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

Bingshuang Liu* Skyler Berg† Jun Li†
Tao Wei‡ Chao Zhang‡ Xinhui Han*

*Peking University, China

†University of Oregon, USA

‡University of California, Berkeley, USA

August 5, 2014

Table of Contents

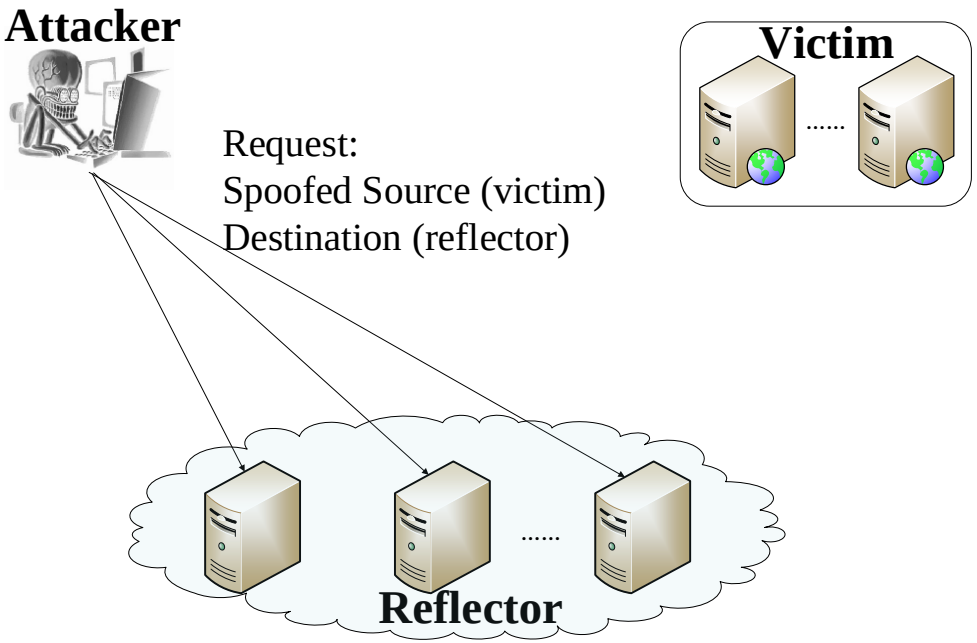
A pervasive threat on the Internet

- Denial of Service (DoS) attacks create heavy traffic to overwhelm services
- Distributed Denial of Service (DDoS) attacks are carried out by multiple machines
- There are 28 DDoS attacks per hour

DRDoS becomes increasingly potent

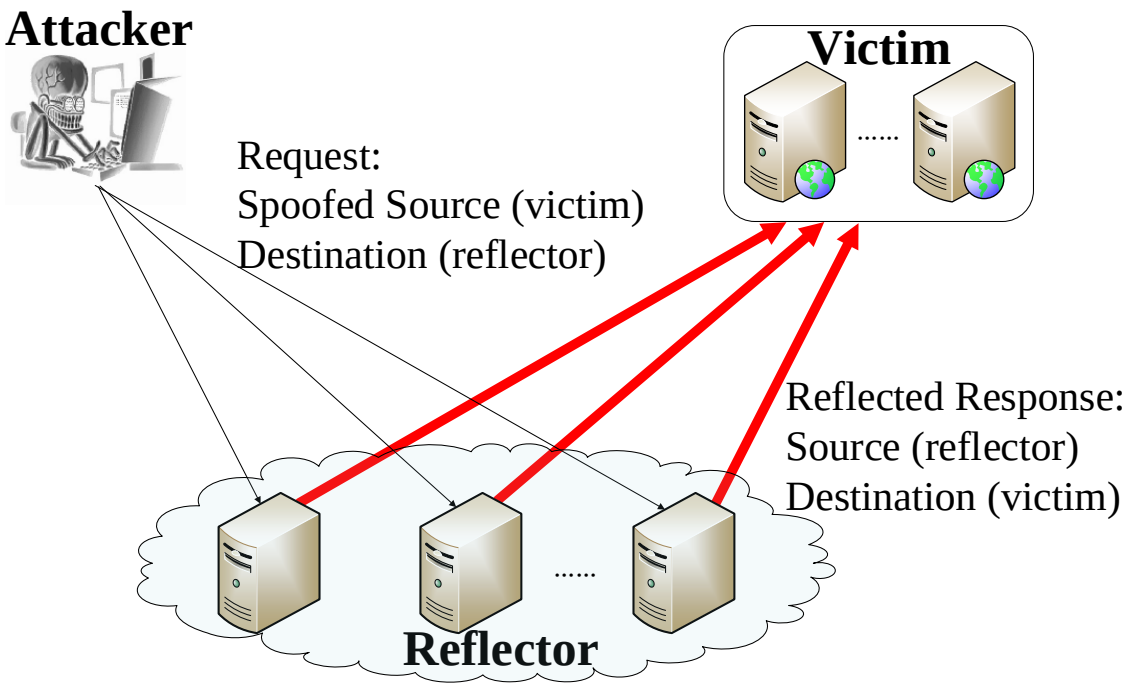
- Distributed Reflective Denial of Service (DRDoS) Attacks
- Higher volume of attack traffic than conventional DDoS
- Attacker gains anonymity

The DRDoS attack



- Attacker identifies a vulnerable service
- Attacker sends requests to the service spoofing the IP to appear to be the victim

The DRDoS attack cont'd



- Service sends response to the victim
- Victim receives response it never asked for, causing congestion

What the attacker gains

- The victim never learns the IP of the attacker
- Attacker can appear to be a diverse array of machines by exploiting a service with many hosts
- The service may respond with a message larger than the attacker's request, creating an amplification effect

Amplification

- The attacker can use a small amount of traffic to generate a large attack
- Amplification factor (AF): The ratio between the volume of response packets and request packets
- AF is the key metric in evaluating the strength of a DRDoS attack

Characteristics of a vulnerable protocol

Vulnerable protocols will:

- be stateless to allow spoofing
- have abundant reflectors publicly accessible
- have small requests that can trigger large responses

Current trends in amplification

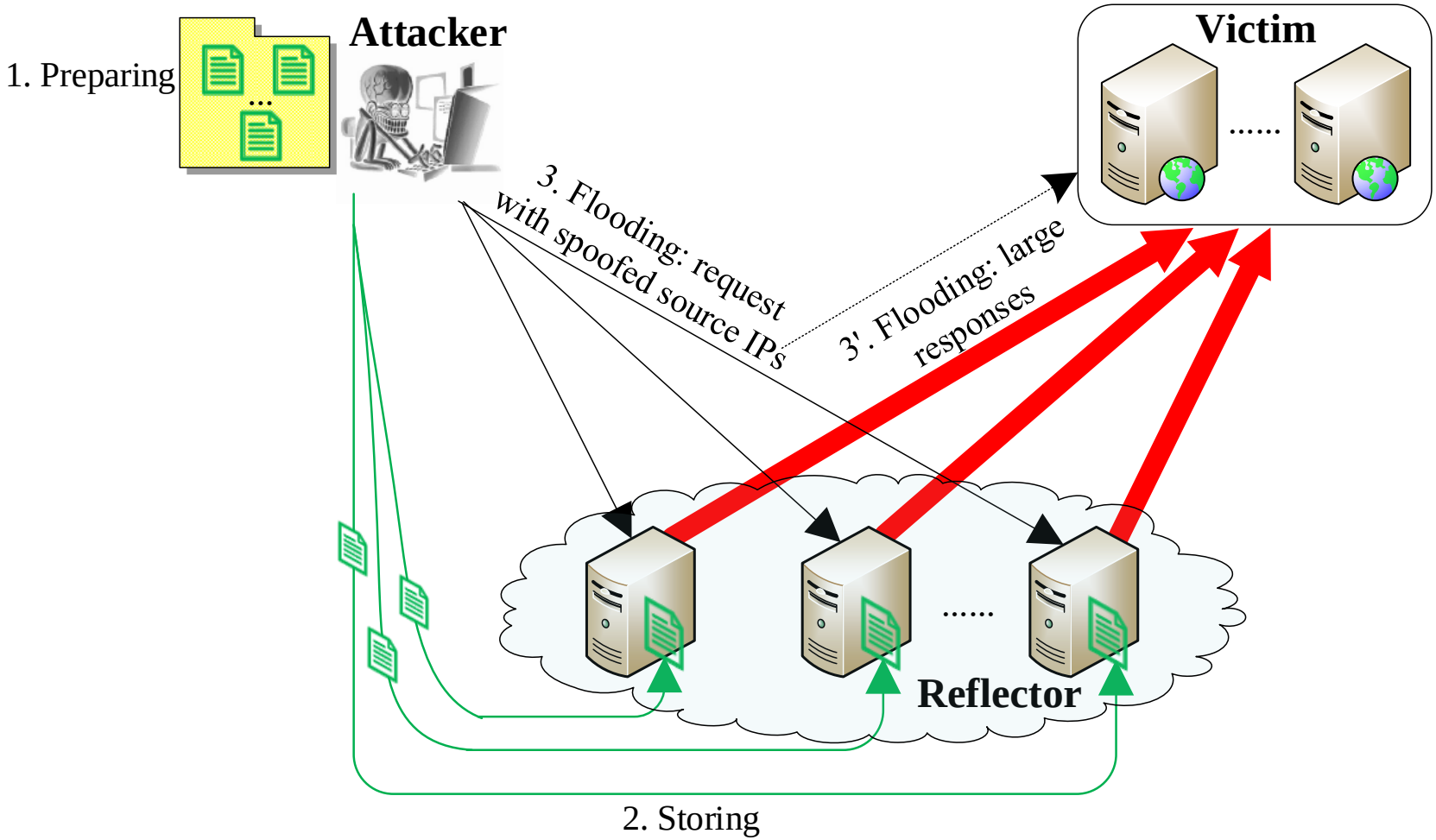
- 2013: Year of the DNS DRDoS (AF 64)
  - Spamhaus incident: 300 Gbps
- 2014: Year of the NTP DRDoS (AF 700)
  - CloudFlare incident: 400 Gbps
- 2015: Can Things Go even Worse?

Table of Contents

Introducing SF-DRDoS

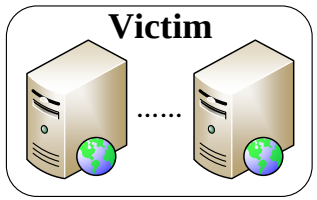
- Store-and-Flood Distributed Reflective Denial of Service attack
- An attack on protocols that allow users to store data (e.g., P2P networks)
- Three stage attack:
  - Preparation stage (obtaining or generating data)
  - Storing stage (storing data at reflectors)
  - Flooding stage (triggering reflected data to victim)

The SF-DRDoS attack



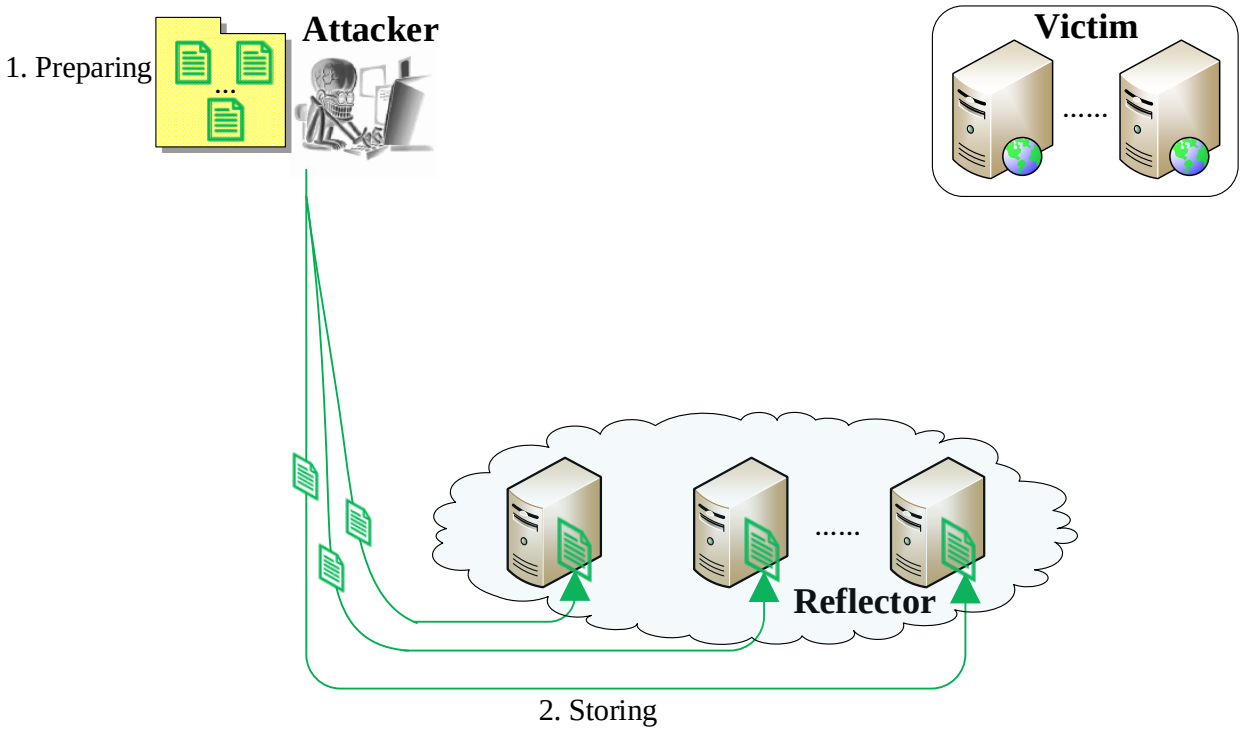
SF-DRDoS: Preparation stage

1. Preparing



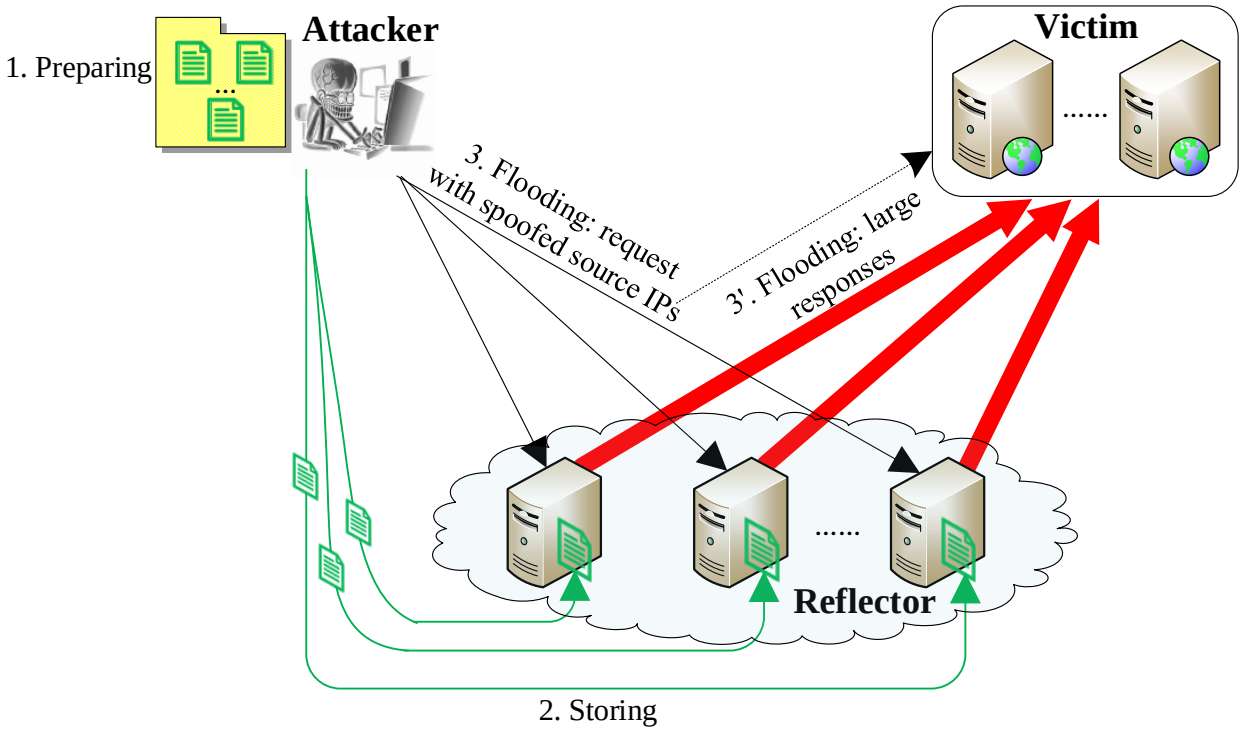
- Create data that conforms to the protocol
- Data should be voluminous compared to request
- Data should be accessible with a small request

SF-DRDoS: Storing stage



- Send prepared data to reflectors with storing requests
- Reflectors will accept storing requests because they conform to the protocol
- Must consider the data's expiration time

SF-DRDoS: Flooding stage



- Each reflector is primed with a perfect entry to generate a large AF
- Attacker can simply send spoofed requests for the stored data

Evaluating SF-DRDoS

- AF is not sufficient because it does not reflect the cost from all stages
- Flooding costs are well represented by traditional AF
- Preparation may incur some cost
- Storage incurs an upfront cost that provides the attacker with a resource until the data expires

Defining metrics

- $$\text{Attack-time AF} = \frac{r}{s}. \quad (1)$$

  where s is the request size and r is the response size at the flooding stage.

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

  where s' is the traffic volume to store data at each reflector and t is how many times these stored data can be used at the flooding stage.

Table of Contents

Kad-based SF-DRDoS

What makes a good SF-DRDoS service?

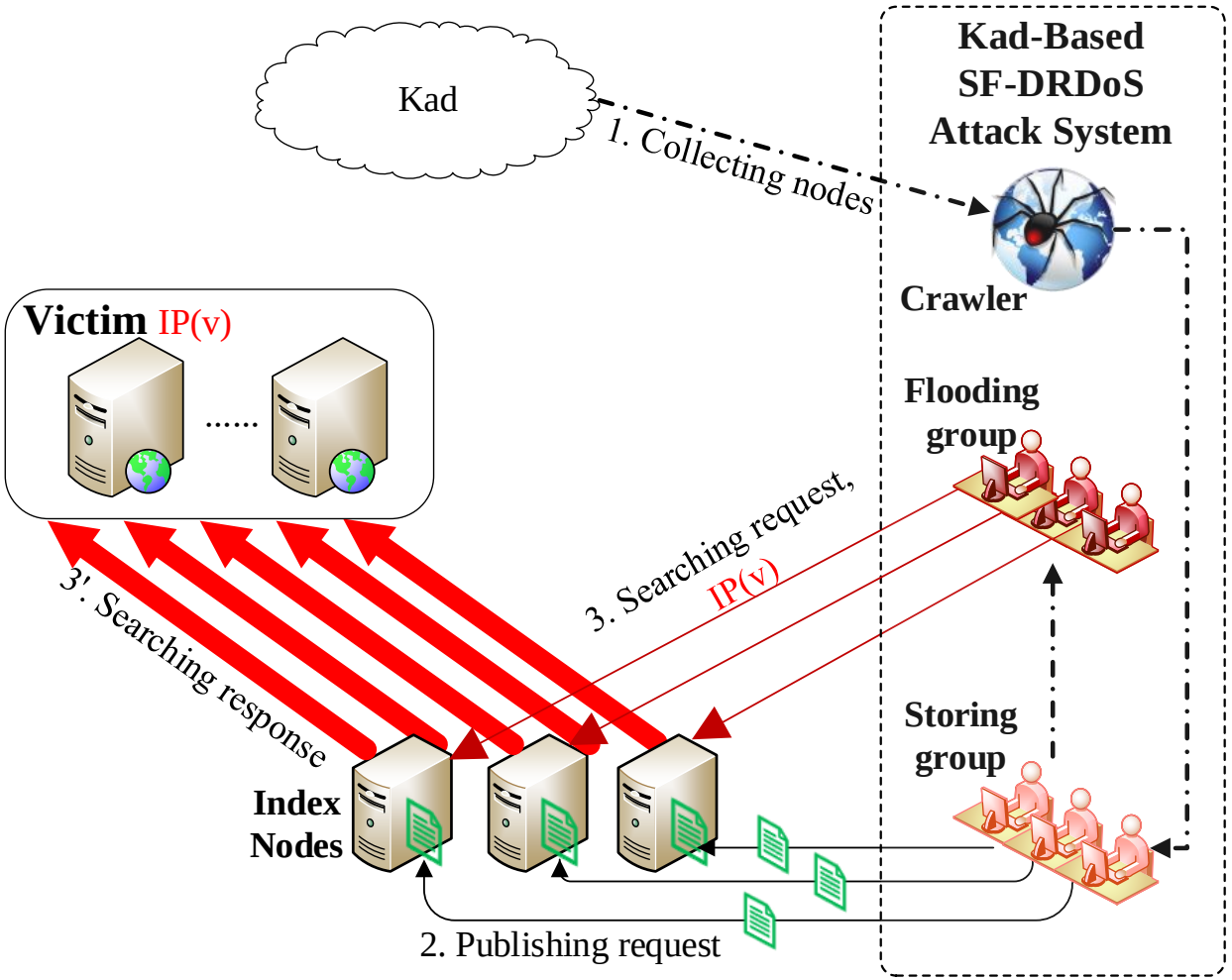
- Stateless
- Large user base
- Amplification
- Users can store content

What properties does Kad have?

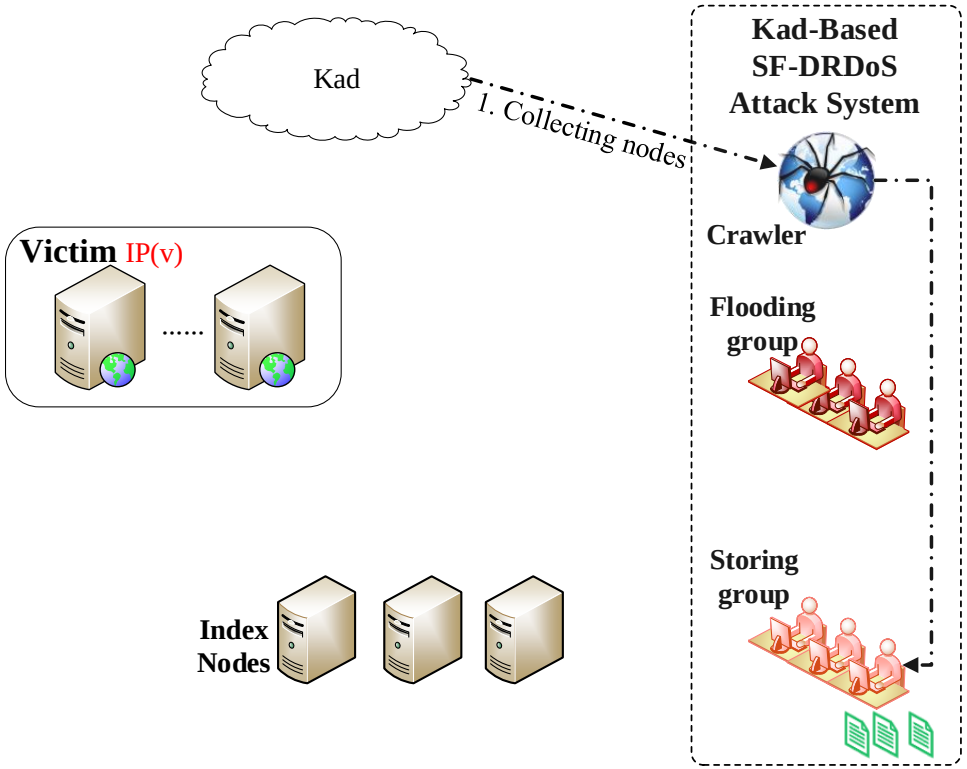
- UDP-based
- 2 million concurrent users
- 1 request: 300 responses
- P2P network

Other P2P networks could face similar vulnerabilities

Design of Kad-based store-and-flood DRDoS attack

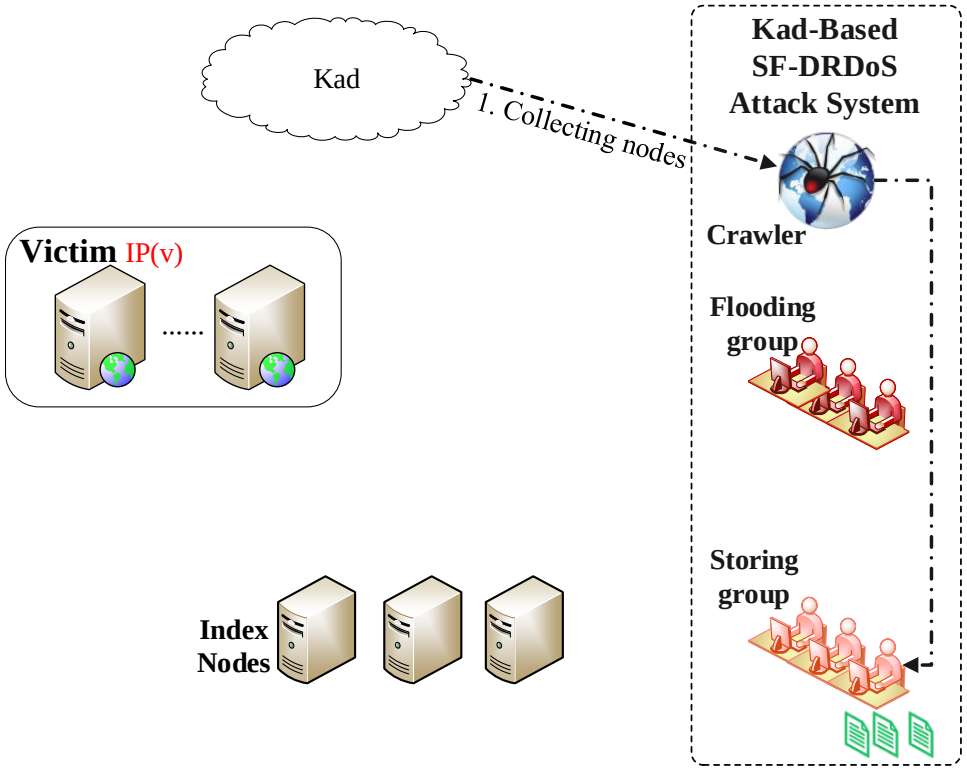


The preparation stage



- Kad nodes associate keywords with filenames indices
- Up to 300 indices can be returned when answering a request
- One keyword-to-file index can be up to 1990 bytes

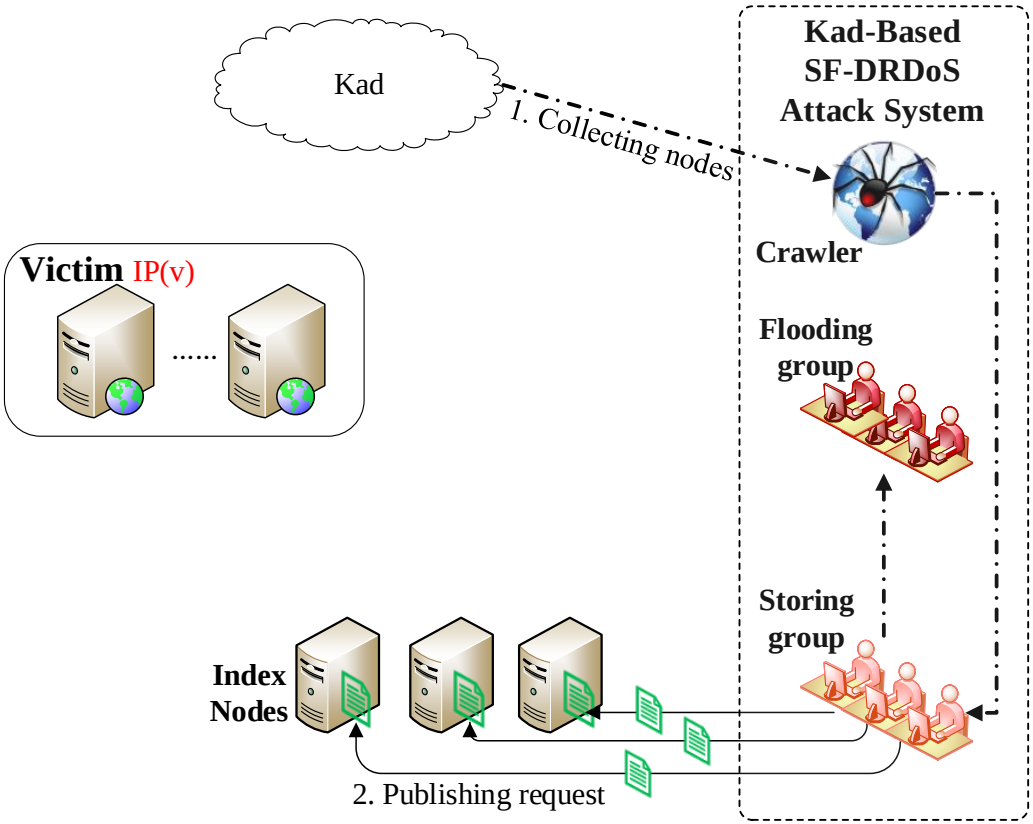
The preparation stage



- Kad nodes associate keywords with filenames indices
- Up to 300 indices can be returned when answering a request
- One keyword-to-file index can be up to 1990 bytes

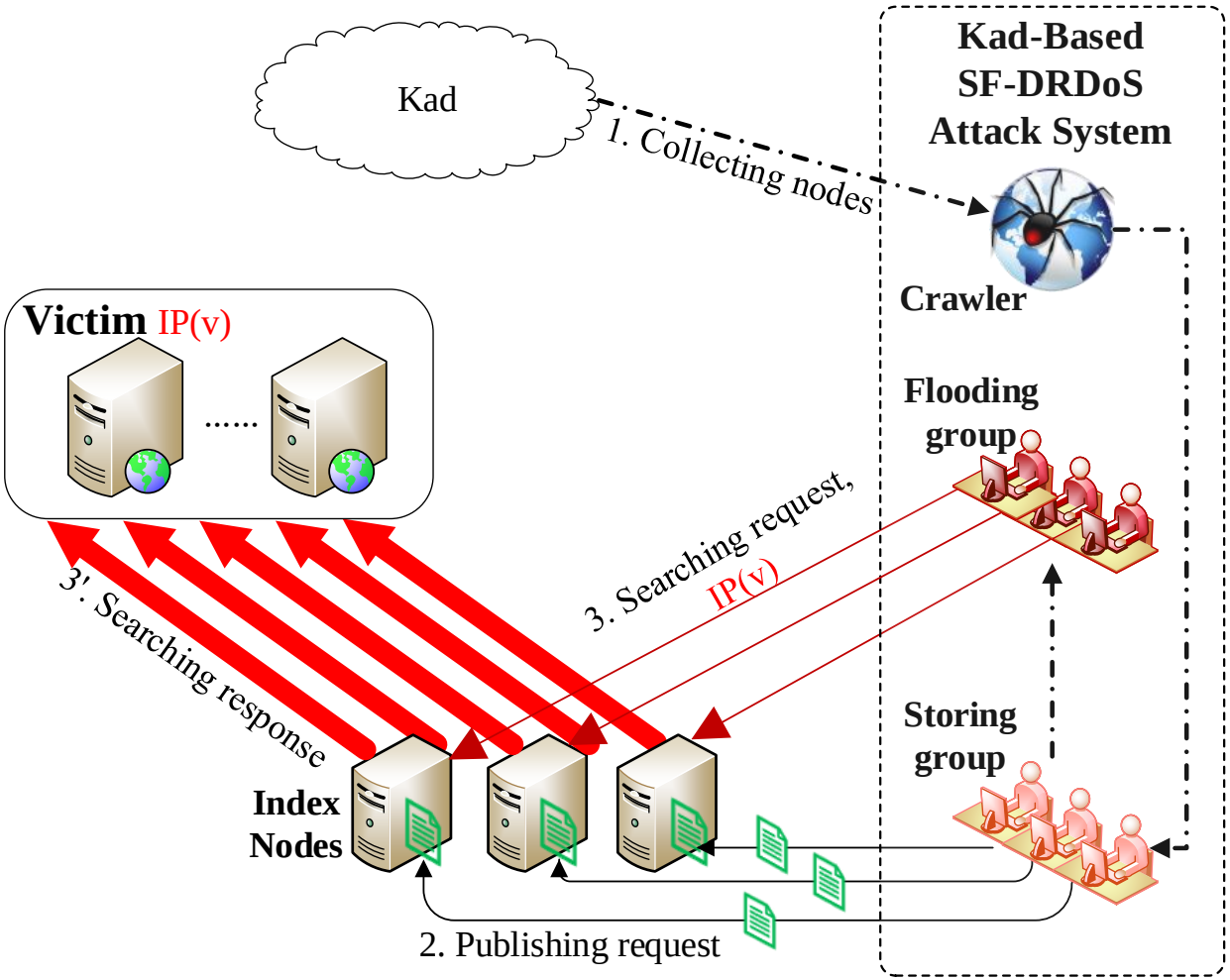
- Original index: $< Titanic, Titanic\ dvd.avi >$
- Make it larger: $< Titanic, Titanic\ asdf...wesf.avi >$

The storing stage



- We send the prepared indices to each Kad node
- Once stored, these indices are valid for 24 hours

The flooding stage



- Start to request indices associated with the specific keyword
- Use the victim IP
- Reflectors send 300 large packets for our 1 packet

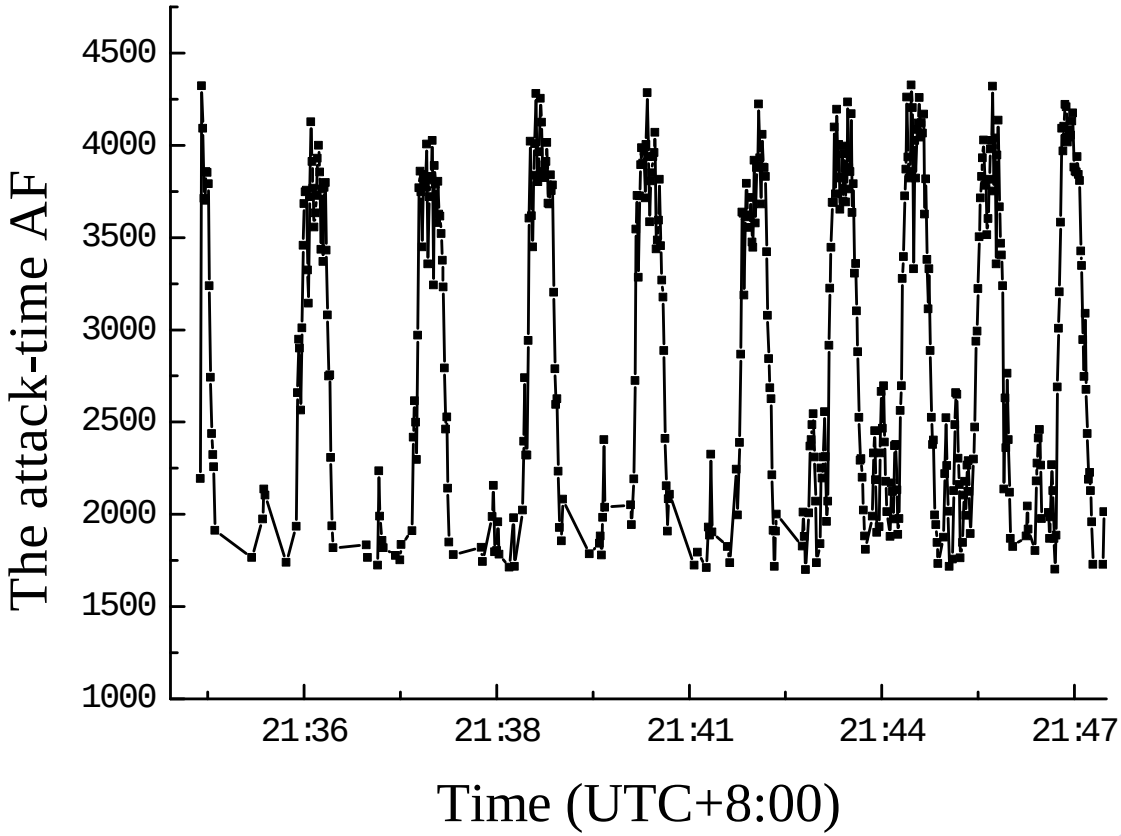
Table of Contents

Experiment Setup

- The attacker located at a Chinese university campus
  - Directly connect to Internet, no BCP38
- The victim located at another American university
  - One public IP and 1Gbps Internet bandwidth
- Exploit 20K Kad nodes
- Store 300 big indices on each Kad node

The attack-time AF

- Average attack-time AF: 2400
- Peak attack-time AF: 4326



The all-time AF

Table : the estimation of all-time AF

DUR (Hour)	0.5	1	2	4	8	12	18	24
All-Time AF	867	1544	2533	3724	4869	5425	5872	6124

- Consider the data's expiration time is 24 hours
- The maximum all-time AF: 6124

Largest possible attack

- Utilize all of Kad's resources
- An attack could reach 670 Gbps
  - One million online Kad nodes at any time
  - The average uplink bandwidth of Kad nodes is 0.67 Mbps
- 280 Mbps upload speed for attacker during attack
- The famous Spamhaus incident was 300 Gbps

Table of Contents

Change Kad?

- Put reasonable limits on the length of filenames and the number of returned indices in Kad
- This simple fix would cap the AF for this particular P2P network
- But it is difficult to deploy these incompatible modifications

Ingress filtering

- BCP 38 stops DRDoS attackes from the start
  - At about 80% deployment
  - Remaining 20% of ISPs are reluctant to deploy BCP ingress filtering
- BCP 38 was written in May of 2000
- It is safe to assume that ingress filtering will not end IP spoofing in the foreseeable future

Traffic filtering

- We believe that traffic filtering is the key to defeating DRDoS attacks
- Victims must be able to create new rules dynamically in response to attacks
- Filtering rules must be pushed as close to the source of the attack as possible
- We give operational examples of such a system based on **BGP flow specification**

Table of Contents

Thank you

- Questions?