DDoS Attacks by Subverting Membership Management in P2P Systems

Xin Sun, Ruben Torres and Sanjay Rao
Purdue University
{*sun19,rtorresg,sanjay*}*@ecn.purdue.edu*

***Abstract*— We show that malicious participants in a peer-to-peer system can subvert its membership management mechanisms to create large-scale DDoS attacks on nodes not even part of the overlay system. The attacks exploit many fundamental design choices made by peer-to-peer system designers such as (i) use of push-based mechanisms; (ii) use of distinct logical identifier (e.g. IDs in a DHT) corresponding to the same physical identifier (e.g., IP address), typically to handle hosts behind NATs; and (iii) inadequate or poorly designed mechanisms to validate membership information. We demonstrate the significance of the attacks in the context of mature and extensively deployed peer-to-peer systems with representative and contrasting membership management algorithms - DHT-based Kad and gossip-based ESM.**

I. Introduction

Peer-to-peer systems are rapidly maturing from being narrowly associated with copyright violations, to a technology that offers tremendous potential to deploy new services over the Internet. The recently released Windows Vista is equipped with its own, under-the-hood P2P networking system [1], and several commercial efforts are exploring the use of peer-to-peer systems for live media streaming and video distribution [2], [3]. Recent studies [4] indicate that over 60% of network traffic is dominated by peer-to-peer systems, and the emergence of these systems has drastically affected traffic usage and capacity engineering.

With the proliferation of peer-to-peer systems, it becomes critical to consider how they can be deployed in a safe, secure and robust manner, and understand their impact on an Internet environment already suffering from several security problems. Peer-to-peer systems enable rapid deployment by moving functionality to end-systems. However, they are vulnerable to insider attacks coming from (potentially colluding) attackers that infiltrate the overlay or compromise member nodes.

Several works [5]–[9] have studied how malicious nodes in a peer-to-peer system may disrupt the normal functioning, and performance of the system itself. In this paper, however, we focus on attacks where malicious nodes in a peer-to-peer system may impact the **external** Internet environment, by causing large-scale distributed denial of service (DDoS) attacks on nodes not even part of the overlay system. In particular, an attacker could subvert membership management mechanisms, and force a large fraction of nodes in the system to believe in the existence of, and communicate with a potentially arbitrary node in the Internet. Such attacks may be hard to detect as the packets arriving at a victim are not distinguishable from normal protocol packets. These attacks may be viewed as a particular kind of reflector attacks [10], however the scale and unique properties of peer-to-peer systems make them worthy of study in their own right.

In this paper, we show that a potential attacker can launch attacks of hundreds of megabits a second on an external node, by exploiting popularly deployed file distribution systems such as eMule [11], and the extensively deployed video broadcast system ESM [12]. The systems represent contrasting applications, and involve different and representative membership management designs - structured DHT-based and unstructured gossip-based. Our attacks exploit fundamental design choices made by peer-to-peer system designers such as (i) use of push-based mechanisms; (ii) use of distinct logical identifier (e.g. IDs in a DHT) corresponding to the same physical identifier (e.g., IP address), typically to handle hosts behind NATs; and (iii) inadequate or poorly designed mechanisms to validate membership information. Overall, the attacks shed new insights on the interplay between membership management mechanisms, and the feasibility of exploiting P2P systems to cause DDoS attacks.

The rest of the paper is organized as follows. Section II describes vulnerabilities in the Kad and ESM systems. Section III shows results demonstrating the feasibility of exploiting these systems for DDoS attacks. Section IV describes guidelines for the design of robust membership management mechanisms.

II. Vulnerabilities in P2P Systems

In this paper, we focus on DDoS attacks triggered by exploiting the membership management algorithms of peer-to-peer systems. The membership management algorithms in a peer-to-peer system enable a node to join the group, and maintain information about other members, even though nodes may join or leave the system. To scale to large group sizes, typical nodes maintain knowledge of only a small subset of group members. Two of the most common approaches for membership management involve

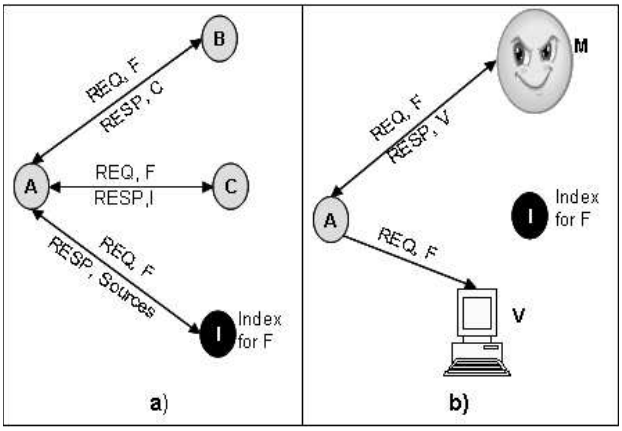

Fig. 1. a) Kad search mechanism. b) Redirection attack

the use of distributed hash tables (DHTs) [13], and gossip-based algorithms. While popular file-distribution systems like BitTorrent [14] and eMule [11] originally relied on centralized servers (trackers) for group management, more recent versions use decentralized mechanisms based on DHTs. Many other systems such as ESM and CoolStreaming [12], [15] employ gossip-like mechanisms to maintain group membership information.

To demonstrate the generality of the issues discussed, we consider peer-to-peer systems targeted at applications with very contrasting properties and very different membership management designs. The particular systems we consider in this work include Kad [11] for file distribution and ESM [12] for video broadcasting. Kad is a DHT based on Kademlia [13], which is supported by the popular eMule [11] client and its clones. Kad is the largest DHT currently used, with more than one million concurrent peers [16]. ESM is a video broadcasting system that employs gossip-based membership algorithms. It is one of the first operationally deployed systems and has seen significant real-world deployment [12].

A. DHT-Based File Distribution:Kad

In Kad, users and files have IDs that are globally unique and randomly chosen from the same ID space of **128** bits. Each node maintains a routing table with a subset of peers that are part of the system. For any given file, there are "index-nodes" which maintain a list of members who own that file. Index nodes are not dedicated nodes but regular participants, who have an ID close to a file ID. For example, in Figure 1.a), node A wishes to download a file F. A must first discover the index-node I, and obtain from it a list of members having the file. To discover I, A will query members that it has in its own table, which are either index nodes or can point A to nodes closer in the ID space to the file ID, which are likely to be index nodes. In our example, A will initially query B which is not an index node for file F. B responds with C whose ID is closer to the ID of F. Next, A will query C who will respond with I. This process can repeat several times, but given the properties of distributed hash tables, convergence of the search process is likely. In our case, I is the index node for file F and will respond to A with a set of *sources* that own a partial or complete copy of the file. Finally, A will contact the sources to begin the download process. Kad performs a similar lookup process for keyword search, file and keyword publishing, and routing table maintenance.

Vulnerability: Kad may be exploited to cause a DDoS attack on a victim that is not part of the Kad network by creating a redirection attack. Whenever the attacker receives a lookup query from a peer, it will return a response containing the victim's IP address and port. For example, Figure 1.b) shows how malicious user M can make user A send a query to V, which is an Internet user, not part of the Kad network. The attack at the victim can be magnified if many valid users contact the attacker when looking for index nodes. In addition, a coalition of attackers could further increase the magnitude of the traffic at the victim. Note all Kad control packets use UDP.

B. Gossip-based Video Broadcast: ESM

ESM is a video broadcasting system built on top of an overlay network. It constructs a multicast tree for data delivery and employs a gossip-based mechanism to propagate the existence of members on the group. Each member A, periodically picks another member B at random, and sends it a subset of group members that it knows. B adds to its list any members that it did not already know, and may send messages to the new nodes as part of normal protocol operations. This membership information is later used by the nodes in the system to change parents when necessary in the multicast tree. Note all ESM control messages use UDP.

Vulnerability: The gossip mechanism to propagate membership information, may be exploited by having malicious users trick valid users into sending protocol related messages to a victim that is not part of ESM. A malicious user M, could generate a gossip message that contains false information about the victim as being part of the group. Valid users will include the victim in their list of known peers. At a later time, the victim could receive a high rate of unsolicited traffic from valid users of the system. The attack can be magnified if malicious users gossip fake messages at a higher rate.

III. Achieving high magnitude DDoS Attacks

We discuss how the vulnerabilities in Kad and ESM may be exploited to create large-scale DDoS attacks.

A. Attack using Kad

The feasibility of misusing DHT-based systems to launch DDoS attacks has been shown by [17] in the context of the Overnet system. However, Kad has important differences from Overnet which makes it more

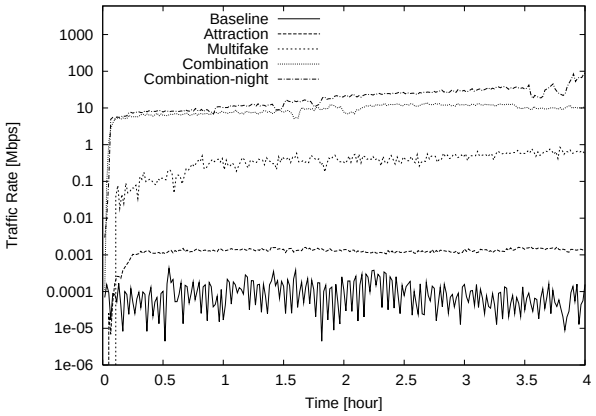

Fig. 2. Sensitivity to the heuristic employed by the attacker to increase the attack

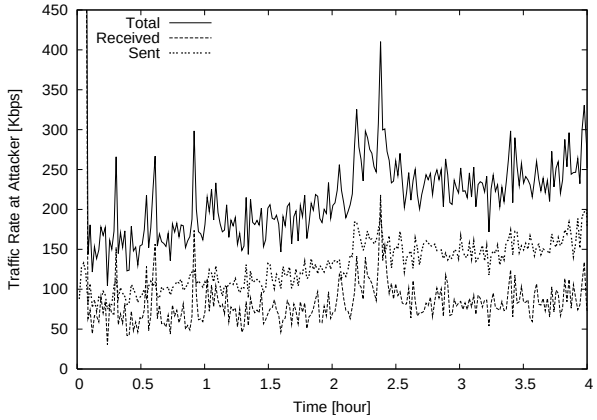

Fig. 4. Traffic seen at an attacker using all heuristics to generate attack

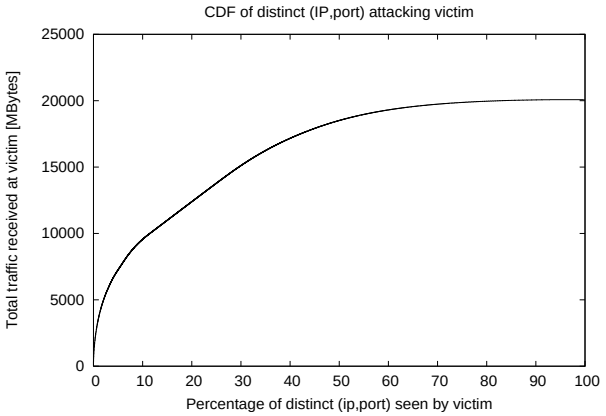

Fig. 3. CDF of the distinct (IP,port) pairs generating traffic at the victim

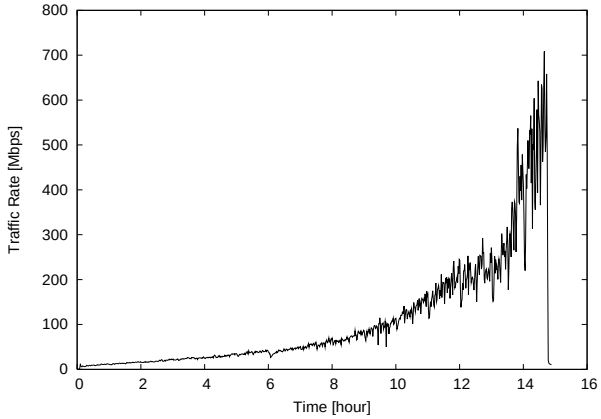

Fig. 5. Total traffic seen at victim, with 200 attackers, over a period of 15 hours.

robust to the routing poisoning attacks presented in [17]. In particular, in Overnet, when a client A hears about a new node V, it uses the information without explicitly verifying that V is actually part of the system. In an attack, if V corresponds to the victim node, several packets may be sent from A to V, before it is ultimately purged from A's table. This was a key reason for attack amplification in [17]. In contrast, in Kad, when client A hears about V, it explicitly probes V, and sends it further messages only if it receives a response. In addition, information about V is propagated to other members only when a response is received. Consequently each poisoning event is associated only with one spurious packet to the victim.

We identify and present fundamentally new vulnerabilities and attacks that can lead to higher amplifications even with the more robust design taken by Kad. Our complete set of heuristics is as follows:

• *Baseline:* As described in Figure 1.a), node A may seek to locate the node nearest to a given target ID F. As part of the operations, it may send a message to a (malicious) node M that A already knows, who in turn responds with the IP address and port of the victim V (indicating that V is part of the group) along with a fake id for V which is closer to F. A then sends query messages to V, as part of its normal operation.

• *Attraction:* The magnitude of the attack above is dependent on the frequency with which other nodes may contact the malicious node. In general, this is small given that the group may involve millions of members, but there are only a few attackers. However, with Kad, a malicious node may proactively push information about itself to a large number of nodes in the system, forcing them to add the node to their routing tables. We believe this feature is important to the Kad design and generalizes to other systems. We suspect that it was introduced to ensure a newly joining node can be learnt by enough members in the group, as well as hosts behind Network Address Translators (NATs). We are currently conducting experiments to better understand the negative impact of disabling the heuristic on Kad.

• *Multifake:* While the attacks above cause several clients to contact the attacker and be redirected to the victim, better amplification can be achieved if the attacker includes the victim's information several times in response to a

query. The key insight behind the attack is the distinction between the *physical identifier* of a participating node such as its IP address, and its *logical identifier*, the node-id in the DHT space. Kad, and indeed many peer-to-peer systems, are designed to allow a participating node to communicate with multiple logical identifiers even though they share the same physical identifier (IP address). This has several advantages, for instance, enabling distinct users behind the same Network Address Translator (NAT) to participate in the system, even though they share the same physical IP address. The *Multifake* heuristic exploits this to achieve large amplifications by having the attacker redirect innocent clients to multiple logical identifiers, all sharing the IP address of the victim. Further, it seeks to achieve even greater magnification by having the attacker include itself in the query responses a small number of times, so that the valid user could be repeatedly attracted to the attacker and redirected to the victim.

Results: We implemented the attackers by incorporating the heuristics above in an aMule client (a clone of eMule). The attackers join the real live Kad network. The victim node is in our laboratory. Each experiment runs for several hours, and we report on magnitudes of attack seen. The experiments employ 5 attackers unless otherwise mentioned.

Figure 2 shows the traffic at the victim with the three heuristics. The X-Axis is the time since the start of the experiment. The Y-Axis is the amount of traffic seen in Mbps. From bottom to top, the first three curves correspond to the attack magnitude with the given heuristic alone. The last two curves correspond to the combination of the previous heuristics at different times of the day. High magnitudes of over 10Mbps seen at the victim when all heuristics are turned on. It is also interesting to see that the entire set of heuristics is required to generate the high attack magnitudes, and any subset is insufficient. We also observed that the magnitude of the attack traffic is sensitive to the time of day. As shown in the last two curves *Combination* was obtained during the day and *Combination-night* was obtained late at night. In our experiments, the attacks could go as high as 100 Mbps in some runs during the night.

Figure 3 shows the distribution of distinct (IP address,port) pairs of innocent clients that are being redirected to the victim when all heuristics are turned on. A point (X,Y) in this graph means that traffic Y is contributed by X percent of distinct IP and port. Over 200,000 distinct (IP,port) pairs were redirected in the attack. As shown, the distribution is not sharply skewed indicating the traffic is not coming from any single client alone which can make the attack difficult to contain.

Figure 4 shows the traffic observed at one attacker, both in terms of traffic sent and received. The Y-Axis is traffic rate in Kbps. The X-Axis is time in hours. The main observation from this graph is that the traffic seen at the attacker is only about 250Kbps, which while higher than what a normal user sees, is 40 times lower than the traffic seen by the victim. Even if the total traffic at all attackers is consider, there is still a magnification factor of 8. A point to note is the spike at the start of the experiment. This is due to the attraction heuristic where a malicious node attempts to insert itself in the routing tables of several other nodes. We discuss the implications in the next paragraph.

We considered whether even higher attack magnitudes are possible by combining larger number of nodes and increasing the rate of the attraction heuristic. Figure 5 shows an attack generated using 200 malicious nodes scattered around Planetlab, with the victim in our laboratory. Traffic of over 700 Mbps was received by the victim after 14 hours of experiment. This far exceeded what we feared, and indicates the criticality and seriousness of the problem. We abandoned further experiments on this line given the seriousness of the attacks. An ISP of one of the attacker nodes was concerned whether the node was running a random port-scan attack. This was because each attacker probed around 100,000 Kad nodes as part of the attraction heuristic, and not all nodes responded since they were no longer in the system. While this offers hope that such attacks could be detected, it may be feasible to evade detection by reducing the rate at which malicious nodes spread information about themselves to others. Significant attack magnitudes may still be achieved, though it may take longer for the attacks to ramp up to these values. We have conducted (carefully controlled) experiments to confirm this observation.

B. Attack using ESM

We exploit the vulnerability described in Section II where a malicious node M sends gossip messages to an innocent client C, falsely indicating that the victim V is part of the group. Similar to *Multifake* in Kad, we augmented the heuristic to achieve greater attack magnitudes by including the IP address of the victim several times, each time with a different logical identifier. ESM also makes use of logical identifiers (called uid in [12]) distinct from IP address and port information, primarily to handle issues with NATs. Like Kad, ESM allows a participating node to communicate with multiple logical identifiers even though they share the same physical identifier (IP address). Again this is motivated by NATs.

Results: We conducted experiments in Planetlab using the attacker described above. Figure 6 shows the magnitude of the DDoS attack in comparison to the total number of ESM clients. We fixed the percentage of malicious clients to be 10% and varied the total number of clients. The traffic seen by a victim is several Megabits a second, a factor of 1000 more than control traffic seen by a normal ESM member (about 3 Kbps). Further, the attack traffic increases approximately linearly as the number of

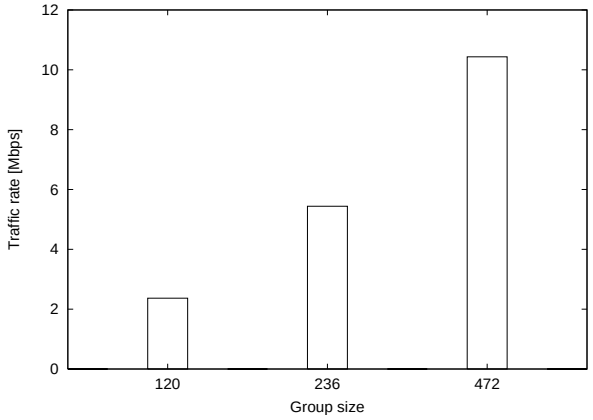

Fig. 6. Sensitivity to number of clients. Percentage of malicious clients fixed to 10%

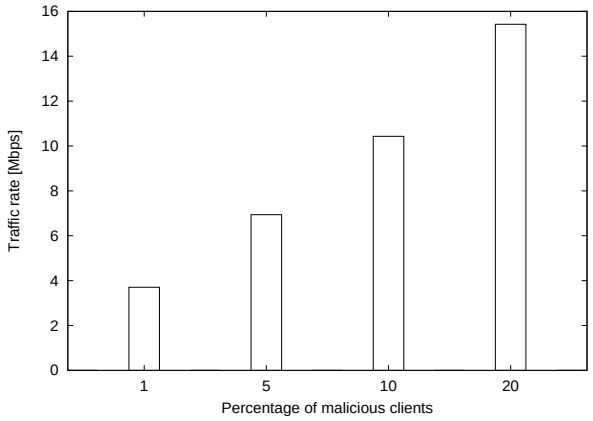

Fig. 7. Sensitivity to percentage of malicious clients. Total number of clients fixed to 472

participants increase. In a real scenario, involving tens of thousands of participating nodes, the attack magnitude could be orders of magnitude higher. The experiments above assume that 10% of the hosts are malicious. Figure 7 plots the attack traffic fixing the number of clients at 472, and varying the percentage of malicious clients. Even a very small fraction of malicious clients can cause a serious attack at the victim, with 1% of nodes being malicious resulting in attacks of 4Mbps at the victim.

IV. DDoS Resistant Membership Management

While several "point-solutions" may be feasible to limit the specific attacks we presented in Section III, we believe these attacks are symptomatic of more fundamental issues that must be carefully addressed in designing robust membership management protocols. We believe a three-pronged strategy is required:

• The protocol must limit the ability of the attacker to redirect or infect a large number of innocent clients. One strategy is to favor pull-based designs where any information conveyed by a member is always in response to a prior solicitation, over push-based designs, where members may disseminate membership information to other members in an unsolicited fashion. Push-based protocols are more vulnerable to compromise, since an attacker can control the rate at which it can contact other victim nodes. Pull-based algorithms may themselves not suffice. In particular, mechanisms are needed to limit the number of nodes that know an attacker, which may in turn regulate the ability of an attacker to attract queries from innocent nodes toward itself.

• It is important to validate membership information that a node receives. One possibility involves direct validation, where a node directly probes the new member it learns about to verify its existence. Such a scheme may itself become a source of spurious packets to the victim. In fact, this mechanism is present in Kad but did not prove sufficient. Another approach is to adapt ideas from Byzantine-tolerant diffusion algorithms [18], [19], where a node does not directly contact C to validate it, but waits until it learns about C from multiple nodes.

• It is important to limit amplification attacks where a malicious node could repeatedly redirect an innocent client to a victim IP address, but using different logical identifiers for the victim IP each time. Naive approaches to bound the communication may not suffice since there may be actual instances where nodes with different logical identifiers share the same physical identifier (e.g. due to NATs), and further, such bounding heuristics are subject to disconnection attacks where an attacker could disconnect a client (victim) who is really part of the group, by flooding other clients with several logical identifiers for the victim, and the same physical identifier of the victim.

We have discussed these issues further in [20], and have also presented an evaluation of heuristics motivated by these principles.

V. Related Work

It was first observed in [17] that the intrinsic characteristics of P2P systems could be exploited for indirection attacks. In this paper, we have taken this observation much further by presenting new insights on how fundamental design choices made by peer-to-peer system designers may impact the ability of the system to be exploited. Our novel insights include vulnerabilities due to (i) use of push-based mechanisms; (ii) use of distinct logical identifier (e.g. IDs in a DHT) corresponding to the same physical identifier (e.g., IP address), typically to handle hosts behind NATs; and (iii) lack of mechanisms to validate membership information (as in ESM), or use of simplistic validation mechanisms (as in Kad) which may themselves be a source of attack.

Several researchers have shown the feasibility of exploiting P2P systems to launch DDoS attacks. The attacks have been shown on Overnet [17], Gnutella [21], and most recently and parallel to this work, BitTorrent [22], [23]. Ours is the first work to show the feasibility of exploiting

Kad, as well as first to show that the problem also affects P2P systems used for video broadcasting.

The only prior work that presents attacks misusing DHT-based peer-to-peer systems is [17], conducted in the context of Overnet. As described in Section III-A, Kad has better mechanisms to explicitly verify membership information than Overnet, which limits the amplification of the routing poisoning attacks presented in [17]. The attack in [17] exploits a vulnerability in Overnet which enables the attacker to announce the IP and port of any arbitrary node as part of the application payload, and push this information to innocent clients. Instead, in the attacks we present, the malicious node attracts a large number of client requests, and repeatedly redirects them to the victim using the *Multifake* heuristic. Overall, the combination of heuristics used generates significantly higher attack magnitudes than [17], despite Kad having better defenses than Overnet.

[22] considers attacks where the victim is falsely advertised as one of the central entities (trackers) forcing the clients to aggressively contact the victim as part of the protocol. In contrast, our focus is on the distributed DHT-based Kad network. Tackling DDoS attacks using peer-to-peer systems has received attention in the industry [24]. Finally, while our focus is on exploiting P2P systems to launch DDoS attacks, other works have explored attacks caused by DNS and web-server reflectors, and misuse of web-browsers and bot-nets [10], [25], [26].

VI. Summary and Conclusions

We have shown that malicious participants in a peer-to-peer system can subvert its membership management mechanisms to create large-scale DDoS attacks on the Internet. High magnitude attacks of the order of hundreds of Mbps can be created with as few as tens of malicious participants. Our results are shown on mature and extensively deployed peer-to-peer systems with representative and contrasting membership management algorithms - DHT-based Kad and gossip-based ESM. The attacks exploit many fundamental design choices made by peer-to-peer system designers such as (i) use of push-based mechanisms; (ii) use of distinct logical identifier (e.g. IDs in a DHT) corresponding to the same physical identifier (e.g., IP address), typically to handle hosts behind NATs; and (iii) lack of mechanisms to validate membership information (as in ESM), or use of simplistic validation mechanisms (as in Kad) which may themselves be a source of attack. Our ongoing and future work involves design of membership management schemes robust to such attacks, and exploring techniques that can detect DDoS attacks exploiting peer-to-peer systems.

References

[1] "Windows peer-to-peer networking," http://www.microsoft.com/p2p.

[2] S. Ali, A. Mathur, and H. Zhang, ""Measurement of commercial peer-to-peer live video streaming"," in *Proc. of Workshop on Recent Advances in P2P Streaming*, 2006.

[3] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, ""Insights into pplive: A measurement study of a large-scale p2p iptv system"," in Proc. Workshop on Internet Protocol TV (IPTV) services over World Wide Web in conjunction with WWW2006, May 2006.

[4] K. Cho, K. Fukuda, and H. Esaki, ""The impact and implications of the growth in residential user-to-user traffic"," in *Proc. ACM SIGCOMM*, 2006.

[5] E. Sit and R.Morris, "Security Considerations for Peer-to-Peer Distributed Hash Tables," in *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS)*, March 2002.

[6] J. Douceur, "The Sybil Attack," in *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS)*, March 2002.

[7] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D.S.Wallach, "Security for structured peer-to-peer overlay networks," in *Proceedings of OSDI 2002*, December 2002.

[8] A. Singh, T.-W. Ngan, D. P, and D. Wallach, "Eclipse Attacks on Overlays: Threats and Defenses," in *Proceedings of INFOCOM 2006*, April 2006.

[9] D. Wallach, "A Survey of Peer-to-Peer Security Issues," in *International Symposium on Software Security*, November 2002.

[10] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks. Computer Communication Review 31(3), July 2001."

[11] EMule, http://www.emule-project.net.

[12] Y. Chu, A. Ganjam, T. S. E. Ng, S. G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early Experience with an Internet Broadcast System Based on Overlay Multicast," in *Proceedings of USENIX*, June 2004.

[13] P. Maymounkov and D. Mazieres, ""Kademlia: A peer-to-peer information system based on the xor metric."," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, 2002.

[14] BitTorrent, http://www.bittorrent.org.

[15] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming," in *Proceedings of IEEE INFOCOM*, 2005.

[16] D. Stutzbach and R. Rejaie, "Improving Lookup Performance over a Widely-Deployed DHT," *proceedings of IEEE INFOCOM*, 2006.

[17] N. Naoumov and K. Ross, ""Exploiting p2p systems for DDoS attacks"," in *"International Workshop on Peer-to-Peer Information Management"*, May 2006.

[18] D. Malkhi, Y. Mansour, and M. Reiter, "Diffusing without false rumors: On propagating updates in a byzantine environment," *Theoretical Computer Science*, vol. 299, no. (1-3), pp. 289–306, 2003.

[19] Y. Minsky and F. Schneider, "Tolerating malicious gossip," *Distributed Computing*, vol. 16, no. 1, pp. 49–68, February 2003.

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

[21] E. Athanasopoulos, K.G.Anagnostakis, and E. Markatos, ""Misusing unstructured p2p systems to perform dos attacks: The network that never forgets"," in *Proceedings of the 4th International Conference on Applied Cryptography and Network Security (ACNS'06)*, 2006.

[22] K. E. Defrawy, M. Gjoka, and A. Markopoulou, ""BotTorrent: Misusing BitTorrent to launch DDoS attacks"," in *Usenix SRUTI*, 2007.

[23] K. C. Sia, "DDoS Vulnerability Analysis of BitTorrent Protocol," Tech. Rep., 2007.

[24] Prolexic, http://prolexic.com/news/20070514-alert.php.

[25] V. Lam, S. Antonatos, P. Akritidis, and K. G. Anagnostakis, ""Puppetnets: Misusing web browsers as a distributed attack infrastructure"," in *"Proc. of ACM Computer and Communication Security"*, 2006.

[26] "DNS amplification attacks," http://www.isotf.org/news/DNS-Amplification-Attacks.pdf.