# The Store-and-Flood Distributed Reflective Denial of Service Attack

Bingshuang Liu[*], Skyler Berg[†], Jun Li[†], Tao Wei[‡], Chao Zhang[‡], Xinhui Han[*§]
[*]Institute of Computer Science and Technology, Peking University, China
Email: {liubingshuang, hanxinhui}@pku.edu.cn
[†]University of Oregon, USA
Email: {lijun, skylerb}@cs.uoregon.edu
[‡]University of California, Berkeley, USA
Email: {lenx.wei, gausszhch}@gmail.com

***Abstract*—Distributed reflective denial of service (DRDoS) attacks, especially those based on UDP reflection and amplification, can generate hundreds of gigabits per second of attack traffic, and have become a significant threat to Internet security. In this paper we show that an attacker can further make the DRDoS attack more dangerous. In particular, we describe a new DRDoS attack called store-and-flood DRDoS, or SF-DRDoS. By leveraging peer-to-peer (P2P) file-sharing networks, SF-DRDoS becomes more surreptitious and powerful than traditional DRDoS. An attacker can store carefully prepared data on reflector nodes before the flooding phase to greatly increase the amplification factor of an attack. We implemented a prototype of SF-DRDoS on Kad, a popular Kademlia-based P2P file-sharing network. With real-world experiments, this attack achieved an amplification factor of 2400 on average, with the upper bound of attack bandwidth at 670 Gbps in Kad. Finally, we discuss possible defenses to mitigate the threat of SF-DRDoS.**

**Keywords: DDoS, DRDoS, Amplification factor, Kademlia, Store-and-flood**

## I. Introduction

While distributed denial of service (DDoS) attacks have posed a significant threat to Internet security for many years, recently distributed reflective denial of service (DRDoS) attacks have become prevalent and received a lot of attention due to their severity. One of the largest DDoS attacks in history, a UDP-based DRDoS that occurred between Spamhaus and Cyberbunker on March 18, 2013, generated over 300 Gbps attack traffic through DNS amplification techniques [1], a tremendous traffic volume that could bring down virtually any service on today's Internet.

In a typical DRDoS attack, the attacker first sends many requests with a spoofed source IP address—i.e., the address of the victim—to so called *reflector* nodes, which in turn reply with numerous and often voluminous responses to the spoofed IP, thereby flooding the victim. Two key metrics for measuring the severity of a DRDoS attack are **amplification factor**, or **AF**, that is the ratio between the traffic volume of response packets and that of request packets, and **attack ability** that is the amount of attack traffic launched toward the victim. Note that reflectors are usually meant to provide a legitimate service and are seldom aware that they are being exploited to produce a large attack bandwidth.

Unfortunately, DRDoS attacks can be even more dangerous than expected. In particular, we notice in our recent studies that peer-to-peer (P2P) file-sharing applications can be leveraged to conduct more powerful UDP-based DRDoS attacks. We observe three features that make P2P applications particularly attractive for DRDoS attacks: (i) P2P applications use UDP messages frequently, such as the index services provided by Distributed Hash Tables (DHT [2]), making IP address spoofing easy to perform. (ii) All P2P users can freely access and store various data on other nodes in a P2P network, making almost all nodes in the P2P network perfect candidates for DRDoS reflectors. (iii) P2P applications often have a huge user base. At present, the user population of popular P2P file-sharing applications, such as Kad [3] and BitTorent [4], has reached the millions [5] [6].

In this paper, we present a new type of DRDoS attack called **store-and-flood DRDoS** attack, or **SF-DRDoS**. The most notable characteristic of SF-DRDoS is that an adversary prepares and stores carefully crafted data on reflector nodes before issuing spoofed requests with the IP of a victim to reflector nodes to flood the victim. This strategy can yield a much higher AF than previously explored DRDoS approaches [7]. The adversary can further adjust the timing, content, and in particular the volume of the responses.

Furthermore, we present a prototype SF-DRDoS system based on Kad, a popular Kademlia-based P2P file-sharing network. It consists of a crawler to crawl the network, a node group to store index entries, and another node group to flood the victim. We investigate various factors that may affect the ability of the attack, and build a model to help illustrate the relationship between the attacker's bandwidth cost, the reflectors' response sizes, and the AF.

We further conduct real-world experiments to evaluate the effectiveness and flexibility of SF-DRDoS attacks, and found the average AF is about 2400, much higher than the AF achieved by current DRDoS attacks. The peak AF can reach 4326, making it possible to use only a 205-Kbps bandwidth to generate an attack flow of about 865 Mbps. If an attacker had enough bots, it could initiate an attack that costs only about 280 Mbps to generate a SF-DRDoS attack of more than 670 Gbps.

[§] Corresponding author.

Finally, we propose defense solutions to filter out the attack traffic generated by SF-DRDoS attacks. These defenses are based on BGP flow specification [8], and are able to filter attack traffic at the upstream links. Note the attacker obeys the specifications of P2P networks and the attack traffic has no specific characteristics to be distinguished from legitimate traffic.

## II. Background and Related Work

### A. The DRDoS Attack

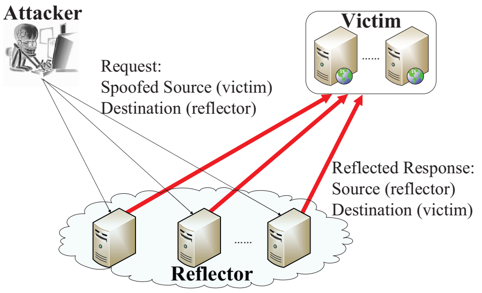


Fig. 1. The working mechanism of a DRDoS attack.

Fig. 1 illustrates the working mechanism of a DRDoS attack. Because the victim can only see that the attack traffic coming from reflectors, the attacker is difficult to locate. To conduct an effective DRDoS attack, the following conditions should be met:

- The transport protocol should be stateless and lack authentication so that the attacker can use spoofed IP addresses in their requests. Otherwise, the potential reflectors cannot be fooled into sending responses to the victim.
- There should be abundant reflectors that are open to all Internet users. Insufficient availability of reflectors caps the attack ability. For example, there are more than 27 million open DNS resolvers on the Internet that can be misused as reflectors [9].
- Some requests should trigger large responses according to the communication protocol in place, thus enabling amplification. Otherwise, the attack ability will be reduced to that of a non-reflective attack or even lower.

Also desirable for attackers are protocols which are difficult to filter, such as those using non-fixed UDP ports or that encrypt or obfuscate messages by default.

### B. Related Work

*1) DRDoS Attack Methods:* Researchers have conducted comprehensive analysis of the DRDoS attack since 2001 [10]. The earliest well-known DRDoS is probably the smurf attack [11], which sends spoofed ICMP echo requests to subnet broadcast addresses to trigger massive echo responses to a victim. Kumar *et al.* [12] further investigated the factors that affect the attack ability of the smurf attack and explained the relation between the attack cost, the reflector network and the final amplified attack traffic. TCP-based DRDoS attacks have also been studied [13], but due to the three-way handshake process, these attacks can only happen during the connection establishment phase, and have no significant amplification effect. The most popular DRDoS attacks on the Internet are UDP-based. These attacks exploit popular Internet services to greatly amplify the attack traffic. Recently, Rossow [7] revisited popular UDP-based protocols, and found 14 protocols are exploitable as reflectors, with the AF reaching as high as 4670 when exploiting NTP severs. However, the computation of the AF only considered UDP payload. When considering the packet header as shown in our paper, the actual AF is only about 700, much lower than that of SF-DRDoS attacks that we will present in this paper.

*2) P2P-based DRDoS and DDoS Attacks:* Similar to our study, a few studies investigated the DRDoS attack via P2P networks. The possibility of reflection and amplification in Kad was found in [14], which utilizes bootstrap requests to gain an AF of about 8. Rossow [7] also investigated the AFs in Kad, discovering the average AF (without including packer headers) to be 16.3. On the other hand, most P2P-based DDoS attacks studied so far are not DRDoS, and are mainly focused on deceiving innocent users into flooding messages toward a victim. To do so, they either trick peers into adding bogus neighbors in their routing tables [14, 15], or provide bogus index entries that a victim owns popular contents [14–19]. While such DDoS attacks have been found in the wild [20], they only generate tens of megabytes of UDP traffic or tens of hundreds of TCP connections per second at most.

*3) Defenses Against DRDoS:* Research on DRDoS defenses can be classified into two categories, network source address validation and traffic scrubbing. The first category includes BCP 38 [21], ITRACE [22], SPIE [23], Hop-Count Filtering [24] and Passive IP traceback [25], SAVE [26], and Ehrenkranz *et al.* [27] surveyed and evaluated the current state of IP spoofing defense. The later includes the statistics-based approach[28], SIFF [29], AITF [30] and StackPi [31]. These solutions, however, only have limited efficacy against DRDoS unless they can be fully deployed.

## III. Store-and-flood DRDoS

In this section, we present the methodology of the store-and-flood DRDoS (SF-DRDoS) attack and then analyze its characteristics. We build a model to evaluate its attack ability, with an emphasis on the AF the attacker can achieve.

### A. Methodology

As shown in Fig. 2, an SF-DRDoS attack typically consists of three stages:

1) Preparing stage: In this step, the attacker prepares data to store at reflectors. To do so, the data must follow the protocol of the exploited services to appear legitimate. Furthermore, the attacker needs to consider how the data may maximize the amplification factor. In P2P networks, for example, such data can be an index entry that contains an extremely long filename.

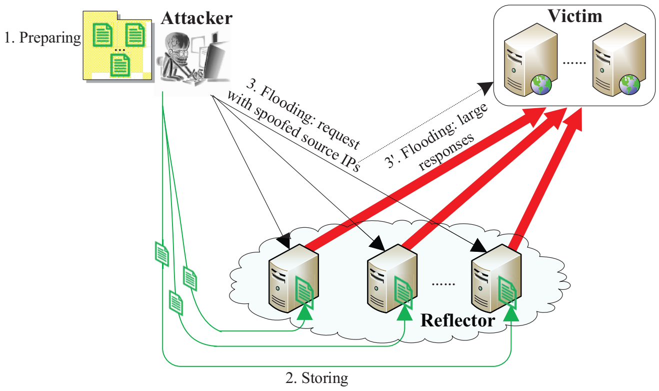

1. Preparing
Attacker
Victim
3. Flooding: request with spoofed source IPs
3'. Flooding: large responses
Reflector
2. Storing

Fig. 2. The working mechanism of a store-and-flood DRDoS attack.

2) Storing stage: Now that the data is prepared, the attacker must store the data at reflectors. Since the data follows the protocol of the exploited service, storing the data will likely go undetected. For example, in P2P networks, the storing process can be as simple as storing index entries in selected peers. The attacker must be mindful of the data's expiration time to ensure that the data will be available during the next stage.
3) Flooding stage: With data stored at reflectors, the attacker can then trigger flooding traffic toward her victim. To do this, the attacker sends reflectors requests for the previously stored data with the source IP address of the victim. Since the requests appear to be from the victim, the reflectors will send all the responses to the victim. The more requests the attacker generates, the more responses the victim will receive, the less available the victim will become to its legitimate users.

Due to the storing stage before flooding, the AF of the SF-DRDoS attack can be much higher than that of the traditional DRDoS attacks. In traditional DRDoS attacks, the attacker depends on data already stored at reflectors. If there is nothing or little data related to a request, the AF could not possibly be as high as desired. Conversely, in SF-DRDoS, because carefully prepared data are stored on reflectors in advance, the attacker can customize every request to generate a large response, thus significantly increasing the AF.

The SF-DRDoS attack offers great flexibilities to the attacker. First, the attacker can configure how many reflectors to use, how much data to store at each reflector, and how large each response will be. Second, the attacker can also control the timing of its attack, as it could determine when to begin and end the storing stage, when to trigger the flooding stage, and even whether to overlap different stages. Finally, the attacker can easily adjust the attack volume by controlling the number of requests and the size of responses.

### B. The Amplification Factor (AF) of SF-DRDoS

While AF is the ratio between the total attack traffic volume launched toward the victim, i.e., *attack ability*, and the total traffic volume invested by the attacker, i.e., *attack cost*, we further introduce two AF metrics to measure the potency of an SF-DRDoS attack: the **attack-time AF** where the attack cost is only the cost during an attack, and the **all-time AF** where the attack cost includes *all* the cost that the attacker invests.

Assume that sending a request of size $s$ will cause a reflector to retrieve the attacker's stored data and generate a response of size $r$. The attack-time AF will simply be:

$$\text{Attack-time AF} = \frac{r}{s}. \tag{1}$$

While traditional DRDoS attacks can be measured using attack-time AF, it is also important to use all-time AF for SF-DRDoS where the cost of storing data on reflectors may be non-negligible. Assume that the attacker must use $s'$ worth of traffic volume to store data at each reflector. If each reflector can be used $t$ times after having data stored at it, then

$$\text{All-time AF} = \frac{r \cdot t}{s' + s \cdot t}. \tag{2}$$

The all-time AF represents the actual ratio of victim resource usage to attacker resource usage. The attack-time AF is more indicative of what attacks will be achievable by an attacker. While an attacker may have a moderate all-time AF, it may have a high enough attack-time AF to disable a victim for a short period of time with a relatively low amount of bandwidth. In effect, the attacker is able to pay for preparing the attack over a longer period of time, then suddenly launch a large attack which the victim cannot handle all at once.

## IV. Store-and-Flood DRDoS on Kad

In this section, we introduce a Kad-based store-and-flood DRDoS attack. First we give an overview of the Kad system, with the emphasis on its characteristics that the SF-DRDoS attack will exploit, then we describe the design and implementation of this attack.

### A. Kad

Kad is a P2P file-sharing network using the Kademlia [32] distributed hash table protocol. In Kad, every participating node has a unique 128-bit identifier called *Kad ID*. Kad supports two types of objects, keywords and files. Every keyword is associated with a 128-bit *key ID*, which is the hash of the keyword, and every file is assigned a 128-bit *file ID*, which is the hash of the file's content. In the 128-bit ID space, Kad calculates the distance of two IDs using bitwise $XOR$ operation. Kad supports two primary operations: *publish* and *search*. A node can publish a *keyword-to-file* index at nodes whose Kad ID is closest to the *key ID* of the keyword, and allow other nodes to search the index using the key ID.

Kad has the following characteristics that a store-and-flood DRDoS attack can exploit:

- All Kad operations are UDP-based, and they do not have handshaking mechanisms at the application level either, thus making IP spoofing easy.
- Kad has two million concurrent users [5] and all of them could be reflectors.

- Kad can provide a large amplification effect. Table I shows the amplification effects that some operations in Kad can already achieve without much elaboration. Moreover, one search request can trigger multiple response packets that collectively encompass 300 indices.
- In Kad it is easy to manipulate the size of response packets, the UDP ports used by Kad are not fixed (in order to avoid censorship), and messages are encrypted and obscured, all making traffic filtering difficult.

TABLE I
AMPLIFICATION FACTORS OF KAD OPERATIONS BASED ON EXPERIMENTS.

| Operation | Request (bytes) | Response (bytes) | AF |
|---|---|---|---|
| Bootstrapping | 64 | 480 | ∼ 8 |
| Routing | 77 | 111 ∼ 336 | 1 ∼ 5 |
| Searching an unpopular keyword | 64 | 230 | ∼ 4 |
| Searching a popular keyword | 64 | 27493 | ∼ 350 |
| Searching an unpopular file | 70 | 260 | ∼ 4 |
| Searching a popular file | 70 | 12000 | ∼ 160 |

### B. Design of Kad-based SF-DRDoS

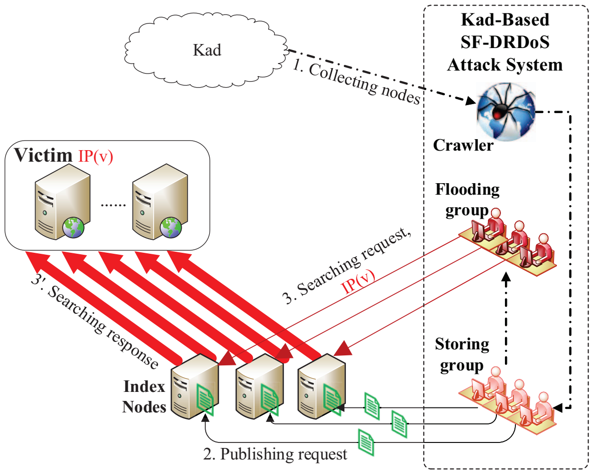


Fig. 3. The design of Kad-based store-and-forward DRDoS attack.

A Kad-based SF-DRDoS attack consists of three basic components: the crawler, the storing group, and the flooding group. As shown in Fig. 3, the crawler can periodically crawl the entire Kad network to collect Kad nodes online and provide a list of them to the storing group. Every online node is a potential reflector represented by a tuple (node ID, IP, UDP port, Kad version). The storing group, upon the receipt of a list of online nodes, prepares 300 large index entries—i.e., the preparing stage of an SF-DRDoS attack, and stores them to each of these nodes—i.e., the storing stage of an SF-DRDoS attack. For each node, the storing group first selects an appropriate keyword which has a key ID that shares at least the first 8 bits (i.e., prefix) with the node's Kad ID, then constructs 300 keyword-to-file indices all with the keyword and a random, extremely long string for the filename field, and finally publishes all these indices. These indices, once stored, will stay valid for 24 hours, and this storing process can be repeated every 24 hours to support a persistent attack.

When it is the time to launch the attack as desired by the attacker, the flooding group can then issue a large number of search requests to look for the keywords for which the storing group has stored indices in Kad. All these requests carry the spoofed source IP address of a victim (IPs$_v$ in Fig. 3). Due to the lack of any handshaking mechanism in Kad, these reflectors cannot verify the authenticity of the source IP. Every request is able to trigger 300 index entries, thus generating massive response packets toward the spoofed source IP address. Consequentially, the links at the victim will be clogged and the victim will be successfully DDoS'ed.

### C. Implementation of Kad-based SF-DRDoS

We implemented the Kad-based SF-DRDoS attack system using aMule, an open-source application which works with Kad. We created a customized client with only the necessary Kad components, and using this client we spawned many Kad nodes to perform the crawling, storing, and flooding. The Kad crawler is a specific implementation of our crawling algorithm proposed in [33], which according to our experiments can gather over 2 million nodes in about 3 minutes. However, since we use a single crawler, rather than a distributed crawler as proposed in [33], our crawler takes longer. Also, we limit the crawler to only collect nodes which can be used as reflectors, e.g., those that do not use firewalls or NAT.

For the storing process, an important parameter is the maximum length of the filename in keyword-to-file indices. When a filename exceeds 1990 bytes, it is not accepted in Kad. We use a smaller size to avoid having any filenames dropped or rejected.

Kad has a flooding control mechanism to limit the rate of request packets from a specific source IP address. Once the request rate from a node exceeds the limit, its requests will be dropped and eventually its IP address will be blacklisted temporarily. For example, every node can issue at most *three* search or publish requests every minute. However, during the storing stage, every publish request can use a different spoofed source IP address, or that of a different bot each time in a large-scale botnet, to stay below the rate limit. This limitation is therefore only relevant to the flooding stage of the SF-DRDoS attack where every request must use one of the IP addresses of the victim. Therefore, in the flooding stage for every IP address of the victim, this attack will send every node three search requests per minute to maximize the usage of each node as a reflector. If the victim has many IP addresses, such as when an entire subnet is targeted, a lot more search requests can be issued.

## V. EXPERIMENT RESULTS AND ANALYSIS

We now describe our experiments with the Kad-based SF-DRDoS attack, results from the experiments, and our analysis.

### A. Experiment Setup

We deploy the attack system on a server with two Intel Xeon CPUs (E5645, 2.40GHz, 24GB RAM) located on a university campus. It spawns 40 customized P2P nodes for the storing group as well as 40 for the flooding group. Each node uses an independent IP address to directly connect to the Internet. All nodes are distributed evenly across the Kad ID space. Every node has a 4 Mbps uplink, rented from a special network where BCP 38 [21] is not deployed, allowing every node to spoof their source IP address.

We conducted a Kad-based experimental attack in May 2013 under real-world conditions toward a victim with a dedicated download bandwidth of 1 Gbps at another participating university. We carefully controlled the attack traffic volume to avoid causing any unexpected network failures. During the experiment, the crawler collected Kad nodes (i.e., reflectors) online from two 8-bits ID zones (0x51 and 0x2E) on an hourly basis, and was able to continuously provide about 20,000 live nodes for the storing group and the flooding group. The storing group then published 300 keyword-to-file index entries to each node collected, where the length of every filename was 1500 bytes. The storing process lasted 30 minutes; as the average Kad node stays online for 165 minutes [34], the 30-minute duration is appropriate to ensure enough nodes remain available with the stored data during the flooding stage. We then launched the flooding stage that lasted for 15 minutes, during which we recorded all attack traffic towards the predefined UDP port at the victim machine. For every node used as a reflector, because it stays online for 165 minutes on average and can accept 3 search requests per minute from each source address, assuming there are $|V|$ different IP addresses of the victim, the node can then receive totally $165 \cdot 3 \cdot |V|$, i.e., $495|V|$, requests.

### B. Environmental Parameters

Two environmental parameters are important to our experiments: the size of the Kad network and the uplink bandwidth of Kad nodes. The former determines the total number of usable reflectors, and the latter is the upper limit of the DRDoS traffic on an individual reflector. We thus estimated both as follows.

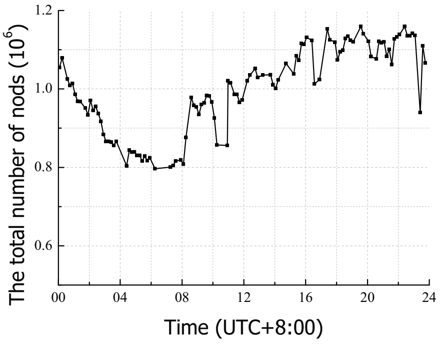


Fig. 4. The total number of Kad nodes simultaneously online on June 22, 2013.

To determine the size of the Kad network, we ran our crawler continuously for 24 hours. The crawler could collect all nodes in the Kad network in an average of 15 minutes. Fig. 4 presents the total number of usable reflectors in Kad simultaneously. Though the number fluctuates over time, it is at least 0.8 million all the time and is often over 1 million, sufficient for conducting a powerful SF-DRDoS attack.

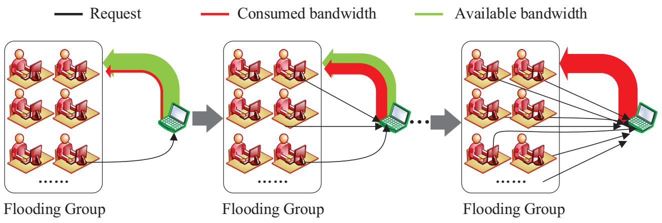


Fig. 5. The schematic diagram of the bandwidth estimation technique.

For the uplink bandwidth of Kad nodes, we developed a customized bandwidth estimation method based on the self-loading periodic streams (SLoPS) technique [35], with its results shown in Fig. 5. Our assumption is that senders will not receive responses from a receiver once the available uplink bandwidth of the receiver is all used up, and the uplink bandwidth of a node is thus the maximum bandwidth consumed by all the response packets to all the senders. After storing enough keyword-to-file index entries at a Kad node, we set up 170 flooding nodes to send searching requests with gradually increasing rates directly to the Kad node's IP, and then measure the maximum response bandwidth. Because Kad does not limit the total amount of traffic used in Kad communications, this bandwidth is exactly the uplink bandwidth of the node in question. Fig. 6 presents the distribution of the uplink bandwidth for 2150 randomly chosen Kad nodes.

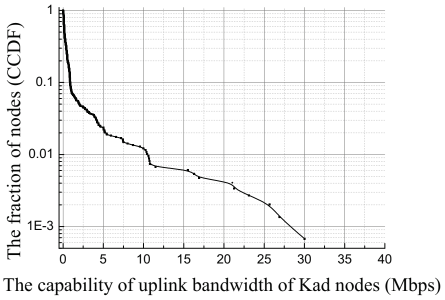
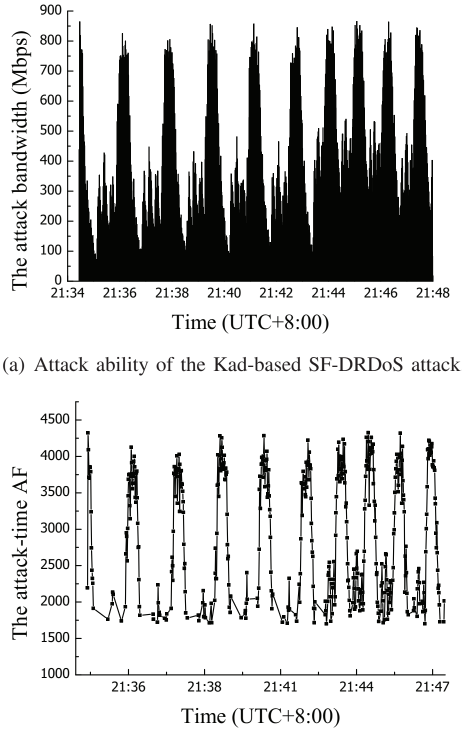


Fig. 6. The distribution of uplink bandwidth of Kad nodes.

### C. Results and Analysis

We now present the results from our experiments under real world conditions. Fig. 7(a) presents the attack bandwidth toward the victim over a 15-minute attack window. The peak reaches 865 Mbps, and the average is 480 Mbps. Meanwhile, as the average attack cost during the attack is only 0.2 Mbps, the average attack-time AF is 2400. Fig. 7(b) further presents the attack-time AF throughout the attack window, where the maximum AF is 4326. Note that due to node overloading and packet loss, this maximum is lower than the theoretical maximum, which is 5980 for 1500-byte filenames, and if we use 1900-byte filenames the maximum AF can be even higher, with a theoretical maximum of 9548. However, the measured AF value of 4326 is already significantly large. In a previous DDoS attack using Kad [14], it utilizes bootstrap requests and gains an AF of only about 8.



(a) Attack ability of the Kad-based SF-DRDoS attack.

(b) Attack-time AF of the Kad-based SF-DRDoS attack.

Fig. 7. Attack ability and AF in the Kad-based SF-DRDoS attack conducted on May 21st, 2013.

The all-time AF for this attack can also be quite high. The storing cost, $s'$, for 300 index entries each with a filename taking 1500 bytes will be $1500 \cdot 300$ bytes. Considering the flooding control mechanism in Kad, in order to fully utilize the uplink bandwidth of reflectors, the target area must have 11 IP addresses. Assuming that the number of IP addresses in the target area $|V|$ is 11, then the number of times a reflector can be used, $t$, is $495 \cdot 11$. The size of a request to a reflector, $s$, is 64 bytes and this request will generate a response of size $r = 300 \cdot 1500$ bytes. Now, referring to Equation 2 in Section III-B, we have:

$$\text{All-time AF} = \frac{(300 \cdot 1500) \cdot (495 \cdot 11)}{(300 \cdot 1500) + 64 \cdot (495 \cdot 11)} = 3069. \quad (3)$$

Now we estimate the total attack ability of the Kad-based SF-DRDoS attack and the corresponding attack cost. The number of usable reflectors in Kad is approximately one million, of which the average uplink bandwidth is 0.67 Mbps. We use the attack-time AF of 2400 from our experiment. Assuming the attacker can utilize her reflectors with the average reflector uplink bandwidth, the maximum attack ability would be approximately 670 Gbps (i.e., 0.67 Mbps multiplied by one million), sufficient to disable most web sites on the Internet. Meanwhile, the attack cost at the flooding stage would be just 280 Mbps, easy for an attacker to obtain.

## VI. Detection and Defenses

In this section, we propose possible methods for detection and defense against SF-DRDoS attacks.

### *A. Detection*

By adopting random ports and encrypted payloads, Kad-based SF-DRDoS attacks are hard to detect. We propose that we can deploy enough honey nodes on Kad to detect such attacks. These honey nodes would act just the same as other normal nodes, and each honey node each can keep statistics about publishing and searching events that the honey node is involved in. According to features of store-and-flood attacks, it is easy to detect abnormal behaviors such as frequently publishing many large index entries (with long filenames) and searching them at a high rate afterwards (i.e., acting as an attacking node), or seeing large index entries published at the honey node itself and then subsequent search requests for the same index entries (i.e., acting as a reflector node).

### *B. Defenses*

We make the following recommendations to defend against SF-DRDoS. First, Kad should make some changes, including answering requests only after validating their sources, limiting the string length and the number of index entries triggered by one request. However, it is difficult to deploy these incompatible modifications in the current Kad network.

Next, as recommended in BCP 38 [21], every ISP should deploy ingress filtering to eliminate the possibility of source IP spoofing as well as reflection attacks, including SF-DRDoS. Ingress filtering requires that when an IP packet departs from a network to enter the Internet, it must carry a source IP address belonging to the ISP. Unfortunately, while nearly 80% of the Internet deploys some type of ingress filtering, the remaining 20% are reluctant to deploy ingress filtering due to technical and economic reasons [36].

Then assuming IP spoofing will not be eliminated in the foreseeable future, we believe that effective traffic filtering is key to defending against SF-DRDoS attacks. Though the design of a comprehensive filtering system is beyond the scope of this paper, we provide an overview of what such a system would require in order to succeed, and give operational examples of such a system based on BGP flow specification [8].

Before discussing challenges in designing a comprehensive system, we consider a scenario in which a TCP server would like to defend itself against SF-DRDoS attacks. Designing a flow specification in this scenario is simple: The TCP server can send a rule to its switch asking that all UDP packets be dropped. The switch then propagates the rule further upstream. This should mitigate any UDP-based SF-DRDoS attacks.

A more comprehensive system, meant to defend systems which cannot afford to simply block all UDP traffic, must be able to effectively deploy filtering rules which can distinguish malicious traffic from legitimate traffic. Such a system would need to either automatically generate suitable rules or listen for rules from a client and then reliably propagate said rules to upstream links. Fig. 8 shows an example of such a system based on flow specification propagating rules to filter packets exceeding a set size.

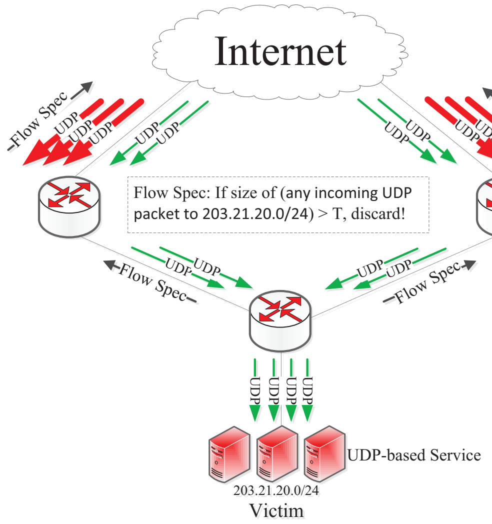


Fig. 8. An example of protecting UDP-based services using BGP flow specification.

## VII. Discussion and Open Issues

To exploit *all* nodes in the Kad network, the attacker would need a bandwidth of nearly 280 Mbps. While this cost may seem prohibitive at first, with sufficient funds, attackers may achieve a high bandwidth by renting botnets. In recent years, botnet attacks have become quite prevalent [37–39], and botnets can be as cheap as $100 for 10,000 bots [40]. Furthermore, botnets with hundreds of thousands of bots are readily employable. With such low costs and a high volume, attacks exploiting every node on a network may be feasible from both a technical and economic standpoint.

Also, the store-and-flood DRDoS attack is not limited to P2P file-sharing networks, such as Kad. A thorough examination of all public UDP protocols is beyond the scope of this paper, however, the same methodology we used could be applied to develop attacks on other UDP protocols. Proprietary UDP protocols for which specifications are not publicly available could be vulnerable to this type of attack.

This paper presents the SF-DRDoS attack and points to upstream filtering as a promising defense. However, the creation of such a system is left as future work. Such a system would need to meet the following demands: (i) it must be able to produce high quality filtering rules; (ii) it must scale to be able to handle many rules from many servers; and (iii) it must be easily deployable. A system meeting these requirements would not only provide defense against SF-DRDoS attacks, but could be generalized to help mitigate any type of DDoS.

## VIII. Conclusion

Distributed denial of service (DDoS) attacks have been around for many years, yet they continue to pose a serious threat to the security of the Internet. Worse, as the Internet adds new applications with very large user bases, the soil for DDoS becomes ever more fertile, sometimes even leading to new, more devastating DDoS attacks.

We elucidate one such new DDoS attack in this paper, which we call *store-and-flood* distributed reflective denial of service attack, or SF-DRDoS attack. The attack can leverage popular peer-to-peer (P2P) applications to flood a victim with an unusually large amplification factor. The attacker can first store carefully prepared data on a large amount of P2P nodes, and then issue specially prepared requests to these nodes to generate responses toward innocent victims. The timing, content, and in particular the volume of the responses are all under the control of the attacker.

We implemented a prototype of SF-DRDoS on the Kad peer-to-peer network, and conducted real-world experiments. Compared with the state-of-the-art amplification factor in DRDoS attacks, the Kad-based SF-DRDoS can achieve a much higher attack-time amplification factor of 2400 on average, with an attack bandwidth as high as 670 Gbps—sufficient to take down most web sites on the Internet

We further discussed defenses against SF-DRDoS attacks, including injecting honey nodes into Kad, deploying BCP 38, and employing BGP flow specification. Together with other DDoS attacks, SF-DRDoS attacks signal the urgent need for new, effective defense solutions.

## References

[1] The largest DDoS attack in history. http://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet.

[2] M. Zhang, M. Dusi, W. John, and C. Chen, “Analysis of UDP traffic usage on internet backbone links,” in *Proceedings of 9th Annual International Symposium on Applications and the Internet (SAINT)*, 2009.

[3] The eMule Project. http://www.emule-project.net.

[4] B. Cohen, “Incentives build robustness in BitTorrent,” in *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, 2003.

[5] M. Steiner, E. W. Biersack, and T. Ennajjary, “Actively monitoring peers in Kad,” in *Proceedings of the 6th International Workshop on Peer-to-Peer Systems (IPTPS)*, 2007.

[6] J. Yu, P. Xiao, Z. Li, and Y. Zhou, “Toward an accurate snapshot of DHT networks,” *IEEE Communications Letters*, vol. 15, no. 1, pp. 97–99, 2011.

[7] C. Rossow, “Amplification hell: Revisiting network protocols for DDoS abuse,” in *the 21th Annual Network & Distributed System Security Symposium (NDSS)*, 2014.

[8] P. Marques, “Dissemination of flow specification rules,” in *RFC 5575*, August 2009.

[9] Open Resolver Project. http://openresolverproject.org.

[10] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 3, pp. 38–47, 2001.

[11] CERT advisory CA-1998-01 smurf IP denial-of-service attacks. http://www.cert.org/advisories/CA-1998-01.html.

[12] S. Kumar, "Smurf-based distributed denial of service (DDoS) attack amplification in Internet," in *the Second International Conference on Internet Monitoring and Protection (ICIMP)*, 2007.

[13] Distributed reflection denial of service. http://www.understandingcomputers.ca/articles/grc/drdos_copy.html.

[14] J. Yu, Z. Li, and X. Chen, "Misusing Kademlia protocol to perform DDoS attacks," in *International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, 2008.

[15] N. Naoumov and K. Ross, "Exploiting P2P systems for DDoS attacks," in *Proceedings of the 1st international conference on Scalable information systems*, 2006.

[16] E. Athanasopoulos, K. G. Anagnostakis, and E. P. Markatos, "Misusing unstructured P2P systems to perform DoS attacks: The network that never forgets," in *Applied Cryptography and Network Security*, 2006.

[17] K. El Defrawy, M. Gjoka, and A. Markopoulou, "BotTorrent: misusing BitTorrent to launch DDoS attacks," in *Proceedings of the 3rd USENIX workshop on Steps to reducing unwanted traffic on the internet*, 2007.

[18] X. Sun, R. Torres, and S. Rao, "DDoS attacks by subverting membership management in P2P systems," in *3rd IEEE Workshop on Secure Network Protocols (NPSec)*, 2007.

[19] K. C. Sia, "DDoS vulnerability analysis of BitTorrent protocol," University of California, Los Angeles., Tech. Rep., 2007.

[20] Z. Li, A. Goyal, Y. Chen, and A. Kuzmanovic, "Measurement and diagnosis of address misconfigured P2P traffic," in *Proceedings of IEEE INFOCOM*, 2010.

[21] P. Ferguson, "Network ingress filtering," in *RFC 2827*, May 2000.

[22] ICMP traceback messages. http://tools.ietf.org/html/draft-ietf-itrace-04.

[23] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-based IP traceback," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 3–14, 2001.

[24] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed IP traffic using hop-count filtering," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 1, pp. 40–53, 2007.

[25] G. Yao, J. Bi, and Z. Zhou, "Passive IP traceback: capturing the origin of anonymous traffic through network telescopes," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 413–414, 2010.

[26] J. Li, J. Mirkovic, M. Wang, P. L. Reiher, and L. Zhang, "SAVE: Source address validity enforcement protocol," in *Proceedings of IEEE INFOCOM*, 2002.

[27] T. Ehrenkranz and J. Li, "On the state of ip spoofing defense," *ACM Transactions on Internet Technology (TOIT)*, vol. 9, no. 2, pp. 6:1–6:29, 2009.

[28] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to DDoS attack detection and response," in *Proceedings of 2003 DARPA Information Survivability Conference and Exposition*, 2003.

[29] A. Yaar, A. Perrig, and D. Song, "SIFF: A stateless internet flow filter to mitigate DDoS flooding attacks," in *Proceedings of 2004 IEEE Symposium on Security and Privacy*, 2004.

[30] K. J. Argyraki and D. R. Cheriton, "Active internet traffic filtering: Real-time response to denial-of-service attacks." in *USENIX Annual Technical Conference*, 2005.

[31] A. Yaar, A. Perrig, and D. Song, "StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 10, pp. 1853–1863, 2006.

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

[33] B. Liu, S. Wu, T. Wei, C. Zhang, J. Li, J. Zhang, Y. Chen, and C. Li, "Splider: A split-based crawler of the BT-DHT network and its applications," in *11th Annual IEEE Consumer Communications & Networking Conference (CCNC)*, January 10–13 2014, p. 9 pages.

[34] M. Steiner, T. En-Najjary, and E. W. Biersack, "Long term study of peer behavior in the KAD DHT," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1371–1384, October 2009.

[35] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, 2002, pp. 295–308.

[36] R. Beverly, A. Berger, Y. Hyun *et al.*, "Understanding the efficacy of deployed internet source address validation filtering," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, 2009.

[37] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[38] F. C. Freiling, T. Holz, and G. Wicherski, "Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks," in *Proceedings of the 10th European conference on Research in Computer Security*, 2005.

[39] D. Dagon, G. Gu, C. P. Lee, and W. Lee, "A taxonomy of botnet structures," in *the Twenty-Third Annual Computer Security Applications Conference (ACSAC)*, 2007.

[40] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, "Measuring pay-per-install: The commoditization of malware distribution." in *USENIX Security Symposium*, 2011.