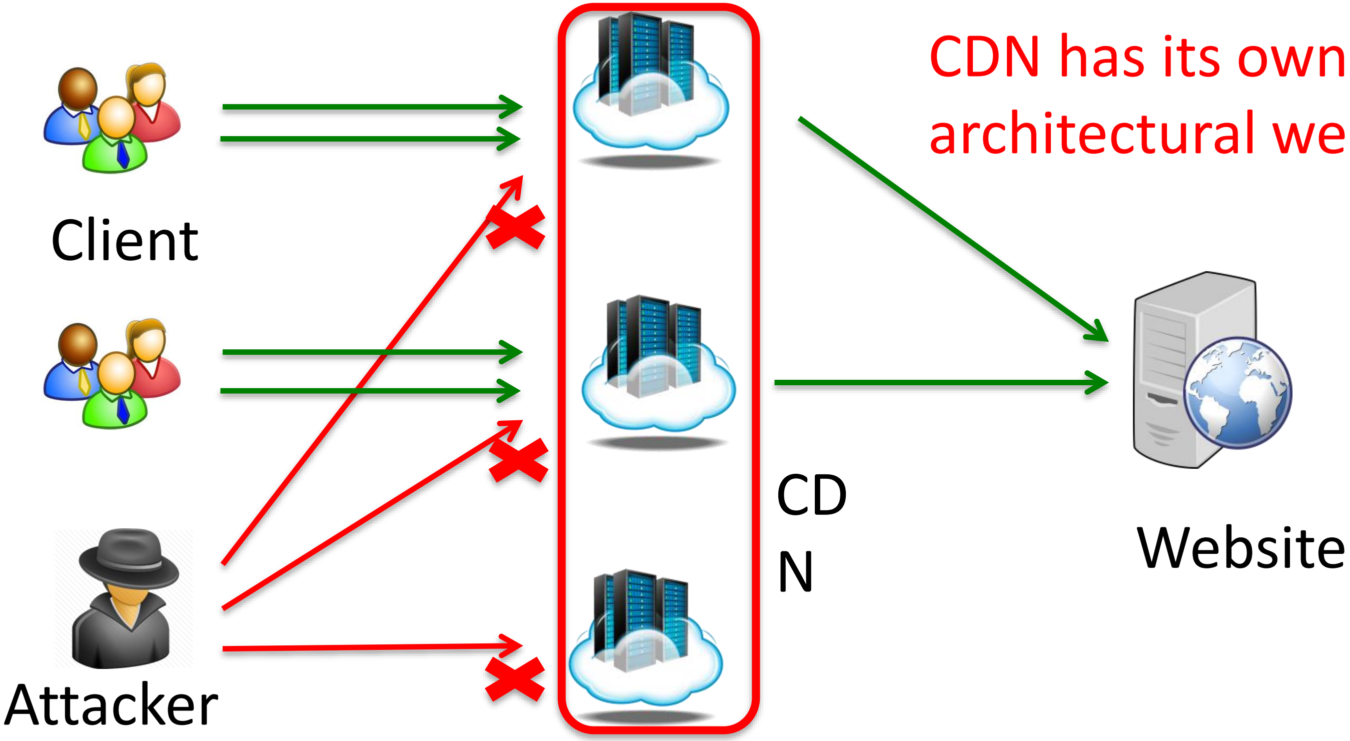
Forwarding-Loop Attacks in Content Delivery Networks

Jianjun Chen, Jian Jiang, Xiaofeng Zheng,
Haixin Duan, Jinjin Liang, Kang Li,
Tao Wan, Vern Paxson

Content Delivery Networks

- CDN is now an important Internet infrastructure, it is a popular solutions for:
  - Performance, Security(WAF), Availability(anti-DDoS)

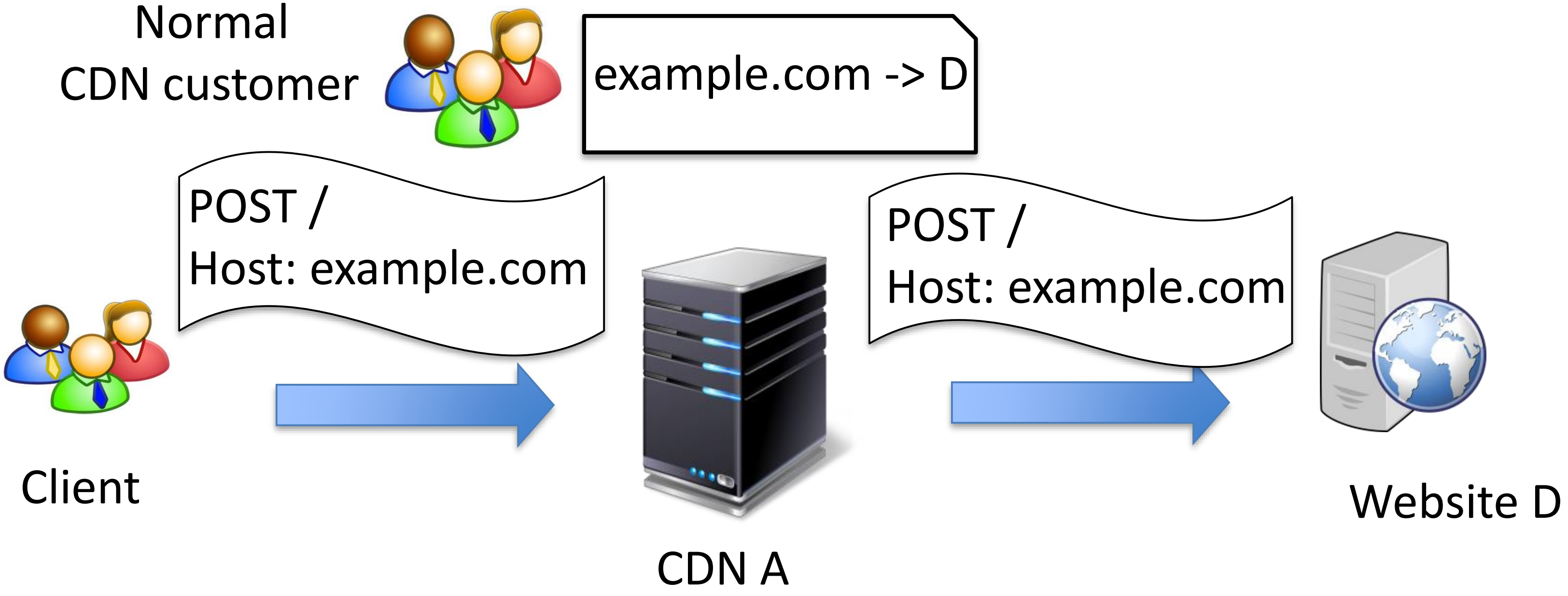


CDN has its own architectural weaknesses

Our work

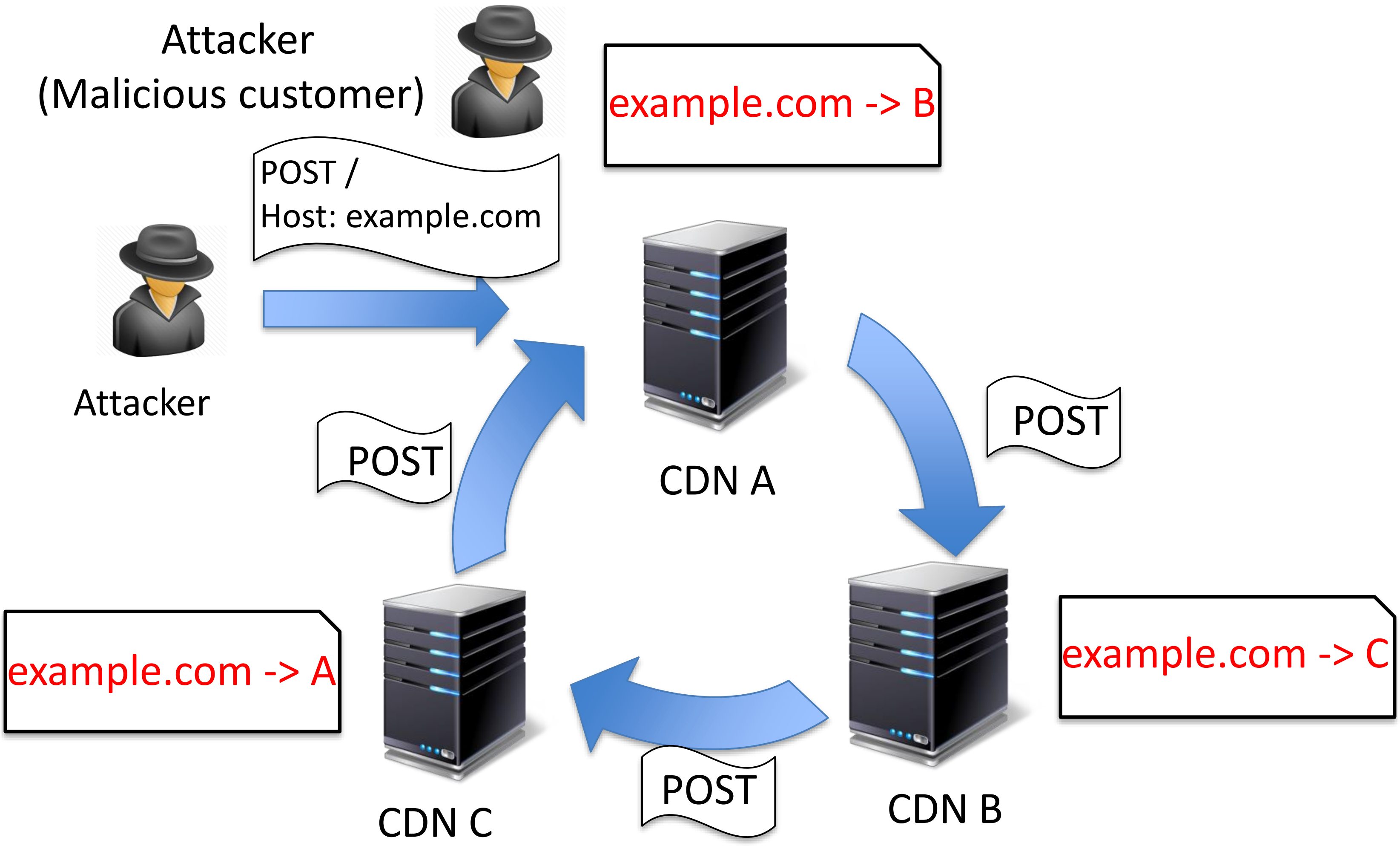
- We present “forwarding loop” attacks that threaten CDN availability.
- We measured 16 popular CDNs and find all of them are vulnerable to such attacks.
- Vendors have acknowledged the problem and are actively addressing it.

The normal forwarding process of CDNs



Customer controls forwarding rules of CDNs

Conceptual view of a forwarding-loop attack



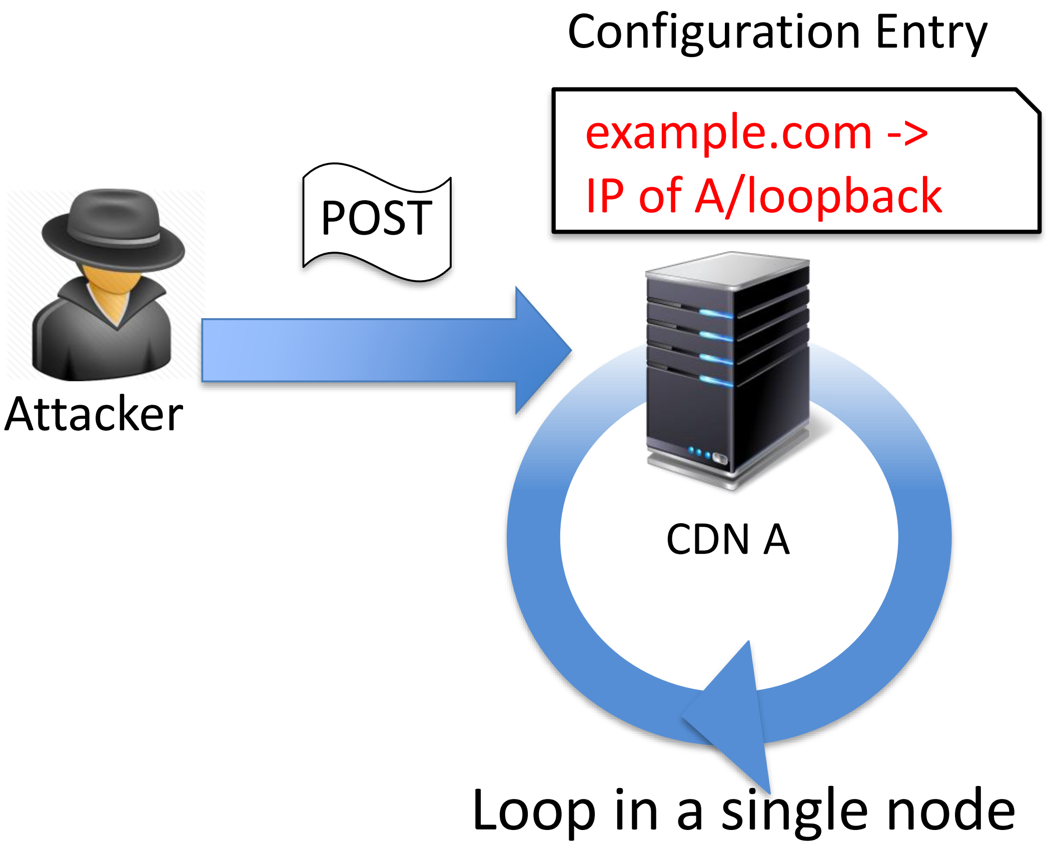
- Malicious customers can manipulate forwarding rules to create loop
- Amplification -> consume resource -> potentially DoS

Practicality of forwarding-loop attacks

- Cost
  - All 16 CDNs provide free or free-trial account
- Anonymity
  - 11/16 CDNs only require an email address
- Some CDNs agreed this attack is severe

- Next we describe 3 types of looping attacks, and 3 factors for enhancing the loop
  - Self loop, intra-CDN, Inter-CDN
  - Abort-forwarding, Streaming, gzip bomb

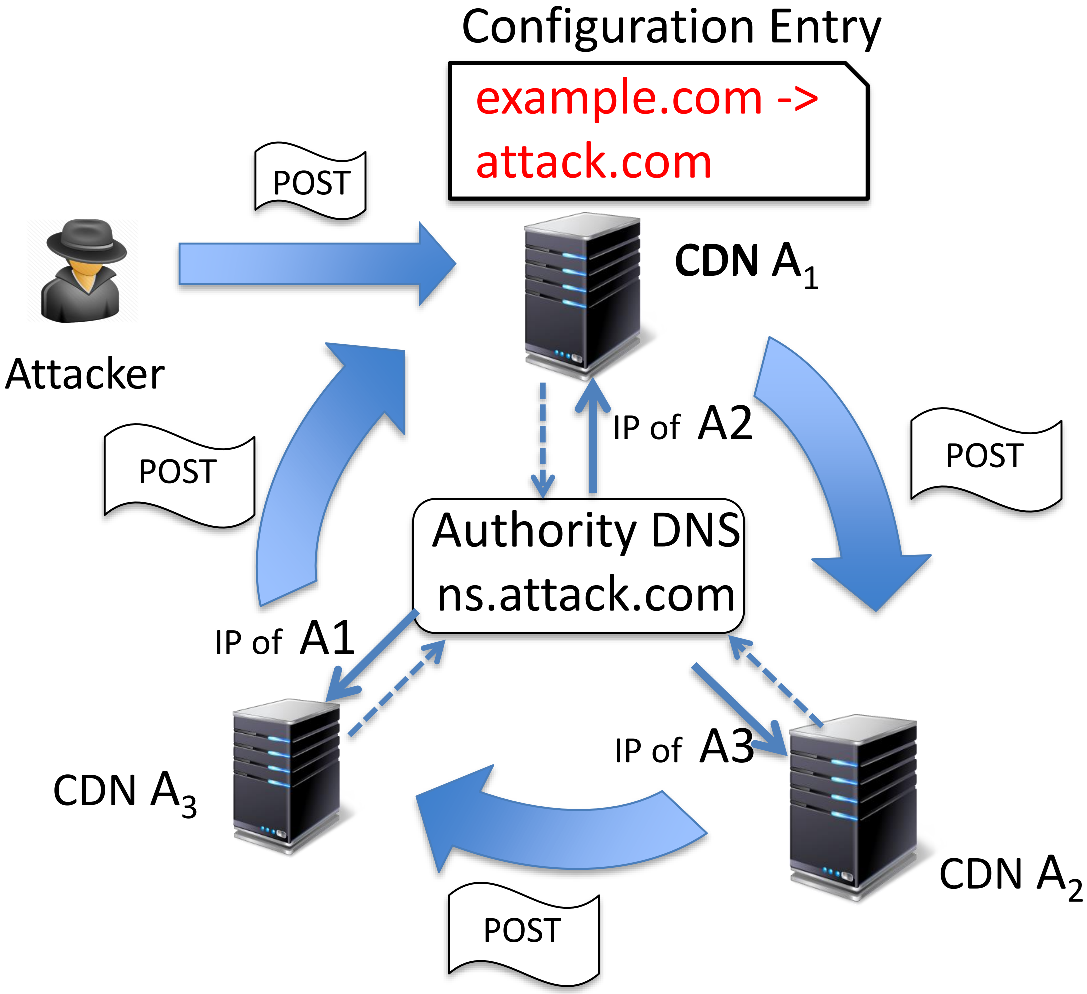
Self loop



Affected vendors(1/16):

- Azure(China)

Intra-CDN loop

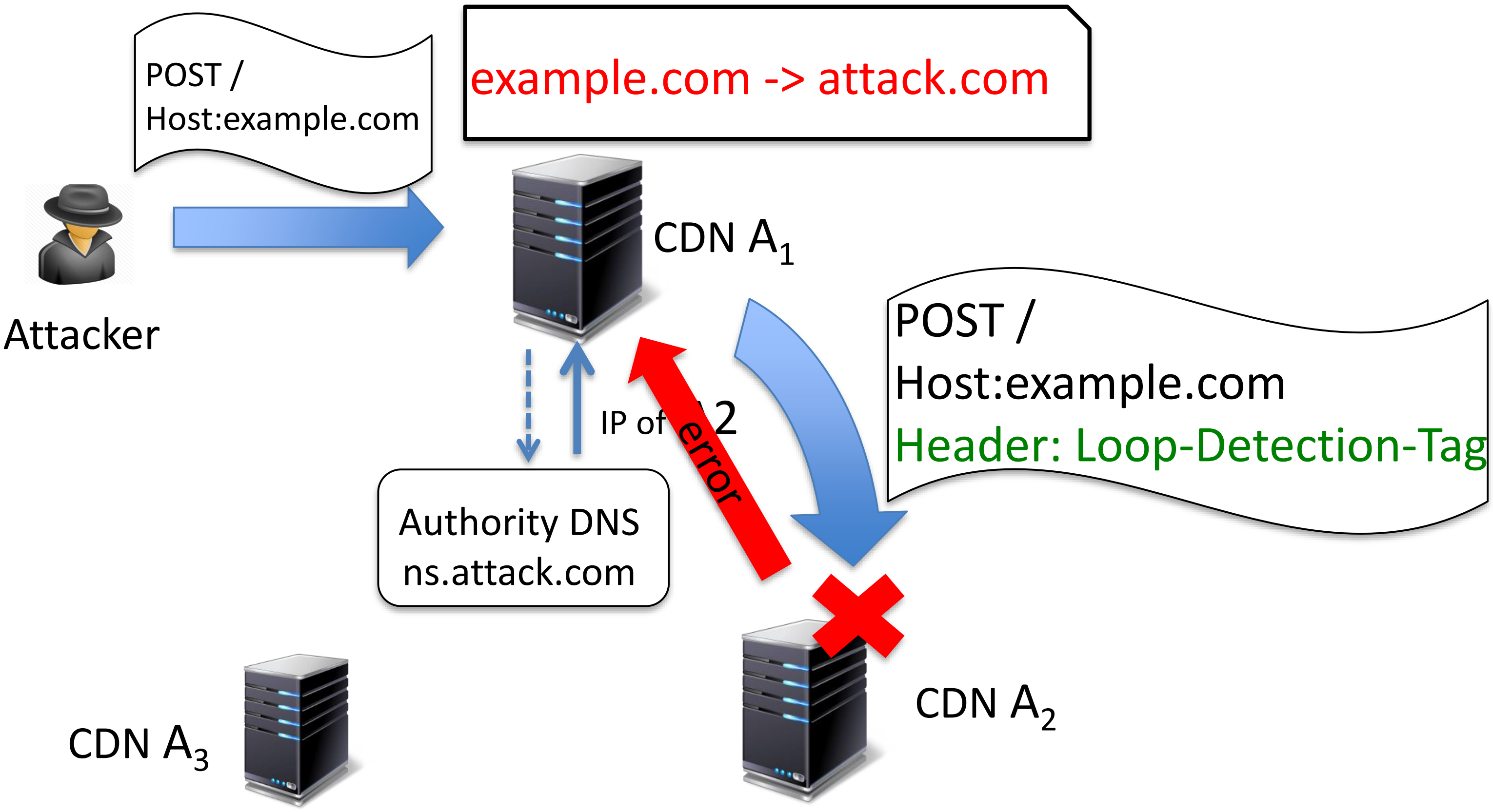


Affected vendors(7/16):

- Azure(China)
- CDN77
- CDNlion
- CDN.net
- CDNsun
- KeyCDN
- MaxCDN

Loop among multiple nodes within one CDN

Loop Detection by CDNs



Current Defenses	Use headers to tag processed requests

Loop-Detection Headers are different

CDN Provider	Loop Detection Header	CDN Provider	Loop Detection Header
Akamai	Akamai-Origin-Hop	**CloudFlare**	X-Forwarded-For CF-Connecting-IP
Alibaba	Via	**CloudFront**	Via
Azure(China)		**Fastly**	Fastly-FF
Baidu	X-Forwarded-For CF-Connecting-IP	**Incapsula**	Incap-Proxy-ID
CDN77		**KeyCDN**	
CDNlion		**Level3**	Via
CDN.net		**MaxCDN**	
CDNsun		**Tencent**	X-Daa-Tunnel

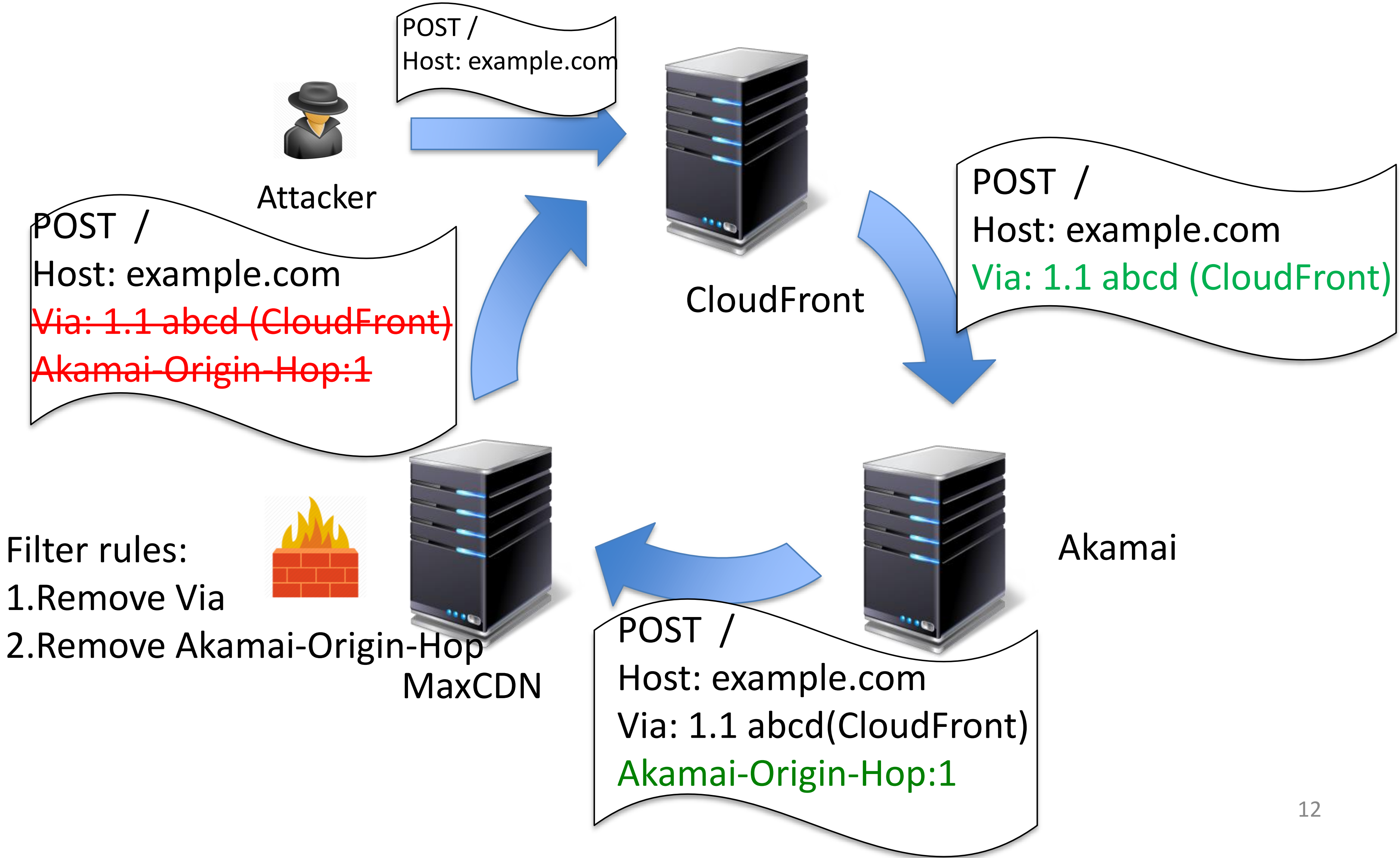
RFC 7230 recommends to use Via header for loop detection

Bypassing CDN defenses

- Chain loop-aware CDNs to other CDNs that can be abused to *disrupt* loop-detection headers
- Abusive features provided by CDNs:

CDN Provider	Reset	Filter
CDN77	Via	
CDNlion	Via	
CDN.net	Via	
CDNsun	Via	
Fastly		No-self-defined
MaxCDN		Any

Inter-CDN loops:



POST /
Host: example.com

POST /
Host: example.com
Via: 1.1 abcd (CloudFront)

POST /
Host: example.com
Via: 1.1 abcd(CloudFront)
Akamai-Origin-Hop:1

POST /
Host: example.com
~~Via: 1.1 abcd (CloudFront)~~
~~Akamai-Origin-Hop:1~~

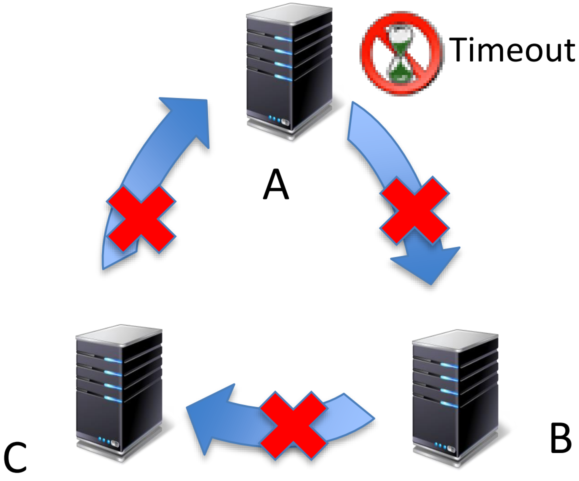
Filter rules:
1.Remove Via
2.Remove Akamai-Origin-Hop

Can a loop last indefinitely ?

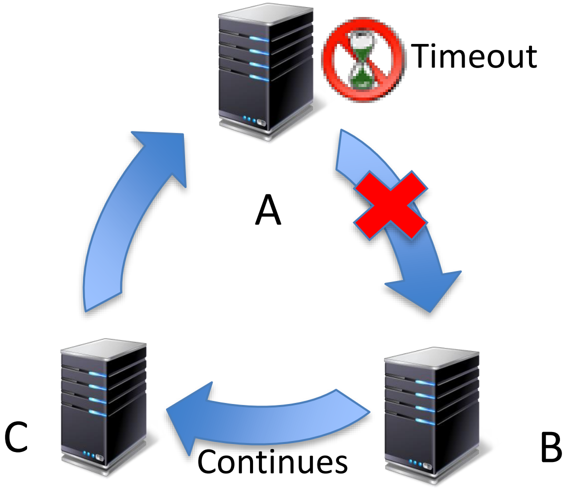
- Limitation on header size might terminates a loop
  - All CDNs limit header size;
  - some CDNs increase header size when forwarding a request;
  - Filtering and reset behaviors can bypass such limitation
- Timeout might also terminate a loop
  - A careful attacking plan can avoid this effect.

Handling timeout

Factors	Attacker countermeasure
Timeout	Add a no-abort-forwarding node(7/16)



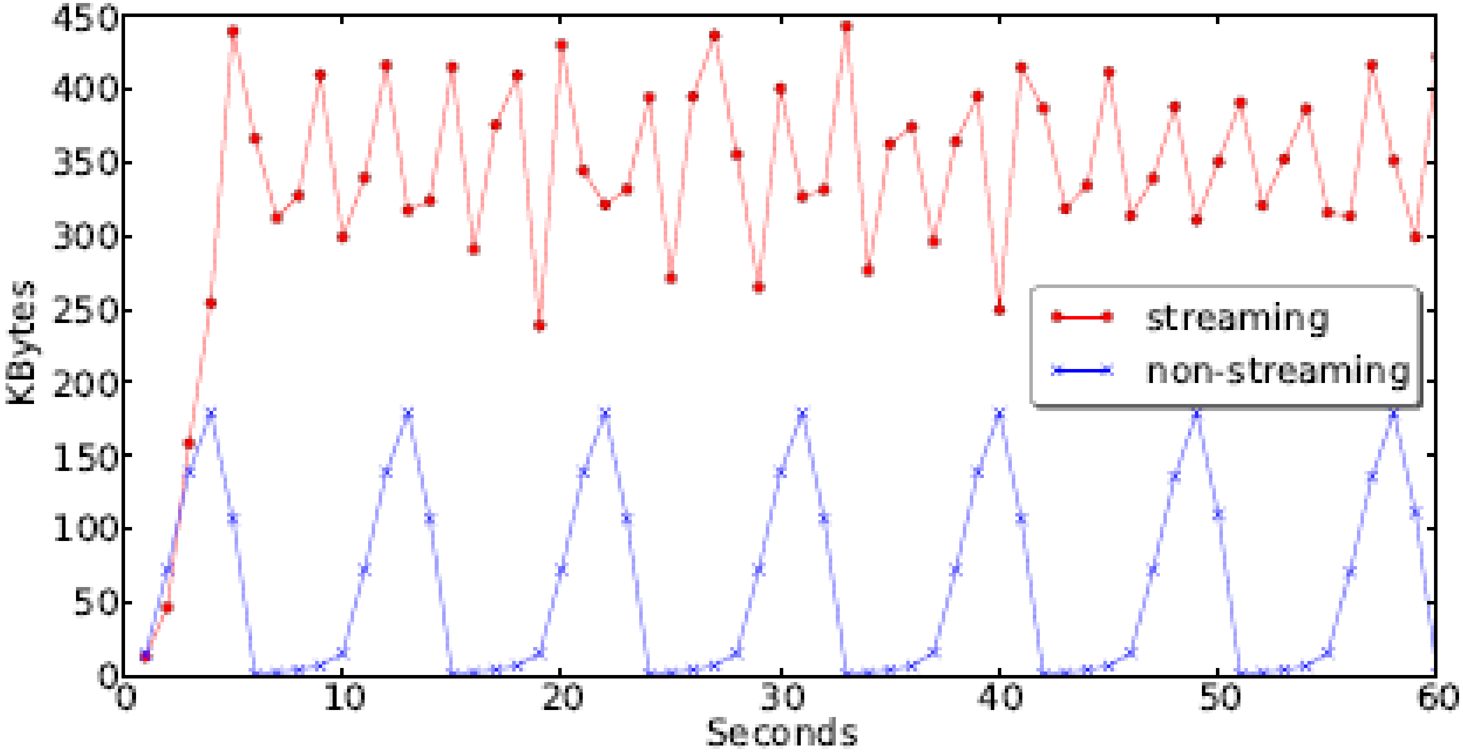
Abort-forwarding



No-abort-forwarding

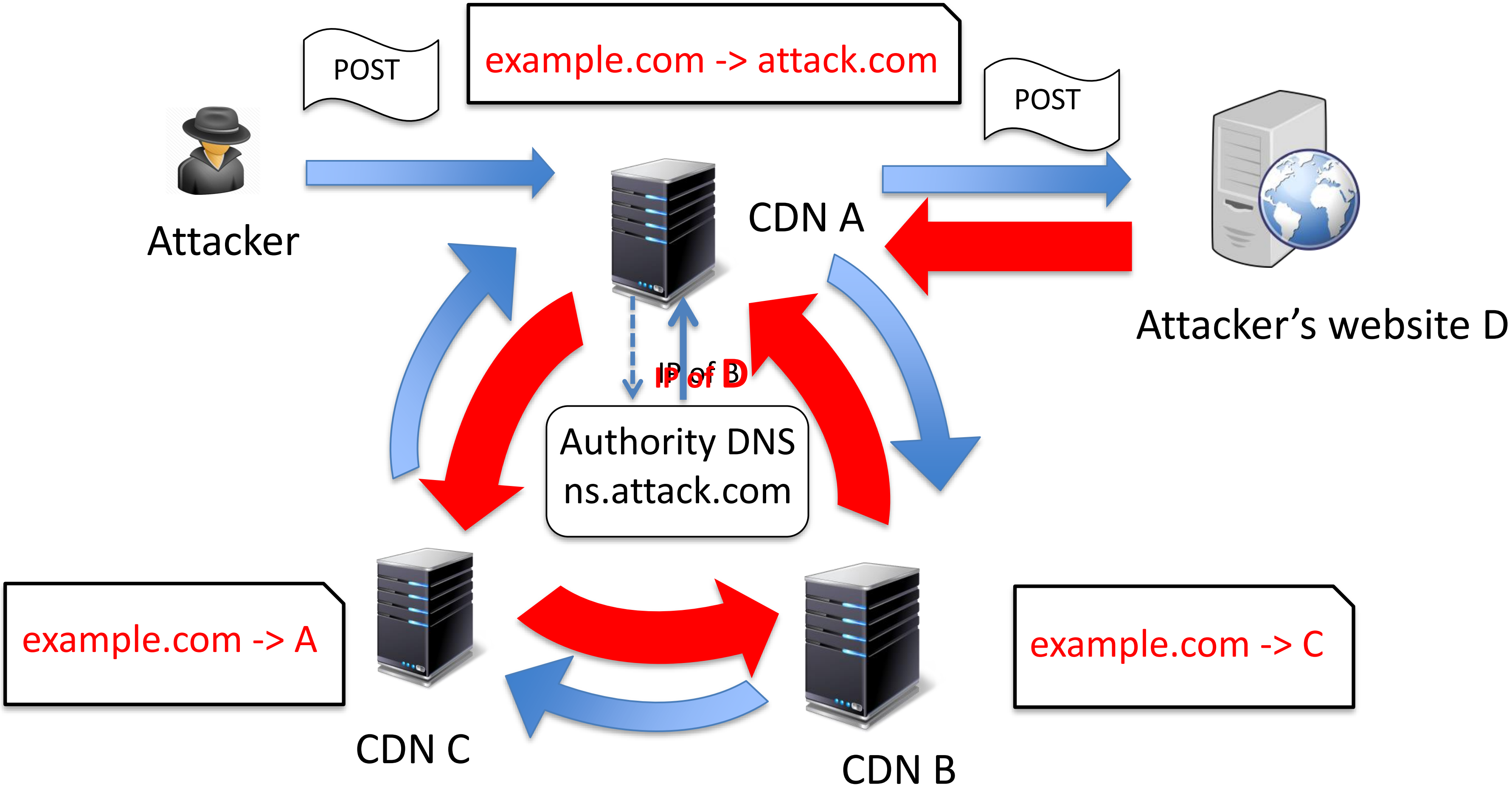
- Experiment
  - A request loops for 5+ hours among CloudFlare, MaxCDN, CDN77 and our control node

How to enlarge attacking traffic?



- Streaming loop
  - faster speed -> overlap -> higher traffic
  - All nodes need to support streaming
  - 7/16 CDNs support request streaming, all CDNs support response streaming

“Dam Flooding” attack: streaming loop with response



POST

example.com -> attack.com

POST

Attacker

CDN A

Attacker’s website D

IP of B

IP of D

Authority DNS
ns.attack.com

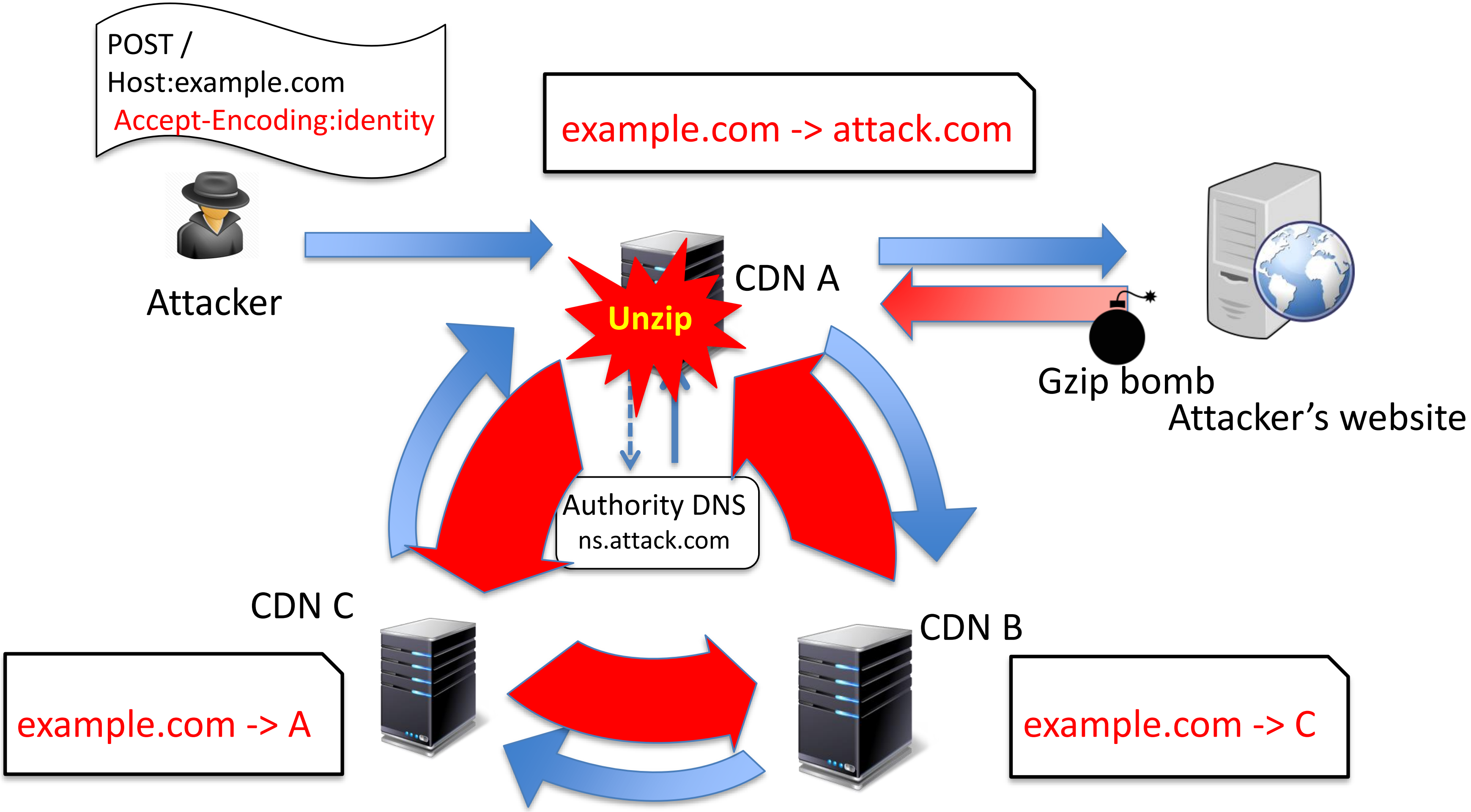
example.com -> A

example.com -> C

CDN C

CDN B

Enhance streaming loop with gzip bomb



```
POST /
Host:example.com
Accept-Encoding:identity
```

- 3 CDNs can be used to uncompress gzip bombs
- Total Amplification Factor = Loop Amplification * Gzip Bomb Amplification(~ 1000)

Defenses

- Unifying and standardizing a loop-detection header,
  - **`Via`** as recommended by RFC
- Interim defenses, independently
  - Obfuscating self-defined loop-detection headers
  - Monitoring and rate-limiting
  - Constraint on forwarding destination

CDN Vendor Feedback

- CDNs are actively addressing it
  - CloudFlare and Baidu implemented **`Via`** header
  - CDN77 and CDNsun will change to not reset **`Via`**
  - Verizon (Edgecast) agreed the problem is serious
  - Tencent evaluates as high risk
  - Fastly actively discussed defenses with us
  - Alibaba are intreseted in interim defenses

Summary

- A variety of implementation issues make forwarding loops a potentially severe attack vector
- A case that highlights the danger of allowing cross-organization, user-controlled (untrusted) policies without centralized administration
- How to enforce standard compliance, especially when global coordination is needed

Acknowledgement

Bai du 百度
CLOUDFLARE
fastly
edgecast
Tencent
Alibaba.com
CDN77.COM
CDN sun
anquanbao.com
Akamai
Azure (China)
amazon cloudfront

Thank you!