps up rapidly (in 10 seconds) and then keeps as the same in the full attack period, however, the effect of routing table reflection attack begins with small and increases linearly in the first one hour, and then increases slowly in following attack duration. The effect of asymmetric attack disappears as soon as attacks stop while the effect of routing table reflection attack can maintain one hour even after attacks stop. Thus, the combination using of these two attacks might be much more sophisticated.

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 gives an overview of Kad. Section 4 presents the vulnerabilities in Kad that can be exploited to turn into a platform for launching DDoS attacks. Section 5 presents the results of real-life experiments on eMule. Section 6 discusses several solutions and we conclude in Section 7.

II. Related Work

Until now, there hasn't been any work that extensively researched on misusing Kademlia protocol as a DDoS engine. However, more and more works concern on P2P-based DDoS attack. K.Cheung Sia et.al.[5] analyses the vulnerability of Bittorrent protocol and implements DDoS attack by sending announcement messages to trackers which declare victim sharing certain resources; Harrington et.al.[12] perform DDoS attack by modifying trackers and replying every peer query with the victim's location information; In [7], the authors propose another method by declaring victim as trackers. They are all lack of analysis and simulation on DHT technique in Bittorrent. N. Naoumov et.al.[4] describe two approaches to create a DDoS engine out of a P2P system: the first involves poisoning the distributed index in the peers; the second involves poisoning the distributed routing tables in the peers; they experiment on Overnet. J. Liang et.al.[6] analyze the index Poisoning attack in detail and experiment on FastTrack and Overnet. E. Athanasopoulos et.al.[3] discuss DDoS attacks by misusing unstructured P2P systems and identify it on Gnutella. X. Sun et.al.[14] analyze the vulnerability of implement of Kad in eMule and ESM, and perform DDoS attack by passively waiting for routing requests and responding with spoofed response messages. They further discuss and simulate some methods to prevent such attacks in [15]. Y. Liu et.al.[11] propose a distributed and scalable method, DD-POLICE, to detect malicious nodes in order to defend unstructured P2P systems from overlay flooding-based DDoS attacks. J. Yu et.al.[10] build DDoS attack model in application layer and propose a defense mechanism by combination of detection technology and currency technology.

III. Overview of Kad

In this section, we give a brief overview of the Kad protocol, which emphasis on those parts that we later exploit to perform DDoS attacks.

Each Kad has a 128-bit ID. Kad computes the distance between two peers by XOR metric of their IDs. They communicate with each other using UDP messages.

- **Routing**

In Kad, when a peer K joins network, it sends BOOTSTRAP request messages to *n* (normally, *n* = 3) known peers. An alive peer which receives the message will respond it with a BOOTSTRAP response message. K then builds its own routing table and sends its location message – HELLO request message to all peers in its routing table. When a peer receives HELLO request message, it will add the location information to its routing table.

- **File sharing**

Then, K moves on to publish files information it is sharing. File publishing process consists of two steps as to convenient for searching and economize memory resources:

1. *Location information publishing*: First, K hashes each file in its file sharing list and obtains a 128-bit file identifier. Then it sends each file identifier and file location (IP and TCP port) to peers close to the file identifier. When peers receive this PUB_SOURCE request message, they update their local file indexes.
2. *Metadata information publishing*: First, K extracts keywords from each name of sharing files and hashes each keyword into a 128-bit key. Then it sends each key, file identifier and metadata information of the file (artist, size, type, etc) to peers close to the key. When peers receive this PUB_ KEYWORDS request message, they update their local keyword indexes.

- **File Searching**

In K, it hashes each keyword that user enters to search, and then send the key into Kad for iteratively searching. When a peer that has records for this keyword receives the message, it responds corresponding records to K. Each record contains file identifier and metadata information of the file. K then displays all matching file identifiers to user. After user selects certain file identifier *I*, K sends location search messages of *I* into Kad for iteratively searching. When a peer that has records for this file identifier receives the message, it responds corresponding records to K. Each record contains file location (IP and TCP port). K then tries to establish TCP connections with these IPs and downloads that file simultaneously.

IV. Vulnerabilities of Kad

Considering millions of simultaneous users in Kad network, we may try to exploit it as a large zombie network (shown in

Fig.1). It just need control peers of Kad at the overlay network layer, taking no account of compromising any computer system. Theoretically, if we use a network that has a million of simultaneous users as a DDoS engine, we may amplify the performance of attack one million times or even more. In this section, we analyze the vulnerabilities of Kad design and implementation and realize above expectation.

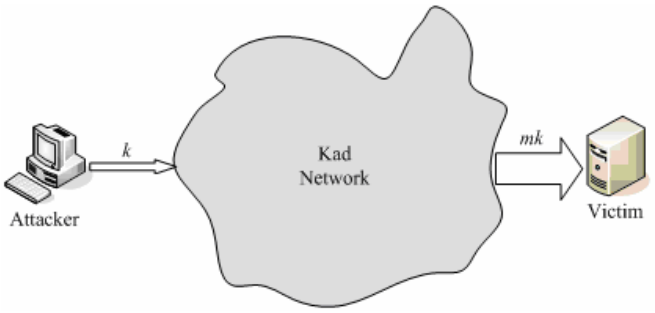

Figure 1. Misusing Kad network as a DDoS engine

Kad network is an overlay network, which overlay the transport layer. To communicate with each other exactly in Internet, when peers publish their BOOTSTRAP, HELLO and PUB_SOURCE request messages, transport layer information (IP address, TCP port, and UDP port) must be included in these messages. In the design of Kad, peers that receive above messages don't verify the location information of source peers and add them into routing tables or local indexes directly. If we declare location information in messages as victim's, some reflection attacks can be launched. The victim can be either peers in Kad network, or any host on Internet (especially Web server, Ftp server, Email sever, etc). We classify these attacks as three types:

- **Asymmetric attack:** Exploiting the difference size or number of messages between request and response, we can send smaller or less request messages and reflect larger or more response messages to victim. E.g. in eMule application, a BOOTSTRAP request message only needs 2 bytes, while a BOOTSTRAP response message that contains information of 20 peers needs 527 bytes. This message pair can enlarge the performance of attack more than 260 times. Compared with following two methods, this method is easy to implement and its power of attack could appear immediately.
- **Routing table reflection attack:** We announce to be neighbors of all peers in Kad, update their routing tables and redirect succedent request messages to victim. In order to implement this attack, we can send spoofed HELLO request messages to as many peers as possible in Kad. The source peer ID in each message is close to ID of target and source IP address and UDP port are replaced with victim's. Then, joining and searching messages to those IDs will be routed to victim.
- **Index reflection attack:** We declare to share some files, update local indexes of lots of peers in Kad, and redirect succedent connecting requests to victim. To implement this, we can send spoofed PUB_SOURCE request messages to as many peers as possible in Kad. The file identifiers in these messages identify some popular files and IP address and TCP port in these messages are replaced with victim's. Then, connecting requests of file downloading to those file identifiers will be routed to victim. If a certain service (such as Web server) is started on this TCP port of victim, full TCP-connection will be established until all connection resources are exhausted.

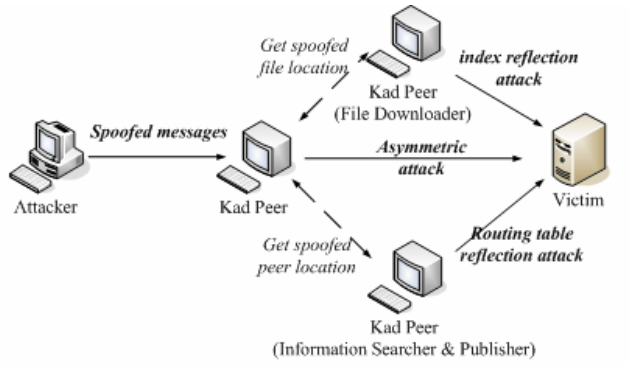

Figure 2. The principle of three attack methods

The principle of above three attack methods is shown in Fig.2. Spoofed-message receivers launch asymmetric attack directly; file downloaders in Kad launch index reflection attack and information searchers or publishers launch routing table reflection attack. The goal of asymmetric attack and routing table reflection attack is to tie up the bandwidth of the victim's access link, while the goal of index reflection attack is to prevent legitimate users from making connections to the victim. All of these methods are to hamper victim from offering service to their legitimate users.

At the end of this section, it should be noticed that these vulnerabilities can be exploited in all Kademlia-based DHT applications.

V. EXPERIMENT

To verify above analysis, we experiment on eMule [13] and show that it is indeed possible to misusing Kad to perform DDoS attacks.

There are two differences in the implement of Kad in eMule:

1. It doesn't include self IP address and UDP port in overlay messages. Peers extract source IP address and UDP port from received UDP messages. We change source IP address and UDP port at IP layer (It is supported in Windows 2000, Linux, etc) when sending spoofed messages.
2. It adds peer information into routing tables directly when receiving a HELLO_REQ message. However, Kad in eMule follows such strategy in that one peer checks information of each peer in its routing table periodically (1 minute) in regard to its lifetime by sending a HELLO_REQ message to the oldest unchecked one. If no response is received in 2 minutes, it will remove that peer information from routing table. Therefore information of each peer could exist 2 minutes at least. However, ***only checked peer information will be set***

alive and propagated to other peers. According to this strategy, the authors in [15] declare that Kad in eMule couldn't be exploited to launch active routing table reflection attack. However, we discover another vulnerability in its implementation: after a peer receive HELLO_REQ messages repeatedly (twice or more) from one same peer in 2 minutes, it will set this peer information alive and propagated it to other peers. Therefore, we overcome above difficulty by continuously sending HELLO_REQ messages to a peer ***twice*** with same source peer ID.

In addition, file location information and file metadata information will keep 5 and 24 hours respectively in eMule.

A. *Experiment Setup*

Our experiment consists of two parts: a crawler which takes charge of collecting information of Kad peers and an attacker which takes charge of publishing attack messages. To collect location information (peer ID, IP address, and UDP port) of peers, the crawler sends BOOTSTRAP_REQ messages to all peers and inserts responded information into our database. It works in multithreading manner. To start the crawler, we should input some initial peers and they could be obtained at the directory of eMule client (config/nodes.dat). The attacker circularly sends spoofed BOOTSTRAP_REQ, HELLO_REQ and PUB_SOURCE_REQ messages to each peer in the database. As peers in Kad join and leave dynamically, we need to re-collect peer information after a certain period because many of them might be invalid. In our experiment, we ran the crawler once every day.

For asymmetric attack, it just needs modifying the source IP and UDP port in BOOTSTRAP_REQ messages. For routing table reflection attack, it also needs generating spoofed peer ID close to target. We keep the first 15 bytes as same and randomly generate last 1 byte. For index reflection attack, we first collect some popular file hashes including movies, images, songs, programs, documents. Lots of eMule websites offer references of top-popular downloading. We used the information at China eMule [14]. Then we choose peer ID close to file hashes. Here we select the peer ID, the first 1 byte of which is same as file hash's. Note that the file size in PUB_SOURCE_REQ messages should be true, as eMule will validate file size when it matches indexes.

We ran a monitor on our victim host. It is a Java program and listens to victim ports (a TCP port for TCP connections and UDP port for UDP traffic). Following data are recorded: the number of newly created connections every minute, the number of connections alive every minute, the seconds each connection keeps, average throughput every minute and the distribution of source IP addresses.

B. *Results and Analysis*

In this section, we analyze the results of experiment in following three aspects: the effect of DDoS attacks by misusing Kad in eMule and how it is influenced by the time the attacker starts, the comparison between asymmetric attack and routing table reflection attack, and the distribution of attacks.

To control the impact on Kad network and Internet, we ran our experiment in limited conditions. The attacker was run as a single-thread and just continued several hours. We describe the performance of DDoS attacks by the number of new TCP connections every minute, the number of connections alive every minute and UDP incoming throughput. It is noticed that the monitor will consider the connection alive until the connecter closes it actively.

1) *The Effect of DDoS Attacks*

We varied the time the attacker started (recorded as time zone of Beijing) and kept the number of file identifiers as 400. The attacker continued 1 hour and the monitor on victim maintained 8 hours. The results are described in Fig.3, Fig.4 and Fig.5.

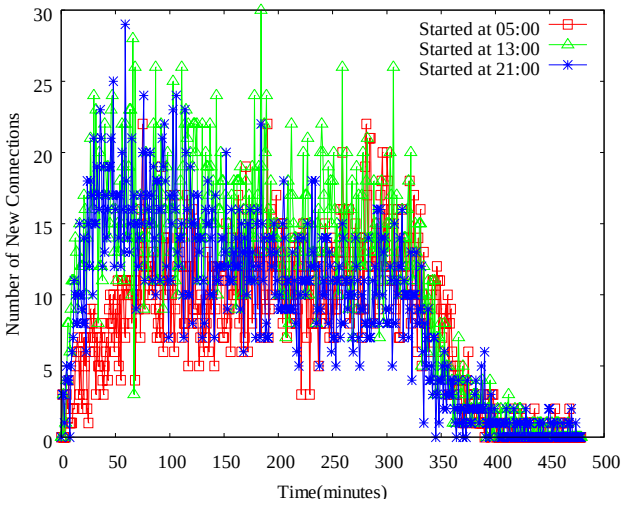

Figure 3. Starting at different time, number of new TCP connections (per minute) over time

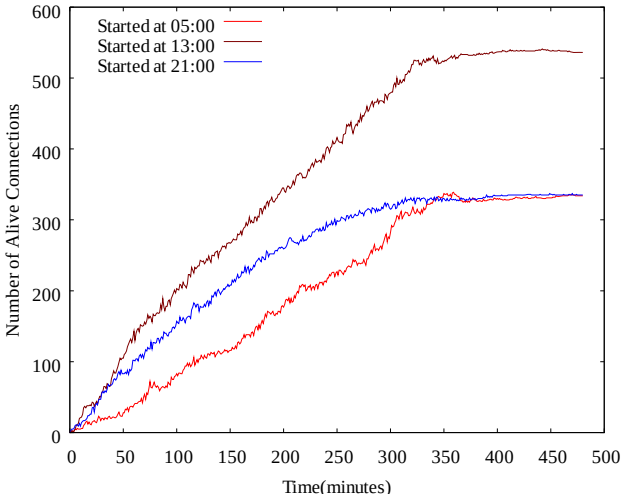

Figure 4. Starting at different time, number of TCP connections alive (per minute) over time

It was recorded that just 117 file identifiers was used in our experiment as the others don't match the first byte of target peer IDs. Fig.3 shows the number of new TCP connections over time. We see that the number of new TCP connections increases while the attacker continues and it maintains about 4 hours before it decreases in following 1 hour. This is due to the file location information just keeping 5 hours in eMule. It is interesting that there are some new connections even 5 hours

after the attacker is stopped. It seems that Kad network in eMule has a kind of "memory" about the file location information. We present the number of TCP connections alive over time in Fig.4. We see that it increases even in more than 4 hours after attacks were stopped and maintains in the following time. This shows that many peers in Kad don't actively close TCP connections once they are established successfully.

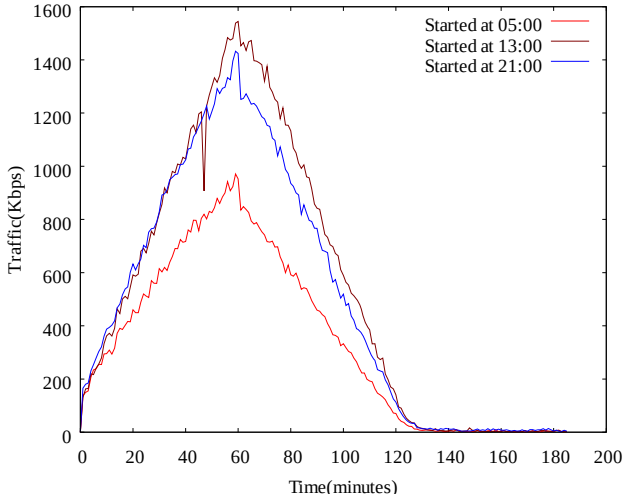

Figure 5. Starting at different time, attack UDP traffic over time

In Fig.5 we present attack UDP traffic in the first 3 hours as it becomes very small 2 hours after the attacker is stopped. UDP traffic increases linearly during the time of attacks and there is a sudden decrease when attacks are stopped which is due to the stop of asymmetric attack. The traffic decreases to about 2 Kbps in the following 1 hour and maintains about 2 hours.

From above three figures, we can see that the effect of attacks is influenced mostly by the time the attacker starts, and the effect at 13:00 is best, and then is 21:00, and the last is 05:00. This is mainly because of the most alive peers and most file-downloaders at 13:00. Table 1 shows average effect of DDoS attacks. The maximal alive TCP connections at 13:00 is 59.6% more than at 05:00 and the maximal UDP traffic is 59.1% more than at 05:00. Under our limited experiment, the average maximal number of TCP connections alive and maximal UDP traffic are 405 and 1,315Kbps respectively.

TABLE I. AVERAGE EFFECT OF DDOS ATTACKS

StartTime	***05:00***	***13:00***	***21:00***	***Average***
Average New TCP Connections per Minute in 4 Hours after Attacks were Stopped	11.4	15.4	12.4	13.1
Total New TCP Connections	3818	5281	4321	4473.3
Max Alive TCP Connections after Attacks were Stopped	339	541	337	405.7
Max UDP Traffic (Kbps)	970.8	1544.1	1432.5	1315.8

2) Comparison between Asymmetric Attack and Routing Table Reflection Attack

In above experiment, we send the same number of asymmetric attack messages and routing table reflection attack messages. However, an asymmetric attack message only needs 2 bytes while a routing table reflection attack message contains 22 bytes at least. To compare the effect between asymmetric attack and routing table reflection attack under same cost of attacker, we started both of them at the same time and the frequency of asymmetric attack messages is 11 times of the other's. The attacker continued 2 hours and the monitor on victim maintained 8 hours. Fig.6 shows their attack traffic.

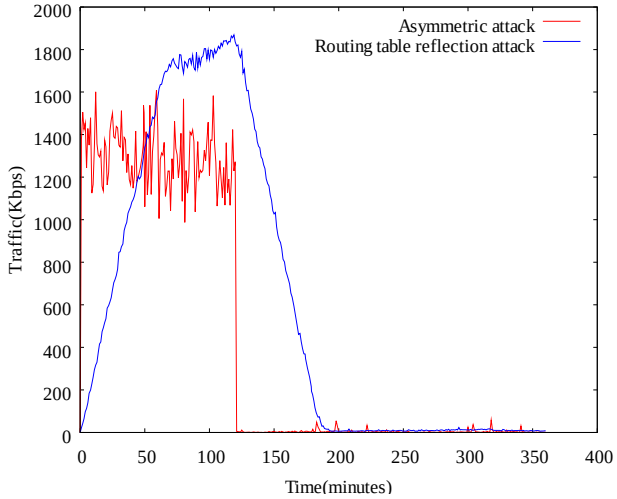

Figure 6. Under same conditions, attack UDP traffic over time

We can see that the effect of asymmetric attack ramps up rapidly in the first few seconds, however it is independent of attack duration and halts as soon as the attacker is stopped; while the effect of routing table reflection attack begins with small and increases linearly in the first one hour, and then increases slowly in following attack duration, which is because of the disappearance of the anterior attack messages as routing information only live one hour in eMule. It maintains about one hour even after attack is stopped. Therefore, these two attack methods can satisfy different needs, one for attacking quickly, the other for attacking gradually. Certainly, the combination of both will make the effect of attacks more sophisticated. We find that there is some UDP traffic at the victim port even 2 hours after the attacker is stopped. It seems that Kad network also has a kind of "memory" about the routing information.

3) The Distribution of Attacks

The CDF of TCP connection durations is shown as Fig.7. About 90% of connections maintain less than 300 seconds and the average duration of all connections are is 230 seconds. There are 3% of connections maintain more than 1500 seconds and the longest duration is 25597 seconds, more than 7 hours. These connections contribute to the large number of alive connections 5 hours after the attacker is stopped.

We recorded the source IPs at TCP connections and UDP traffic. Fig.8 shows the distribution of source IP. The file identifiers we used in attacker are referenced from a China website and most of them are popular in China, Europe and U.S, Thus, we can see that most source IPs of attacks come from Asia and Europe. All of these IPs distribute in more than 100 countries and areas. The broad distribution of source IPs makes defense of these DDoS attacks more difficult.

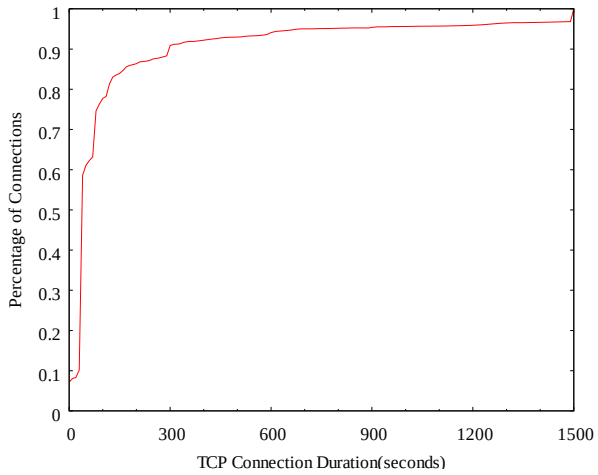

Figure 7. The CDF of TCP connection durations

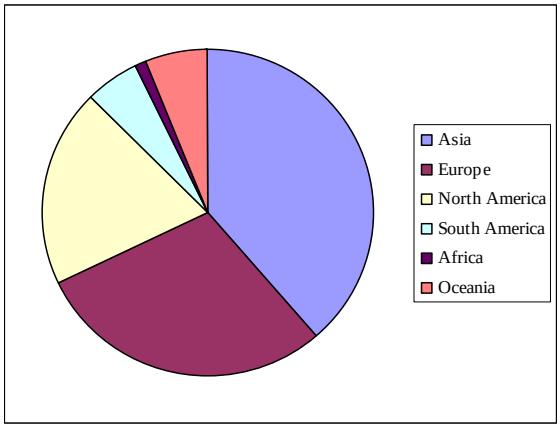

Figure 8. Distribution of source IP on continent

VI. FURTHER DISCUSSION

In this section, we will discuss some methods to amplify the performance of attacks and some solutions to mitigate or defend against these attacks.

A. *Amplifying Attacks*

Due to the characteristics of Kad, the performance of DDoS attacks is mainly influenced by the following factors: the time the attacker starts, the duration the attacker continues, file identifiers used in PUB_SOURCE_REQ messages, total attack messages sent in every second, the number of attackers, etc. It's very important to select proper file identifiers for index reflection attack. If you publish a file that no peer wants to download, there would be no connection request to victim port. Further more, the same file identifiers will induce different effect on different time which is validated in Section 5.2.1.

In real-world DDoS attacks, it is easy to amplify the performance of above experiment one thousand times or more. E.g. select more popular and more file identifiers, start attacks at the time there are more simultaneous users in Kad (such as 13:00~21:00 on Friday), maintain longer time (longer attacks continue, more peers will exchange and save spoofed file and peer information.), decrease the interval attack packets are sent, publish more relative keywords, build multi-layer attack model (combine with traditional intrusion methods), adopt distributed crawlers and attackers, etc. Further more, to protect them from pollution attack, most Kad networks provide notes publishing to comment files. We can offer high FILERATING of our file identifiers by sending spoofed PUB_NOTES request messages. This can enlarge the performance of index reflection attack too.

Since it is easy to detect and defend against single and regular attack, the attacker may employ more sophisticated strategies to evade detection.

- *Combination of three attacks and exploiting several vulnerabilities in one attack*. E.g. to perform asymmetric attack, we can misuse not only BOOTSTRAP message but also some other messages such as file location and metadata information query messages. Misusing many vulnerabilities simultaneously, the attacker can reach her purpose even part of these vulnerabilities are repaired or defended.
- *Variable message-arrival rate*. It may be detected and filtered by upstream ISPs if attacker sends attack messages at continuous and high rate. We can partly evade detection of ISPs by randomly changing message-arrival rate at different time.
- *Sending attack messages to different zones of peer ID*. Kad network may defend attacks by collaboration of some peers. To communicate and manage effectively, this collaboration is usually among some neighborhood peers or same sub-zone peers. We can choose peers alternately from different zones to avoid such defense.

B. *Defending Attacks*

Generally, DDoS defense mechanisms can be placed at three different locations: the source of attacks, middle network, and victim host or network. In the rest of this section, we will discuss several solutions for above attacks. These solutions should be thorough researched, deployed and tested in further work.

- When a peer receives a message, it verifies the location information of source peer firstly by sending a HELLO request message and then abandons it if no response is received in a certain time. This solution can avoid asymmetric attack, routing table reflection attack and index reflection attack, however, the verifying messages should be sent to victim and a new UDP attack will rise. Further more, verifying request message every time will enlarge the traffic in Kad network, as HELLO and PUB_SOURCE request messages update frequently.
- Set a timer *T*, and abandon BOOTSTRAP, HELLO and PUB_SOURCE request messages coming from same IP in a certain time. This solution can limit above attacks, however it is difficult to choose a proper timer and it might need to distinguish those three types of messages.
- Use trust management mechanism. When a peer receives a message, it abandons the message according to the trust degree of source peer. This solution can limit attacks too. Robust membership management proposed in [16] can restrict exploiting Kad network to perform DDoS attacks at certain degree. Our recent work is focusing on this defense method.

- Close source of these applications and encryption communication packets. Then the attacker couldn't analyze the format of messages and therefore can't send spoofed messages. However, some reverse-compile technique and sniffer technique can be used to get above information.
- Detect and filter attack packets at victim. The attack packets generated in Kad networks share some common characteristics, e.g. true source IP, same first several bytes (overlay packet header), etc. This solution can limit attacks too. *DOW* mechanism proposed in [10] can filter some attack packets.

VII. Conclusion and Future Work

In this paper, we showed that it is possible to launch a DDoS attack against any host on Internet by misusing Kademlia Protocol. We classify these attacks as three types: asymmetric attack, routing table reflection attack and index reflection attack. We ran a limited real-world experiment on eMule and the results show that these attacks tie up bandwidth and TCP connection resources of victim. More large scale DDoS attack can be performed according to a little more effort.

We discuss several solutions for these DDoS attack methods. Each solution has both advantages and disadvantages. Our future work will focus on studying and experimenting on these solutions and proposing new defense methods.

References

[1] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher. Internet Denial of Service: Attack and Defense Mechanisms, Prentice Hall PTR, 2004.

[2] 2007 NGN forum, http://www.catr.cn/zhthg/ngn/2007/

[3] E. Athanasopoulos, K. Anagnostakis, and E. Markatos. Misusing Unstructured P2P systems to Perform DoS Attacks: The Network That Never Forgets, in Proc. ACNS 2006.

[4] N. Naoumov and K. Ross. Exploiting P2P Systems for DDoS Attacks, in Proc. of INFOSCALE, 2006.

[5] K. Cheung Sia. DDoS Vulnerability Analysis of Bittorrent Protocol, UCLA Tech. Report, Spring 2006.

[6] J. Liang, N. Naoumov, and K. W. Ross. The Index Poisoning Attack in P2P File Sharing Systems, In IEEE Conference on Computer Communication, Barcelona, Spain, April 2006.

[7] K.El Defrawy, M.Gjoka, and A.Markopoulou. Bottorrent: Misusing BitTorrent to Launch DDoS Attacks, in Usenix SRUTI, Santa Clara, 2007.

[8] R. Brunner. A performance evaluation of the Kad-protocol, Master Thesis, 2006.

[9] P.Maymounkov and D. Mazières. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric, the First International Workshop on Peer-to-Peer Systems, p.53-65 , 2002.

[10] J. Yu, Z. Li, H. Chen, and X. Chen. A Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks,Third International Conference on Networking and Services, 2007. ICNS. Page(s): 54-54.

[11] Y. Liu, X. Liu, W. Chen, and X. Li. Defending P2Ps from Overlay Flooding-based DDoS. ICPP 2007

[12] J. Harrington and C. Kuwanoe, C. Zou. A BitTorrent-Driven Distributed Denial-of-Service Attack. SecureComm 2007.

[13] Emule, http://sourceforge.net/projects/emule/

[14] China Emule, http://www.emule.com.cn/

[15] X. Sun, R. Torres, and S. Rao. DDoS Attacks by Subverting Membership Management in P2P Systems, NPSec 2007.

[16] X. Sun, R. Torres, and S. Rao. Preventing DDoS Attacks with P2P Systems through Robust Membership Management, Tech. Rep., 2007.