Yet Another Traffic Black Hole: Amplifying CDN Fetching Traffic with RangeFragAmp Attacks

Chi Xu[1,2], Juanru Li[1(✉)], and Junrong Liu[2]

[1] Shanghai Jiao Tong University, Shanghai 200240, China
{sjtu_xuchi,jarod}@sjtu.edu.cn
[2] ZhiXun Crypto Testing and Evaluation Technology Co. Ltd., Shanghai 201601, China

Abstract. Content Delivery Network (CDN) has been widely used nowadays as an important network infrastructure to provide fast and robust distribution of content over the Internet. However, an inherent weakness of CDN involved network service is its content fetching amplification issue, that is, the network traffic among the origin server and CDN surrogate nodes is maliciously amplified due to some crafted requests. Such requests can be multiplied by the forwarding of the CDN, posing a serious performance threat to the origin server. Particularly, when the HTTP range request mechanism, which allows the server to respond only a portion of the HTTP message to the request of client, is used, the risk of content fetching amplification is significantly increased. Therefore, defenses against such kinds of traffic amplification have been deployed to protect CDN users from being over charged.

In this paper, we revisited HTTP range request cased content fetching amplification issue and evaluated the deployed defenses of mainstream CDN providers. Specifically, we proposed Range Fragment Amplification (RangeFragAmp) attacks, a new variation of CDN content fetching attack related to HTTP range request mechanism. The proposed RangeFragAmp attacks have concealment and bandwidth consumption capability. Our pentests against five CDN providers with more than 2.5 million users demonstrated that all of their CDNs were vulnerable to RangeFragAmp attacks. Particularly, S-RFA attack, one of the two types of RangeFragAmp attacks, can achieve an amplification factor of 11345 on *Baidu AI Cloud*. We have reported the issues to the involved CDN providers, and expected our study could help CDN designers and developers build more robust systems.

Keywords: CDN security · HTTP range request · Amplification attack · DDoS

This work was partially sponsored by the Shanghai Science and Technology Innovation Fund (Grant No. 19511103900), the National Key Research and Development Program of China (Grant No. 2020AAA0107800), and the National Natural Science Foundation of China (Grant No. 62002222).

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 439–459, 2021.
https://doi.org/10.1007/978-3-030-92635-9_26

1 Introduction

As an important infrastructure of the Internet, CDN security has received widespread attention. Some resource consumption attacks on the CDN have been found to threaten the security of the CDN itself and servers hosted on it. For example, *CDN loop* attacks constructing loops in CDN aim to reduce the availability of CDN [1,2]. *No-abort* attack, which depletes the bandwidth of the origin server by rapidly dropping the CDN-client connections [3,4]. The amplification attack is an attack in which an attacker can use an amplification factor to multiply his power [5]. The most typical amplification attacks on CDN include: DNS Amplification (*DNS-A*) attack [6], UDP Reflection Amplification (*UR-A*) attack [7], and Range-based Amplification (*RangeAmp*) attacks.

The range request in the HTTP protocol is a mechanism that should be used to improve transmission performance. However, the flaws in the protocol design and the negligence of the CDN in the implementation process allow this mechanism to be used for amplification attacks. For instance, *RangeAmp* attacks [4] disclose the vulnerabilities of HTTP range requests on CDNs.

Although mainstream CDN providers claimed that they have deployed defenses against *RangeAmp* attacks, we found HTTP range requests are complex and it is difficult to implement a comprehensive defense against all variations of malicious range request. Therefore, we revisited those defenses deployed on popular CDNs and tested their effectiveness by proposing a new class of amplification attack against CDN surrogate nodes and origin servers. Our proposed Range Fragment Amplification (RangeFragAmp) attacks leverage the weakness of current fragment based transmission to implement a CDN content fetching amplification, which not only affects the performance of the origin server but also the CDN system. In particular, RangeFragAmp attacks include two kinds of attacks: Small Range Fragment Amplification (S-RFA) attack and Overlapping Range Fragment Amplification (O-RFA) attack. In both attacks, an attacker constructs an HTTP request within a minimum transmission fragment range to the CDN, and drive it to fetch a large range of data from the origin server. Since the size of fragment would not change according to the request range, the attacker only needs to first determine the range size of the fragment, and then requests a specific range that crosses two fragments to produce a series of considerable amplification attacks.

Unlike existing CDN traffic amplification attacks such as *RangeAmp* attacks, our proposed RangeFragAmp attacks rely on multiple range requests to implement traffic amplification. Since the crafted requests comply with RFC 7233 [8], deployed defenses could not effectively detect and block such malicious requests. To the best of our knowledge, we are the first to discuss the security risks posed by malicious range requests against fragment based transmission in CDN application scenarios.

To measure whether mainstream CDN providers adopted a well-protected range forwarding policy against RangeFragAmp attacks, we built various experimental environments and simulated the impact of different amplification attacks on CDNs. We tested five mainstream CDN providers including *Alibaba Cloud*,

Tencent Cloud, *Huawei Cloud*, *Baidu AI Cloud*, and *CloudFront*. As shown in Table 1, we found most CDN products, even though they are immune to existing attacks such as *DNS-A* and *UR-A* attacks, our proposed RangeFragAmp attacks (especially the (S-RFA) attack) could circumvent the protection. This demonstrates CDN providers did not well understand the root cause of range request based traffic amplification attacks, and hence we proposed a better mitigation scheme to comprehensively protect CDNs against such kinds of threats.

Table 1. Summary of CDNs against existing traffic amplification attacks and our proposed RangeFragAmp attacks

CDN provider	*DNS-A* [6]	*UR-A* [7]	*SBR* [4]	*OBR* [4]	S-RFA	O-RFA
Alibaba Cloud [9]	✗	✗	✗	✗	✓	✗
Tencent Cloud [10]	✗	✗	✗	✗	✓	✓
Baidu AI Cloud [11]	✗	✗	✗	✗	✓	✗
Huawei Cloud [12]	✗	✗	✗	✗	✓	✗
CloudFront [13]	✗	✗	✗	✓	✓	✓

Ethical Consideration. We specially registered experimental accounts for all of our evaluation. In our experiments, we simulated the attacks against our own cloud servers so as not to influence real world network services. Furthermore, when evaluating the feasibility and amplification factor of RangeFragAmp attacks, we used cloud servers with small bandwidth (i.e., 1 Mbps). Therefore, the workloads of tested CDN servers would not increased too much, and our tests would not cause any usability impact on the CDN servers or other cloud servers hosted on them.

Furthermore, we have contacted all influenced CDN providers and reported the vulnerabilities we found to them, and we have received feedback from four of them. For instance, some of the CDN providers planned to adopt a more flexible dynamic fragment mechanism to avoid such amplification.

2 Background

In this section, we first briefly introduce essential features of CDN, then particularly discuss the HTTP range request mechanism and some relevant amplification attacks in CDN applications.

2.1 CDN Overview

CDN is an important Internet infrastructure composed of edge node server clusters distributed in different geographical areas [14]. It not only improves the performance for the websites of its customers but also provides security features such as DDoS (Distributed Denial-of-Service) protection mechanisms [15].

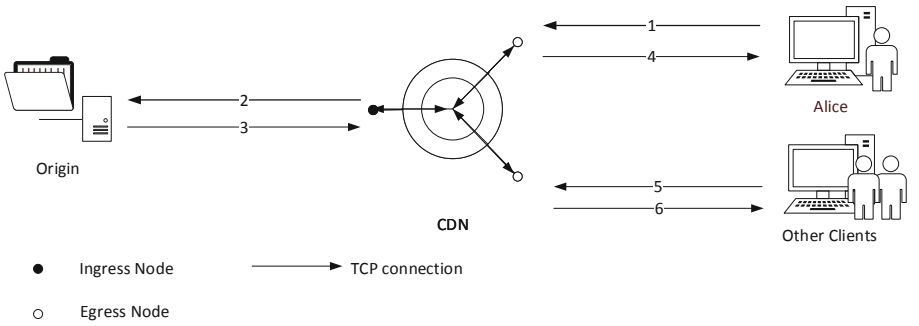

Fig. 1. Multiple segments of connectivity in a CDN environment.

As shown in Fig. 1, the CDN network can be divided into two parts: the central nodes and the edge nodes [16]. The central nodes are usually responsible for global load balancing and content management of the entire system, while the edge nodes play a crucial role in content distribution and caching usually. Edge nodes can be divided into ingress nodes and egress nodes according to their locations and functions. In general, the egress nodes are closer to the client, so they are responsible for the access of client and content distribution. In Fact, a normal client can only establish connections with the egress nodes most of the time. Similarly, the ingress nodes are closer to the origin server, so CDN usually forwards requests through the ingress nodes to obtain the latest contents from the origin. In general, this process is also called *Range Origin Fetch*.

Actually, there are multiple connection paths in the CDN environment between the client and the origin, such as the connections *client-cdn* between the client and the egress nodes, the connections *cdn-origin* between the ingress nodes and the origin, and connections *cdn-cdn* among egress nodes and ingress nodes in CDNs [17]. Besides, a kind of cascade relationship can be established within a CDN or between CDNs [18].

For convenience, we indicated an external CDN near the client as ExCDN, and an internal CDN close to origin as InCDN. Therefore, there are at least 3 TCP connections among client, ExCDN, InCDN and origin as shown in Fig. 1.

Usually, CDN will try to find response content from the caches of edge nodes preferentially when receiving requests from clients [19]. If the cache misses or expires, the CDN will try to forward requests for the latest content data from the origin server through the ingress nodes, and restore it in the caches of the edge nodes. When the client or other users in the neighboring area request the same content again, the request can be fulfilled immediately, through the response from the caches of edge nodes.

For potential consumers who are willing to try using CDN, performance improvement and cost become the things they care about most. Since the network usually expresses fluctuates due to its own features, the extra cost brought by CDN has become an important reference for consumers when choosing a CDN [20].

2.2 HTTP Range Request Mechanism on CDN

With the advent of the mobile Internet era, more and more large-scale media files have been moved to the Internet. Luckily, the range request in the HTTP protocol allows the server to respond only a portion of the HTTP message to the request of client [8]. In fact, HTTP range requests are particularly helpful when sending large media files or used for resume broken transfer and downloads [21].

As an accelerator for content delivery, a reasonable CDN should have the ability of segmented caching for large media files or documents. For example, the client can specify the transmission range of an image to obtain the contents of a specific fragmentation, instead of receiving the whole image. As for the edge nodes, they should be able to cache and respond to the requests of other clients autonomously in a while.

These requirements have been solved when range request from HTTP was introduced into the CDN. In fact, clients often fail to receive a complete file because of a canceled or interrupted HTTP connection [22]. In terms of efficiency, the client expects that it can continue to retrieve the rest of the data in subsequent requests after acquiring a part of the data, rather than retrieving all the data at once. Meanwhile, capturing part of the data on one request is also good for devices that are running out of storage space.

2.3 Differences in CDNs Handling Range Requests

Different CDN providers have different policies for handling range requests. However, there are still no clear definitions or considerations in related protocols or RFCs to help developers hand HTTP range requests in CDN. Different CDN providers choose to implement range forward policies based on different perspectives, including business, operational and technical views.

At present, there are four basic range requests forward policies deployed on CDN, including:

* *Laziness* - Forward the *Range* header without change.
* *Deletion* - Remove the *Range* header directly.
* *Fragment* - Send the *Range* header in fragments.
* *Continuance* - Request all remaining content from specified *Range* header.

Previously, someone also proposed the *Expansion* policy [4], but after research and evaluation, we believe that it has been replaced with *Fragment* and *Continuance* to meet the new security requirements. In practice, the CDN providers use one or a combination of several basic policies. When *Range Origin Fetch* is off by default, most CDNs prefer to adopt the *Deletion* policy [4].

When the *Range Origin Fetch* is enabled, most CDN providers use *Fragment* policy partially or completely. Moreover, The CDNs will divide the range into several fragments according to a preset size when they receive some requests

containing a wide range, and then forward the request to the origin server in turn. This heuristic method of fragmentation provides a new opportunity for malicious attackers, who just need to make sure the fragment size of each CDN provider in advance - which is not difficult to do - to implement *SBR* attacks within the scope of each fragment. Although some possible mitigation on the CDN side has been proposed in the paper by Li et al., they suggested adopting *Laziness* policy to complete defend against a *SBR* attack or applying *Expansion* policy but not extend the byte range too much e.g. 8 KB [4]. Unfortunately, we are disappointed to find that most CDNs just ignore them.

2.4 Amplification Attacks

As one of the most popular and effective DDoS attacks, an amplification attack is any attack where an attacker is capable of using an amplification factor to multiply its power [5]. The amplification factor can be protocol vulnerabilities (e.g., *UR-A* attack), security negligence of the specifications (e.g., *DNS-A* attack), or both (e.g., *RangeAmp attacks*). Amplification attacks are "asymmetric", which meanings a relatively small number or low level of resources is required by an attacker to cause a significantly greater number or higher level of target resources to malfunction or fail [5,23]. Examples of amplification attacks on CDN include *DNS-A* attack [6], *UR-A* attack [7], and *RangeAmp*[1] attacks [4].

DNS-A attack is a typical DDoS attack based on reflection. Attacker leverages the functionality of open DNS resolvers in order to overwhelm a target server or network with an amplified amount of traffic, rendering the server and its surrounding infrastructure inaccessible [6]. A *DNS-A* attack based on DNSSEC can leads to an amplification factor of 44 [24].

An *UR-A* attack will send a series of special requests based on the UDP service to the victims, and the identity of the attacker will be hidden by forging IP address. This action will generate much larger response traffic to the victim than the requested data. An *UR-A* attack can reach an amplification factor of 556 [7].

RangeAmp attacks are kind of high efficiency amplification attacks, which allow attackers to exploit the range implement vulnerabilities and damage DDoS protection mechanism of CDNs [4]. *RangeAmp* attacks include two types: Small Byte Range (*SBR*) attack and Overlapping Byte Ranges (*OBR*) attack. A *SBR* attack can lead to an amplification factor of 43093, and an *OBR* attack have ability to reach an amplification factor of 7432, which both pose severe threats to the serviceability of CDNs and availability of websites.

3 RangeFragAmp Attack

In this section, we propose a novel RangeFragAmp attacks. This kind of attacks exploit the vulnerabilities to the upgraded range forwarding mechanism of the

[1] *RangeAmp* attack include two types: Small Byes Range (*SBR*) attack and Overlapping Byte Ranges (*OBR*) attack.

CDN, which can break through the existing CDN defense mechanism and bring significant disruptive impacts on the origin server hosted on CDNs and surrogate nodes of CDN itself.

RangeFragAmp attacks include two types: Small Range Fragment Amplification (S-RFA) attack and Overlapping Range Fragment Amplification (O-RFA) attack. The main difference between S-RFA attack and O-RFA attack is on the crafted range of attack requests. A S-RFA attack amplifies on a single Fragment interval, which means it only needs tiny traffic to launch attacks and implement a stable amplification factor. Therefore, a S-RFA attack is more suitable for the attacker with small network bandwidth. However, an O-RFA attack amplifies the traffic across multiple fragment intervals, they need considerable traffic to launch attacks. Meanwhile, the amplification factor of an O-RFA attack depends on the size of the fragment and target resource both. Although O-RFA attack has a better performance when facing a large target resource, it still needs greater bandwidth for attackers.

3.1 Threat Model

In fact, the *Deletion* and *Expansion* policies are beneficial for CDNs to improve service performance [4]. However, improper *Range* request forwarding mechanism will take huge potential security risks to CDNs and origin servers they host, when attackers launch *RangeAmp* attacks. Therefore, CDN providers have upgraded range request forwarding mechanisms – introducing the *Fragment* policy and the *Continuance* policy – to help protect CDN and origin from *RangeAmp* attacks. But we notice that, these fresh policies still require CDN to retrieve many more bytes from origin server than ones requested by client – in a fragmented range.

On the other hand, when the cache of edge nodes is missed, CDN will directly forward fragment range request to origin server. If CDN has no mechanisms to check whether the *Range* header contains several similar parts already stored in the cache, the response sent by CDN can be larger than the one by the origin server. These cases will cause significant traffic differences between different fragment sizes and connections in the path from client to origin server.

The striking traffic differences cause by *Fragment* policy will bring a kind of variant range-based amplification attacks, denoted RangeFragAmp attacks. We also discern two scenarios of a RangeFragAmp attacks and demonstrate them in Sect. 3.2 and Sect. 3.3.

In RangeFragAmp attacks, an attacker can send many maliciously craft but legal range requests to the CDN, as shown in Fig. 2. Different from the previous *RangeAmp* attacks, now the attacker cannot send small range requests naively. Instead, he or she must be prepared before "work". Because the fragment sizes that are usually not the same in different CDN providers, have tremendous affection for amplification effects. "Fortunately", the size of fragment can be easily found in the official documentation (such as *Alibaba Cloud*), or captured on the server of attacker which has been accessed to the same CDN with victim in advance.

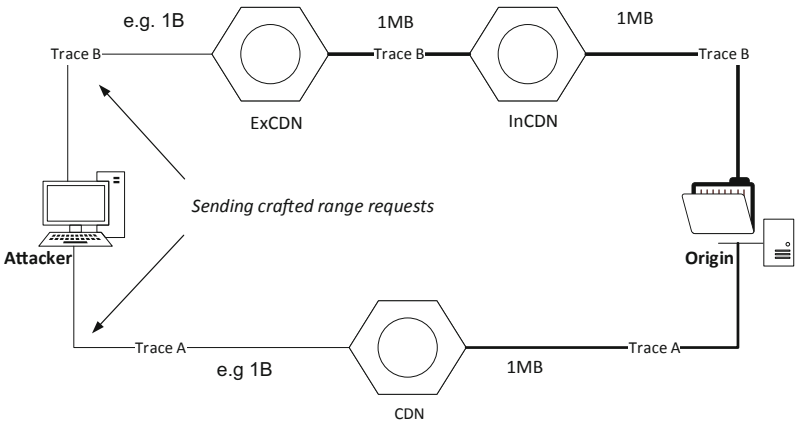

Fig. 2. General construction of RangeFragAmp attacks.

Unlike *RangeAmp* attacks, origin servers are always victims when the attacker launches RangeFragAmp attacks no matter in which scenario of *Trace A* or *Trace B* as shown in Fig. 2. Moreover, the ExCDN and the InCDN in *Trace B* can be other victims when they have been cascaded together by owner or attacker.

3.2 S-RFA Attack

If a CDN adopts the *Fragment* policy to handle range requests, an attacker can craft *Range* header with a few bytes in each fragment range to launch RangeFragAmp attacks. After upgrading the range request forwarding mechanisms, we confirm that all of the CDNs have dropped purely *Deletion* or *Expansion* policy in order to protect from *SBR* attacks. When users enable the *Range Origin Fetch*, they tend to send more fixed-size fragments when they receive range requests. Therefore, an attacker can craft a series of requests that *Range* header with a small byte range in every interval of fragment size to launch RangeFragAmp attacks. We call it S-RFA attack. In a S-RFA attack, the *cdn-origin* connection will transport a much larger traffic than *client-cdn* connection influenced by fragment size. Therefore, an attacker has the ability to consume a large number of bandwidth resources to the origin server through the CDN without determining its real IP address.

As shown in Fig. 3, once the attacker has determined the size of the target file and the provider of CDN that victims used, he or she can launch RangeFragAmp attacks by writing a series of range requests. For example, if a victim has a video of 5 MB on the server and the fragment size of CDN he used is 512 KB (524288 bytes), an attacker is able to craft 10 different requests such as "Range: bytes = 1–1", "Range: bytes = 524289–524289", "Range: bytes = 1048577–1048577" to implement amplification attack. As a result, origin server will return an entire

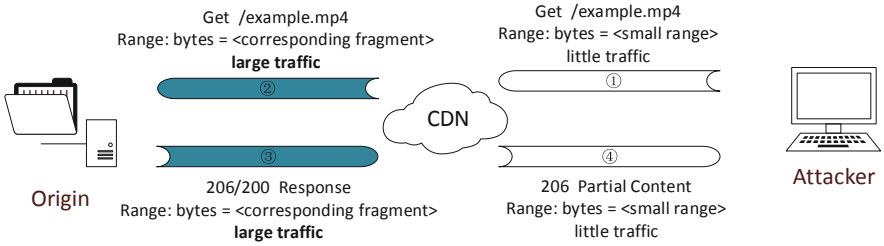

Fig. 3. Flow and example construction of a S-RFA attack.

fragment of target resource like "Range: bytes = 0–524287", "Range: bytes = 524288–1048575", but edge nodes will return only a partial content specified by *Range* header, which can be as small as 1 byte if not considering transmission overhead.

In a S-RFA attack, the size of response traffic in the client-cdn connection is just hundreds of bytes (which is small). When the CDN adopts *Fragment* policy, response traffic between the CDN and origin server completely depends on the size of the fragment adopted by the CDN provider and the size of target resources selected by the attacker. In fact, the bigger the fragment, the larger the amplification factor. On the other hand, the bigger the target resource, the more attack fragments can be crafted, which means the more bandwidth consumption could be done by the attacker.

3.3 O-RFA Attack

If the ExCDN adopts the *Fragment* policy and the InCDN returns corresponding full fragment response without checking whether range contains several similar parts already stored in the cache, an attacker can craft a series of *Range* header that contains a large number of similar but not identical overlapping fragments to launch fresh RangeFragAmp attacks. We name this O-RFA attack. Different from an *OBR* attack, we focus on bandwidth consumption in fragment conditions adopted by CDNs now. In an O-RFA attack, the *ExCDN-InCDN* and *InCDN-origin* connections both transport a much larger traffic than *client-ExCDN* connection, which makes the attacker have significant ability to consume the bandwidth available from ExCDN to origin.

As illustrated in Fig. 4, an attacker launches a series of elaborate range requests that spanning n fragment intervals. For example, when n equals 1, *Range* header could be like "Range: bytes = 524287–524288", "Range: bytes = 10485765–1048576", then sends them to ExCDN. Since CDNs have adopted *Fragment* policy to protect from *OBR* attack now, the fragment range actually forwarded by ExCDN and InCDN covers n complete fragments scope that overrides the minimum range requested by the attacker. In this case shown in Fig. 4, ExCDN and InCDN transmit *Range* header like "Range: bytes = 0–10485765" and "Range: bytes = 524288–1572863" respond to the requests of attackers. As

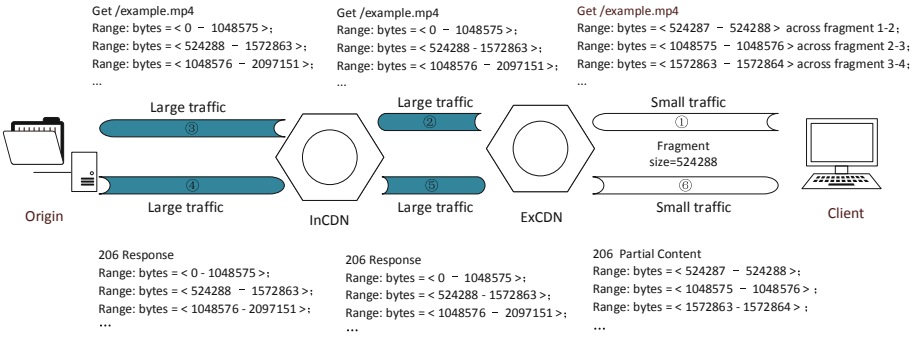

Fig. 4. Flow and example construction of O-RFA attack.

we can see, for an O-RFA attack, the attacker only needs 2-byte requests to consume 1 MB traffic of both *ExCDN-InCDN* and *InCDN-origin* connections in a time.

In an O-RFA attack, when the target resource is fixed, the traffic consumed by the *ExCDN-InCDN* and *InCDN-origin* connections is directly proportional to the fragment size of the deployed CDN and the number of overlapping fragments of the range request crafted by the attacker. Intuitively, the larger the fragment scope covered by the range request, the more traffic will be consumed for *ExCDN-InCDN* and *InCDN-origin* connections.

However, there is a trade-off relation between the amplification factor and the overhead of attacker in an O-RFA attack. The amplification factor of an O-RFA attack will decrease as the number of overlapping fragments in the range request increases until the traffic consumption of attacker is nearly equal to transmission by the origin server.

4 Real-World Evaluation

To evaluate the feasibility and severity of RangeFragAmp vulnerabilities in the wild, we designed and conducted some experiments. Firstly, we have verified whether *RangeAmp* vulnerabilities have been patched up. Then, we examine the five most representatives CDNs in order to find out which are vulnerable to a RangeFragAmp attack. Lastly, we calculate the actual amplification factors and analyze the practical impacts. In all experiments, our origin server is the same Linux server with 1vCPU 2 GiB, CentOS 7.3 64-bit, and 1 Mbps of bandwidth. And our origin website is powered by Nginx/1.16.1 with default configuration deployed.

4.1 Consideration in Selecting CDN Providers

We tested five of the most mainstream CDN providers, including *Ali Cloud*, *Tencent Cloud*, *Huawei Cloud*, *Baidu AI Cloud* and *CloudFront*. According to

the previous research of Kashaf et al., CDN services are highly centralized and oligarchic [25]. The five providers we selected have supply service for nearly 75% of websites which have accessed CDN in China [26].

At the same time, all the CDN providers we selected open their services to individual users and provide free trial or large-capacity traffic packages. In fact, we only spent less than 200 RMB to complete all the experiments.

For experimental security and accuracy of results, we deploy our server individually behind these CDNs and apply their default configuration in all subsequent experiments.

4.2 S-RFA Attack Evaluation

Table 2. Range forwarding behaviors vulnerable to S-RFA attack (Considering the readability of the table, we list only three range requests for each CDN.)

CDN	Fragment size (Bytes)	Vulnerable range requests	Forwarded range requests
Alibaba Cloud	524,288	1–1	0–524,287
		524,289–524,289	524,288–1,048,575
		1,048,577–1,048,577	1,048,576–1,572,864
Baidu AI Cloud	1,048,576	5,242,879–5,242,879	0–7,340,031
		1,048,579–1,048,579	7,340,032–11,534,335
		15,728,639–15,728,639	11,534,336–15,728,639
Tencent Cloud	4,096	1–1	0–4,095
		4,097–4,097	4,096–8,191
		8,193–8,193	8,192–16,384
Huawei Cloud	524,288	1–1	0–524,287
		524,289–524,289	524,288–1,048,575
		1,048,577–1,048,577	1,048,576–1,572,864
CloudFront	1,048,576	1–1	0–1,048,575
		1,048,577–1,048,577	1,048,576–2,097,151
		2,097,153–2,097,153	2,097,152–3,145,728

Feasibility of S-RFA Attack. In our experiment, we upload three media files of different sizes (3 MB to 50 MB) to our origin server in advance. In order to ensure the accuracy of experimental results, before each experiment, we will use the *Pre-refresh* provided by the CDNs console, which can delete the specified contents in the cache of all edge nodes. Besides, in Table 2 and Table 4 we summarize the existing Range forwarding mechanisms of chosen CDNs through the response header received in the origin server. we try to evaluate its practical advantages and disadvantages. In addition, the monitoring windows watched for traffic on

CDN console is another crucial reference to help us figure out how much economic consumption can be caused by RangeFragAmp attacks.

Our experimental results show that all tested CDNs have upgraded their range forwarding policies. In fact, they have no longer naively adopted *Deletion* or *Expansion* policies when the origin owner chooses to enable *Range Origin Fetch* on CDN, which means a *SBR* attack produces little effect on origin server now. Meanwhile, the *Range Overlap Checking* mechanism makes it difficult for *OBR* attacks to have a performance impact on the origin server too.

As illustrated in Table 2, all of five major CDNs are vulnerable to a S-RFA attack. The second column lists the fragment size when range request have been forwarded by CDNs, and the third column lists attack range we crafted in experiment. Lastly, the forth column presents the policies CDNs handing the corresponding *Range* headers. The details are shown below:

1) ***Alibaba Cloud*** adopts the *Fragment* policy to hand the *Range* header. Specifically, *Alibaba Cloud* has two ways to forward requests after enabling the *Range Origin Fetch*: *Normal Mode* and *Mandatory Mode* [27]. The *Laziness* and *Fragment* policies are adopted in the *Normal Mode*. In other word, the first range request sent by the client will be forwarded to the origin server directly. Furthermore, the subsequent requests will be sent back to the origin with the size of 512 KB. If the *Mandatory Mode* is enabled, all the requests of clients that should be back to the origin server have to remain a fixed size of 512 KB.
2) ***Tencent Cloud*** implements *Fragment* and *Laziness* policies after the *Range Origin Fetch* is enabled. However, We find two different fragment sizes during experiments. When the range data requested by the client is bigger than 4 MB, *Tencent Cloud* CDN will forward each range in the size of 1 MB, otherwise, it will forward the request in size of 4 KB.
3) ***Baidu AI Cloud*** adopts *Fragment* and *Continuance* policies when range request is enabled. In our experiment, we find *Fragment* policy decides range upper boundary of forwarded request, which means the closest fragment covering the request of client will be selected as range start point. As for the lower boundary, CDN adopts the *Continuance* policy that requests all of the remaining content of the specified resource. The connection between CDN and origin will be interrupted as the *client-CDN* disconnects automatically when clients have received all the content they want.
4) ***Huawei Cloud*** implements the *Fragment* policy when *Range Origin Fetch*. That is to say, all the range requests forwarded to the origin server will be transmitted in the size of 512 KB.
5) ***CloudFront*** enables the *Range Origin Fetch* by default. Moreover, it adopts a *Fragment* policy. In fact, the requests of clients will be forwarded in a fixed fragment size of 1 MB back to the origin server.

The Amplification Factor of S-RFA Attack. As shown in Table 3, all the CDNs we selected are vulnerable to a S-RFA attack. Our wild experimental results show that the amplification effect of S-RFA attack is positively correlated

with the fragment size. The bigger the fragment size on CDN is adopted, the greater the amplification factor under the attack of S-RFA will be. At the same time, we also found that although the fragment size is the same, the amplification factor has different values in different CDNs. For example, the amplification factor of *Baidu AI Cloud* is thousands of times larger than *CloudFront*, but they have the same fragment size 512 KB.

In order to explain this magical phenomenon, we carried out in-depth research. We found differences in the handling of forwarding range requests. *CloudFront* strictly forwards client requests according to the fixed size of the fragment. However, *Baidu AI Cloud* will automatically load the head 3 MB resource content for the first time. In addition, *Baidu AI Cloud* will implement a looser fragment strategy, which makes CDN may not only return to the corresponding fragment. For example, in our experiment we found that it would return at least four fragments at a time. This mechanism that should have improved the *Range Origin Fetch* hit rate causes a great amplification factor.

Table 3. The amplification varies with fragment size and target file size in a S-RFA attack

CDN	Fragment size (Bytes)	Amplification factor		
		3 MB	23 MB	50 MB
Alibaba Cloud[a]	524,288	519	510	507
		531	529	525
Baidu AI Cloud[b]	1,048,576	11,345	6,783	5,952
Tencent Cloud	4,096	15	14	14
Huawei Cloud	524,288	702	700	698
CloudFront	1,048,576	1,347	1,342	1,330

[a] *Alibaba Cloud* has two range request forwarding modes: *Mandatory* and *Normal*. The difference between them is mainly about how to handle the first client request, which leads to a slight difference in the amplification factor.

[b] *Baidu AI Cloud* uses the *Fragment* and *Continuance* policies that make S-RFA attack amplification factor have huge differences depending on the size of target recourse.

On the other hand, we also find that the amplification factors of CDNs (except *Baidu AI cloud*) do not fluctuate significantly with the size change of target resources under a S-RFA attack. We believe a S-RFA attack amplification factor is strongly related to the fragment size. The larger the size of transmitted fragments is, the greater the scope that an attacker can influence.

4.3 O-RFA Attack Evaluation

Feasibility of O-RFA Attack. Table 4 shows that *Tencent Cloud* and *CloudFront* are vulnerable to an O-RFA attack when they are deployed as ExCDN

Table 4. Range forwarding behaviors vulnerable to O-RFA attack (Considering the readability of the table, we only list the three range requests when the number of overlapping fragments for each CDN is equal to 1.)

CDN	Fragment size (Bytes)	Vulnerable range requests	Forwarded range requests
Tencent Cloud	4,096	4,095–4,096	0–8,191
		8,191–8,192	4,096–12,287
		12,287–12,288	8,192–16,383
	1,048,576	1,048,575–5,242,880	0–6,291,455
		2,097,151–6,291,455	1,048,576–7,340,031
		3,145,727–7,340,031	2,097,152–8,388,607
CloudFront	1,048,576	1,048,575–1,048,576	0–2,087,151
		2,097,151–2,097,152	1,048,576–3,145,728
		3,145,727–3,145,728	2,087,152–4,194,304

or InCDN. In our experiment we cascade vulnerable CDNs. The second column lists the fragment size which vulnerable CDNs adopted. Then, the third and fourth columns show the range request we crafted and corresponding responses CDNs forwarded.

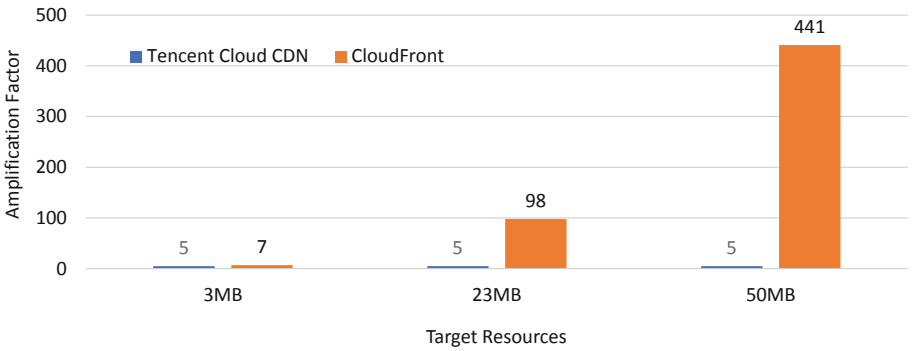

Fig. 5. The amplification varies with fragment size and target file size in an O-RFA attack (The fragment size of *Tencent Cloud* has two different values according to the specific data ranges requested by clients. Here we only present the one that can be successful in an O-RFA attack.)

The Amplification Factor of O-RFA Attack. As shown in Table 5, *Tencent Cloud* and *CloudFront* are vulnerable to O-RFA attack when used as ExCDN or deployed separately like Fig. 2. There are several situations that attackers can perform amplification attacks. 1) When vulnerable CDNs are used as ExCDN, the *ExCDN-InCDN* connection and the *InCDN-Origin* connection will transport much traffic generated by O-RFA attack if the InCDN does not enable the *Range*

Origin Fetch. If InCDN enables the *Range Origin Fetch*, the *InCDN-origin* connection has less impact from O-RFA attack, but *ExCDN-InCDN* connection still suffers a lot of malicious traffic. 2) When a vulnerable CDN is used in a cascading way, an O-RFA attack can produce the same amplification factor between the *ExCDN-InCDN* connection and the *InCDN-origin* connection. That is to say, an O-RFA attack can affect both the CDNs themselves and the original server. 3) In addition, an O-RFA attack has the same impact when the vulnerable CDNs are used alone just shown in Fig. 2 (*Trace B*).

For *Tencent Cloud*, an O-RFA attack works best when the number of overlapping fragments is less than 2, and the increment of amplification factor will begin to decline significantly after the *Threshold*[2] is exceeded. Therefore, for *Tencent Cloud*, the maximum amplification factor of an O-RFA attack is 5, and it does not change with the size of the target resource.

In contrast, when *CloudFront* is attacked by O-RFA, the amplification factor will increase significantly as the target resource becomes larger.

4.4 Severity Assessment

Serious Damage to the Economic Availability of the Website. In the experiment, we find that all the CDNs tested are charged for the actual traffic deliver to client by default, but the victims still have to pay a huge traffic bill after being maliciously attacked. Malicious attackers are able to consume the bandwidth of the origin server and CDN at little cost by means of RangeFragAmp attacks. This condition will become more intractable when mass cloud storage services such as Object Storage Service (OSS) have been wildly introduced into CDNs for acceleration and economy.

A Concealed and Efficient DDoS Attack. Traditional CDN protects the origin server from direct DDoS attacks by masking its IP address [28]. RangeFragAmp attacks does not have to locate the real IP address of the origin server but only needs to send crafted range requests to the CDN to launch amplification attacks that are able to influence the availability of websites. Unfortunately, the vulnerable CDN has no alert under its default configuration when we launch RangeFragAmp attacks.

4.5 CDN Providers Feedback

We reported our findings to CDN providers, and most of them gave us positive feedback. Most CDN providers stated that they had been aware of the potential threat posed by *Small Range Request* since *RangeAmp* attacks were disclosed. In fact, they had already made some corresponding improvements. But it seems that a hasty patch which lacks time for security analysis and experimental verification brings some new troubles. In the future, they will keep evaluating the security

[2] Here the *Threshold* is equal to 2.

risk (including our RangeFragAmp attacks) of *Range Origin Fetch* in CDNs. At the same time, they also promise to upgrade and deploy a new range request forward mechanism once they come up with an appropriate method.

As for the size of the fragment, that indicator has the greatest influence on the amplification factor of RangeFragAmp attacks. All providers indicate that, the current value of the fragment size comes from the experience-based setting during the product design stage. In the future, they will refer to our mitigation solutions and keep improving the stability and security of *Range Origin Fetch* in subsequent upgrade.

5 Mitigation

In this section, we will further discuss the root cause of RangeFragAmp vulnerabilities, limitations of our attack, and mitigation solutions.

5.1 Root Cause Analysis

The naive upgrade of *Fragment* policy on CDN without a comprehensive security analysis is the root cause of the RangeFragAmp vulnerability. The paper proposed by Li et al. has fully discussed the flaws in the HTTP range request on RFC 7233 [8]. Moreover, they found there were no additional illustrations on range request in the newest HTTP/2 protocol which made *RangeAmp* can threat both in HTTP/1.1 and HTTP/2 [8,29]. They hold the opinion that, unclear definition and security negligence of the specification constitute the root cause of *RangeAmp* vulnerabilities and the implementation flaw of greatly worsen it.

Through the analysis of the experimental results, we believe that the root cause of RangeFragAmp attacks and *RangeAmp* attacks are similar. There are few considerations about proxy forwarding scenarios like CDN when designing protocol of HTTP range request. Differently, CDN providers have more influence on RangeFragAmp attacks. For a S-RFA attack, the fragment size will affect the amplification factor directly. We notice that, there is a large variation in the size of the fragment among diverse CDNs. To find out the reasons, we have contacted the CDN provider. In the experiment, we find that *Tencent Cloud* performs best in a S-RFA attack test due to its barely 4096 (4K) fragment size. Their engineers respond that, the fragment size they set is based on the need for 4K alignment of the cache system on the edge nodes, which can speed up the cache reading and integration. At the same time, they also set different transmission fragment sizes for different requested files. In contrast, other providers claim that their fragment sizes are based on the experience of developers or just from designs of peers, without sufficient security reasons. Therefore, we believe that the lack of security analysis in planning and designing phase leads to the differences in the performance of defending S-RFA attack among CDN providers.

As for an O-RFA attack, it is caused by the CDN lack of boundary checking and loose forwarding mechanisms for range requests. In our experiments, we find that two vulnerable CDNs are at risk of being exploited by O-RFA attack, though

they have different performance features. When the *Threshold* is exceeded, the probability that the request is responded to by the cache in the edge node will be greatly increased. That is why the amplification factor attacked by O-RFA in *Tencent Cloud* is not sensitive to the size of target resource.

As for *CloudFront*, it adopts an extremely loose range request mechanism, and it even automatically corrects most of the illegal requests from clients, which causes the amplification factor of an O-RFA attack to increase significantly according to the size of the target file.

5.2 Limitation

Actually, RangeFragAmp attacks have some limitation and it not suitable for all scenarios. On the one hand, a S-RFA attack has a larger amplification factor and wider victim target. On the other hand, the effect of traffic consumption is not as good as an O-RFA attack. Once all the fragments of the target resource have been transmitted or stored in the cache of edge nodes, a S-RFA attack will become futile. Because the edge node will directly respond to subsequent range request of the attacker from the cache. In fact, a S-RFA attack is more often used to launch attacks from small bandwidth condition such as cellular networks. Whereas, an attacker can launch S-RFA attack free of geographic restriction.

Besides, an O-RFA attack has a better traffic consumption effect. Because its amplification factor is related to the size of the target resource. Therefore, an O-RFA attack can not only attack the origin server but also affect the performance of CDN itself. However, an O-RFA attack has stricter requirements on the bandwidth and range fragment combinations of the attacker compared with a S-RFA attack. Therefore, it is more suitable for the attacker who has a large bandwidth, such as the terminal in high-speed Ethernet.

5.3 Solutions

Actually, previous studies have proposed many mitigation schemes. For example, Li et al. suggested fixing this problem from the server side, CDN side and protocol side [4]. After evaluation, we believe that it too difficult to resist this kind of amplification attack without affecting the normal host on server side. Because it is hard to distinguish the message of the attacker from the normal request, especially for distributed CDN nodes. On the other hand, revising a well-defined and security-aware RFC is indeed a solution able to fix this problem thoroughly. However, the new protocol still requires many researchers to analyze and design, and the protocol upgrade process often takes several years.

Therefore, we believe that improving the range request forwarding policy on the CDN side is the most effective and low-cost approach at present. For example, the edge nodes of CDN should switch to different fragment sizes according to the range of client requests and resource capacity, just like *Tencent Cloud* does. In addition, adopting an incremental caching mechanism on the edge node can reduce the pressure on the origin server effectively when CDN forward request to the origin server. Besides, a better CDN should adopt some more stringent range

forwarding policies, including boundary checking and input semantic analysis. Moreover, it is supposed to avoid unnecessary expansion interpretation and reject unreasonable access requests. Last but not the least, we agree with Li et al. on their opinion that CDNs should perform a full security evaluation before supporting new protocol features [4].

6 Related Work

HTTP Range Security. Our work is based on *RangeAmp* attacks [4]. And as far as we know, there is no more academic literature discussing the security risks posed by range requests in CDN the environment before. After *RangeAmp* attacks were published, we revisited the rationale for the attack and tested its latest performance on the most representative CDNs. Our research shows that all the CDNs selected have upgraded the range forwarding policy and fixed the vulnerabilities exploited by *RangeAmp*, but the improved forwarding mechanism has brought new problems. Therefore, we propose a novel range fragment amplification attack based on the existing range request forwarding mechanism to implement the traffic consumption and amplification attack on the CDNs and origin servers.

CDN Security. As an important network infrastructure, CDN is favored by users for its accessibility and security protection [30]. According to reports [31], there is nearly one-fifth of the current Internet traffic transmitted through CDNs. Therefore, the security of CDNs has always been concerned by researchers. For example, Triukose et al. [3] proposed an attack which have abilities to exhaust the bandwidth of origin server by dropping the *front-end* connections rapidly. However, this attack has been proved ineffective in most CDNs [32]. Furthermore, DDoS protection of vulnerable can be directly nullified and abused to attack the origin server through *RangeAmp* attacks proposed by Li et al. [4]. They also mention that, an *OBR* attacker can set a small TCP receive window to make himself only receive little data. In Fact, we evaluate these attacks in our experiment, and find most CDNs tested can mitigate them now. In contrast, our proposed RangeFragAmp attacks can adapt to the new range request forward mechanism on the CDNs, and bypassing the DDoS protection provided by CDNs for the origin server. In addition, we also take economic factors into consideration. Actually, Our RangeFragAmp attacks can be used as a potential scheme of EDoS attack, which can effectively destroy the economic sustainability of websites.

Amplification Attacks. Our research is also a kind of amplification attack which has long been well studied. For example, Booth et al. [7] revealed that, an UDP amplification attack can reach an amplification factor of 556 by recruiting UDP servers on the Internet as reflectors. Mairos et al. [33] introduced a new breed of DNS amplification which has an amplification factor of 44. In fact, the

factor of amplification attack is enlarged with the introduction of CDN [34]. For example, *RangeAmp* attacks can even have an amplification factor of 43,300 [4]. Therefore, the harm of amplification attacks in CDN is more serious to a certain extent. RangeFragAmp attacks we proposed have a larger amplification factor than those in the traditional network. Even compared with *RangeAmp* attacks that same to use the range request mechanism of CDN, RangeFragAmp attacks have better concealment and adaptability to the capabilities of attackers. In fact, our experiment shows that the former has been mitigated by most CDNs.

7 Conclusion

In this paper, we revisited HTTP range request cased content fetching amplification issue and evaluated the deployed defenses of mainstream CDN providers. We propose a novel amplification attacks, RangeFragAmp attacks, which take advantage of the defect of the HTTP range request mechanism on CDNs. At the same time, we evaluated the feasibility and severity of the RangeFragAmp attacks in the wild. Lack of security considerations and arbitrary setting of fragment size in the planning stage are the root causes of the vulnerability. In addition, we disclosed our findings to CDN providers and offered our mitigation solutions to them. In the future, we hope to cooperate with CDN providers and academic researchers to discuss CDN security and potential improvement measures.

Acknowledgment. We would like to thank the research and development engineers of the CDNs. The fruitful discussion with them helps us better understand the working mechanism of CDN and the design principle behind it, which provides ideas for our subsequent research. At the same time, we are grateful to the reviewers for their comments and suggestions, their feedback helped us build a better job.

References

1. OpenCDN: The idea of traffic amplification attacks (2013). http://drops.wooyun.org/papers/679
2. Chen, J., et al.: Forwarding-loop attacks in content delivery networks. In: NDSS (2016)
3. Triukose, S., Al-Qudah, Z., Rabinovich, M.: Content delivery networks: protection or threat? In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 371–389. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04444-1_23
4. Li, W., et al.: CDN backfired: amplification attacks based on HTTP range requests. In: 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 14–25. IEEE (2020)
5. Radware: Amplification Attack. https://www.radware.com/security/ddos-knowledge-center/ddospedia/amplification-attack
6. CloudFront: What is a DNS amplification attack? (2020). https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/

7. Booth, T.G., Andersson, K.: Elimination of DoS UDP reflection amplification bandwidth attacks, protecting TCP services. In: Doss, R., Piramuthu, S., Zhou, W. (eds.) FNSS 2015. CCIS, vol. 523, pp. 1–15. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19210-9_1
8. Fielding, R., Lafon, Y., Reschke, J.: Hypertext transfer protocol (HTTP/1.1): range requests. IETF RFC7233, June 2014
9. Alibaba Cloud: Alibaba Cloud Content Delivery Network. https://www.alibabacloud.com/product/cdn
10. Tencent Cloud: Tencent Cloud Content Delivery Network. https://intl.cloud.tencent.com/product/cdn
11. Baidu AI Cloud: Baidu AI Cloud Content Delivery Network. https://intl.cloud.baidu.com/product/cdn.html
12. Huawei Cloud: Huawei Cloud Content Delivery Network. https://www.huaweicloud.com/en-us/product/cdn.html
13. Amazon Web Services: Amazon CloudFront Content Delivery Network. https://aws.amazon.com/cloudfront/. Accessed 10 Feb 2021
14. Liu, X., Yang, P., Dong, Y., Ahmed, S.H.: A comparative analysis of content delivery capability for collaborative dual-architecture network. In: Romdhani, I., Shu, L., Takahiro, H., Zhou, Z., Gordon, T., Zeng, D. (eds.) CollaborateCom 2017. LNICST, vol. 252, pp. 63–72. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00916-8_7
15. Tencent: Tencent cloud security white paper. Technical report, Tencent Cloud Security Team & Tencent Research Institute Security Research Center, June 2019
16. Wen, Y., Chen, Y., Shao, M.-L., Guo, J.-L., Liu, J.: An efficient content distribution network architecture using heterogeneous channels. IEEE Access **8**, 210988–211006 (2020)
17. Falchuk, B., Żernicki, T., Koziuk, M.: Towards streamed services for co-located collaborative groups. In: 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 306–315. IEEE (2012)
18. Biazzini, M., Serrano-Alvarado, P., Carvajal-Gomez, R.: Towards improving user satisfaction in decentralized P2P networks. In: 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 315–324. IEEE (2013)
19. Tencent Cloud: Enterprise Content Delivery Network User Guide Product Documentation. Tencent, April 2020
20. GlobalDots: How to Evaluate and Implement a Multi-CDN Strategy. https://www.globaldots.com/content-delivery-network-explained
21. Park, K., Pai, V.S.: Deploying large file transfer on an HTTP content distribution network. In: WORLDS (2004)
22. Hong, K., Kim, Y., Choi, H., Park, J.: SDN-assisted slow HTTP DDoS attack defense method. IEEE Commun. Lett. **22**(4), 688–691 (2018)
23. Deshpande, T., Katsaros, P., Basagiannis, S., Smolka, S.A.: Formal analysis of the DNS bandwidth amplification attack and its countermeasures using probabilistic model checking. In: 2011 IEEE 13th International Symposium on High-Assurance Systems Engineering, pp. 360–367. IEEE (2011)
24. Anagnostopoulos, M., Kambourakis, G., Kopanos, P., Louloudakis, G., Gritzalis, S.: DNS amplification attack revisited. Comput. Secur. **39**, 475–485 (2013)
25. Kashaf, A., Sekar, V., Agarwal, Y.: Analyzing third party service dependencies in modern web services: have we learned from the Mirai-Dyn incident? In: Proceedings of the ACM Internet Measurement Conference, pp. 634–647 (2020)

26. chinaz.com: CDN cloud real-time observation (2021). https://cdn.chinaz.com/. Accessed 15 Apr 2021
27. Alibaba Cloud: Configure range origin fetch (2021). https://help.aliyun.com/document_detail/27129.html. Accessed 30 Jan 2021
28. Parno, B., Wendlandt, D., Shi, E., Perrig, A., Maggs, B., Yih-Chun, H.: Portcullis: protecting connection setup from denial-of-capability attacks. ACM SIGCOMM Comput. Commun. Rev. **37**(4), 289–300 (2007)
29. Belshe, M., Peon, R., Thomson, M.: Hypertext transfer protocol version 2 (HTTP/2) (2015)
30. Luglio, M., Romano, S.P., Roseti, C., Zampognaro, F.: Service delivery models for converged satellite-terrestrial 5G network deployment: a satellite-assisted CDN use-case. IEEE Netw. **33**(1), 142–150 (2019)
31. Data Economy: Data economy frontline. Edge computing as the most misunderstood weapon of the IoT world (2019). https://www.clearblade.com/press/data-economy-frontline-edge-computing-as-the-most-misunderstood-weapon-of-the-iot-world/
32. Ife, C.C., Shen, Y., Stringhini, G., Murdoch, S.J.: Marked for disruption: tracing the evolution of malware delivery operations targeted for takedown. arXiv preprint arXiv:2104.01631 (2021)
33. Sieklik, B., Macfarlane, R., Buchanan, W.J.: Evaluation of TFTP DDoS amplification attack. Comput. Secur. **57**, 67–92 (2016)
34. Guo, R., et al.: Abusing CDNs for fun and profit: security issues in CDNs' origin validation. In: 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS), pp. 1–10. IEEE (2018)