End Users Get Maneuvered: Empirical Analysis of Redirection Hijacking in Content Delivery Networks

Anonymous Author(s)

ABSTRACT

The success of Content Delivery Networks (CDNs) relies on the mapping system that leverages the dynamically generated DNS records to distribute client's requests to a proximal server for achieving optimal content delivery. However, the mapping system is vulnerable to malicious hijacks, as (1) it is very difficult to provide precomputed DNSSEC signatures for dynamically generated records in CDNs and (2) even considering DNSSEC enabled, DNSSEC itself is vulnerable to replay attacks. By leveraging *crafted* but *legitimate* mapping between end-users and edge servers, adversaries can hijack CDN's request redirection and nullify the benefits offered by CDNs, such as proximal access, load balancing, and DoS protection, while remaining undetectable by existing security practices. In this paper, we investigate the security implications of dynamic mapping that remain understudied in security and CDN community. We perform a characterization of CDN's service delivery and assess this fundamental vulnerability in DNS-based CDNs in the wild. We demonstrate that DNSSEC is ineffective to address the problem, even with the newly adopted ECDSA that is capable of achieving *live signing* for dynamically generated records. We then discuss practical countermeasures against this redirection manipulation.

KEYWORDS

CDN; End-User Mapping; DNS; DNSSEC; Anycast

ACM Reference Format:

Anonymous Author(s). 2017. End Users Get Maneuvered: Empirical Analysis of Redirection Hijacking in Content Delivery Networks. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Content Delivery Networks (CDNs) play an important role in the Internet ecosystem by delivering a large fraction of the Internet content to end users with high availability, performance, and scalability. Typically, CDNs place a large number of edge servers (*a.k.a. surrogates*) at geographically distributed edge networks, enabling content caching and proximal access for end-users. The user requests for the content hosted by CDNs are served at the "edge" via request redirection to improve user-perceived performance and balance the load across server clusters. Moreover, CDNs are able to provide a security portal of protection mechanism against distributed denial-of-service (DDoS) attacks by redirecting users from overwhelmed nodes [23, 80]. Therefore, how to redirection or route client requests among edge server clusters is a central function of CDNs and its design choice is critical to CDNs.

Conference'17, July 2017, Washington, DC, USA
2017. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The majority of today's CDNs leverage the Domain Name System (DNS) as the core of their mapping systems to redirect a client's request into a nearby edge server. Since a DNS-based mapping system is based on real-time measurements of server and network conditions, it can provide fast, accurate, and fine-grained control for request redirection. Thus, its usage has been dominant in leading CDN vendors for operating a large number of edge servers, such as Akamai. However, such a DNS-based mapping system requires DNS records to be very dynamic, which restrains those DNS-based CDNs from authenticating their mapping DNS records by using DNSSEC signatures. This is because DNSSEC was originally designed for static records. Due to its prohibitively high computational overhead, traditional RSA-based DNSSEC is not a feasible solution to secure dynamic DNS records in the context of CDNs.

In this paper, we conduct a large-scale empirical study to investigate the security implications in the DNS-based mapping system of CDNs, which can be exploited by adversaries to hijack the operation of request redirection in a very stealthy manner. Our work makes following contributions:

- **Illustration of Redirection Hijacking Attacks in CDNs:** As DNSSEC is vulnerable to a replay attack, we illustrate that an adversary can utilize a *legitimate* mapping record (i.e., a *replayed* message) to override CDN's server selection and redirect a certain group of users to an edge server chosen by the adversary. Worrisomely, even the newly adopted Elliptic Curve Digital Signature Algorithm (ECDSA) that is capable of providing real-time DNSSEC signatures is ineffective to detect and prevent such attacks.
- **Characterization of Operational Practices of CDN's Request Routing:** To assess the magnitude of this vulnerability, we characterize the content delivery operations of popular CDN vendors and perform the threat analysis to elaborate on the ineffectiveness of DNSSEC against redirection hijacking via detailed case studies. We find that 16 out of 20 popular CDN providers suffer from various degree of security threats posed by redirection hijacking.
- **Measurement of Practical Impacts of Redirection Hijacking:** We quantitatively measure the practical impacts caused by redirection hijacking, such as performance degradation and cached content popularity. Moreover, we examine more serious threats, by which adversaries could exploit redirection hijacking to direct end-users to unresponsive edge servers, resulting in the nullification of CDN's benefits (e.g., load balance and DoS mitigation) and the violation of CDN's service commitments.
- **Challenges and Practical Considerations of Countermeasures:** Finally, we present the challenges of addressing this redirection hijacking from different perspectives, and elaborate

on corresponding countermeasures in practice and their limitations.

The remainder of this paper is organized as follows. In §2, we review the background of CDN operations and DNS security. In §3, we present the threat model and the redirection hijacking attack. In §4, we characterize CDN's operations and perform a large scale threat analysis, illustrating that DNSSEC is not an effective solution. We then discuss the impact of current practice and potential countermeasures in §5. We survey related work in §6, and finally we conclude the paper in §7.

2 BACKGROUND

2.1 Content Delivery Network

2.1.1 DNS-based Mapping System. The mapping system plays a critical role in CDN's request routing for directing each client's request to an appropriate surrogate, which is proximity to a client and has (1) sufficient resource capacity to be responsive and (2) high possibility with requested content cached, leading to an improved user-perceived performance (e.g., low latency). Traditionally, the mapping system uses a client's local recursive DNS resolvers (LDNSes or rDNSes) as the representation of the local area network where a client resides to determine the client's location and assign a neighboring edge server cluster. However, this approach has become inaccurate due to (1) poor location proximity between clients and their LDNSes [59, 69] and (2) increasing usage of public DNS services such as Google Public DNS and OpenDNS. To this end, the EDNS-Client-Subnet (ECS) extension [35] has been proposed to rectify the problem of location discrepancy between clients and their recursive DNS resolvers.

EDNS-Client-Subnet (ECS). Within ECS, the information of network prefix of a client's IP address (i.e., local network from which a query is sent) is included in the option field of a DNS query. Namely, this extension enables the DNS-based mapping system to use the direct knowledge of a client's location, instead of using its LDNS. A recent study by Chen *et al.* [32] showed that Akamai's end-user mapping[1] rolled out by ECS had been providing significant performance benefits for the clients behind public DNS services.

Load Balancing. The load balancing module of a DNS-based CDN such as Akamai typically selects proper surrogates by a two-level assignment [32, 58]: global load balancing and local load balancing. The global load balancing relies on network measurements to select a server cluster, typically geographically close to a client's network. Then, the local load balancing assigns the individual server(s) from the chosen cluster, leveraging the combined information such as responsiveness and capacity.

2.1.2 Anycast Routing. The deployment of the DNS-based dynamic mapping requires extra infrastructure and operational supports. Therefore, some new CDN providers then enable their CDN platforms by leveraging anycast routing, which announces the same IP address(es) from multiple distributed endpoints. BGP routing protocol selects a shortest Autonomous System (AS) path to reach each advertised IP address block, and thus the end users located in different areas will be directed to different topographically-closest locations via BGP routing.

Since anycast-based CDNs rely on the Internet routing protocols for request redirection, conceptually they are immune to redirection hijacking attacks. However, we observe that in practice some anycast CDNs are also leveraging DNS-based mapping to improve accuracy and performance, making themselves vulnerable to request routing manipulation.

CDNs may leverage anycast in different strategies: anycasting nameservers, or anycasting web servers, or both. Note that our study only involves the way that a CDN directs users to web servers. Specifically, anycasting nameservers means that clients will connect the nameservers via their anycast addresses, but does not affect the process of end-user redirection. In particular, if a CDN utilizes anycast DNS but still uses DNS-based redirection, it will also be vulnerable to redirection hijacking attacks.

2.2 DNS Cache Poisoning Attack

DNS provides a vital naming service for users to locate Internet resources by translating domain names to numerical IP addresses, and thus the correctness of DNS resolution is the fundamental anchor for the operation and security of the Internet. Due to its crucial role in accessing Internet services, DNS is an attractive target of adversaries and has been exploited for various malicious purposes. One of the most serious threats to DNS is that adversaries trick a resolver to accept fraudulent DNS records as legitimate responses from authoritative nameservers, known as record injection or cache poisoning attacks [24, 29, 49, 73].

DNS cache is intrinsically vulnerable to record injection, because a recursive resolver cannot ensure whether a received response is from a legitimate authoritative nameserver or a miscreant entity. The general practical approach for mitigating a cache poisoning attack involves the *challenge-response* defenses [48], including transaction-ID (TXID) randomization, source port randomization, or the 0x20 encoding [37], in order to enable a resolver to validate the legitimacy of received responses via the randomized values within the requests.

Although those countermeasures increase the difficulty of injecting fraudulent records, insufficient adoptions and deployment [41, 43, 70] have made many rDNSes still vulnerable to cache poisoning attacks. Large-scale DNS poisoning attacks are still widely observed on the Internet [18, 21, 22]. Furthermore, all those efforts aiming to increase the entropy of DNS queries are only effective against the *off-path* attackers; an adversary, which can monitor network traffic and interpret the transaction packets, is still able to construct a forged DNS response with correct parameters to bypass all the challenge-response defenses and pollute the content of cache, i.e., launch a Man-in-the-Middle (MitM) attack.

2.3 DNSSEC

In order to secure the process of DNS resolution, especially against MitM attacks, DNSSEC [28] uses the digital signatures to validate DNS responses. Within DNSSEC, each resource record set (*RRset*)

[1]In [32], the "*end-user mapping*" is used to dedicatedly describe ECS-based mapping (compared to the *NS*-based mapping which uses LDNSes). To be clarified, in this paper we use "*DNS-based mapping*" to include both ECS-based and NS-based mapping. In most cases, unless specified, we do not differentiate the "DNS-based mapping" and the "end-user mapping" since they have identical implication in the context of dynamic mapping.

is signed and verified by public key cryptography: a recipient of a signed RRset (i.e., *RRSIG* record) validates the signature via the public key (i.e., *DNSKEY* record) of signer. The *trust of chain*, starting from "*trust anchor*" at the root zone, ensures that each key is trusted and able to be validated (i.e., via *DS* record provided by its parent zone to authorize the *DNSKEY* which is used to sign the RRset).

DNSSEC Zone Enumeration. Within DNSSEC, to provide authentication for negative response (i.e., authenticated denial of existence), a Next-SECure (NSEC) record lists and signs a pair of lexicographic consecutive names in the zone, indicating that no names exist between the NSEC's owner name and the "next" name. However, NSEC records expose the existence of names in the zone; this then allows adversaries to enumerate NSEC records and walk through the zone space to learn all the (sub)domains and associated IP addresses, i.e., the zone enumeration attack, resulting in undesired policy violation or more complex attacks [55].

In order to make the zone enumeration more difficult, the alternative NSEC3 record [55] lists the cryptographically hashed names rather than valid (sub)domain names. However, it is still vulnerable when adversaries apply an dictionary attack by querying non-existent names and guessing real names [11, 40]. Thus, NSEC5 [40] is then proposed to replace the NSEC3's *unkeyed* hash with a new *keyed* hash generated by separate secondary keys.

Another technique to mitigate the zone enumeration is "On-line Signing" [72, 82] (i.e., White Lies" [39]). Instead of disclosing real domains or precomputed hashes, on-line signing creates on-demand signature, proving non-existence for a specific name by listing its derived predecessor and/or successor. However, this approach has two major drawbacks [82]: (1) with traditional RSA algorithm, it introduces significant computational load for authoritative nameservers to generate the real-time signatures, resulting in potential denial of service attacks, and (2) the primary private keys have to be distributed among nameservers, increasing the risk of key leakage.

Live Signing by ECDSA. To mitigate the zone enumeration and DNSSEC amplification attacks [78], the Elliptic Curve Digital Signature Algorithm (ECDSA) [44] has been employed as an alternative cryptosystem for DNSSEC [79]. Different from traditional RSA-based schemes, ECDSA leverages the Elliptic Curve Cryptography (ECC) to generate signatures with dramatically reduced computational overhead and signature size. More importantly, while validating an ECDSA signature is fundamentally slower than validating an RSA signature [44, 76], the significantly reduced computational overhead (about 10x faster in signing [12]) allows ECDSA to sign all of the necessary RRSIG records "on-the-fly" [12], i.e., a *live signing* mechanism, providing a potential solution in the context of dynamically generated records at the "edge" of the Internet.

3 THREAT MODEL

3.1 Attacker Model

The key feature of a redirection hijacking attack is that an adversary can inject *crafted* but *legitimate* records to a recursive DNS resolver to manipulate the dynamic mapping inside CDNs. We assume that the adversary is capable of bypassing the challenge-response mechanism and injecting such *legitimate* records into DNS caches, regardless if DNSSEC is used since DNSSEC itself is vulnerable to replay attacks. Specifically, the Man-in-the-Middle (MitM) attacker can easily bypass the countermeasures of randomization by sniffing the network packets and observing those parameters. On the other hand, the off-path adversaries can guess the authentication parameters (i.e., source port number and TXIDs) effectively by applying different techniques (e.g., fragmentation attacks [42, 70] or socket overloading [43]) against the insufficient randomization or vulnerable implementations [41], to launch the record injections remotely and affect a large number of recursive DNS resolvers. A recent work [51] has demonstrated that, with the feasibility of exploiting MitM attacks and those parameter-guessing techniques, more than 92% of current DNS platforms on the Internet are still vulnerable to record injection; even the popular public DNS platforms are vulnerable to the *indirect injection*, in which a poisonous record is injected in advance and becomes effective after some other records expires.

Within CDNs, we assume that adversaries do not have to harvest the surrogate servers [31, 75] or profile CDN's mapping algorithm; they only need to use several selective mapping records to override the CDN's server selection.

3.2 Redirection Hijacking Attack

In comparison to the normal operations of a DNS-based mapping system in CDN, Figure 1 illustrates how a redirection hijacking attack works: an adversary exploits the dynamic end-user mapping to manipulate an end-user's access to edge networks. Normally, the Content Provider delegates its name resolution to the CDN vendor's mapping system, typically via either CNAME redirection as shown in Figure 1 or directly hosting the NS records in CDNs. When a client's request for a content object (❶) is redirected into a CDN's nameserver (❷), the mapping component examines the incoming queries (e.g., the client's IP prefix in ECS), performs real-time topological mapping based on network measurements, and returns an optimized assignment (❸ ∼ ❹) that directs the client to a close, responsive edge server [32] (❺).

Since the dynamic mapping between end-users and edge servers makes it impractical to pre-sign a mapping record with traditional RSA-based DNSSEC, we also consider that the ECDSA could be used as the alternative solution to provide on-demand signatures for those dynamically generated records in CDNs. However, even the mapping records with ECDSA signatures are still vulnerable to redirection manipulation. This is because (1) in operational practices, the validity period of a DNSSEC signature (including ECDSA) should be long enough[2] to enable easy administration and avoid query load peaks (see §4.4.1 in RFC 6781 [52]) and (2) the validation of DNSSEC signature cannot detect if a message is forwarded or replayed to a different recipient by a third party. An adversary can simply fetch a legitimately signed mapping record that was used or is being used for a different client's network and inject it into the resolver's cache. Because the injected record, which is generated by a legitimate authoritative nameserver but for a different group of clients, carries a valid signature, the resolver will accept it for caching after a successful signature validation. Once the injected record is accepted, the client requests will be redirected

[2]Cloudflare's ECDSA-based signatures are with the validity period of two days. The expiration time of traditional RSA-based DNSSEC signature in practice is normally set to one month [52].

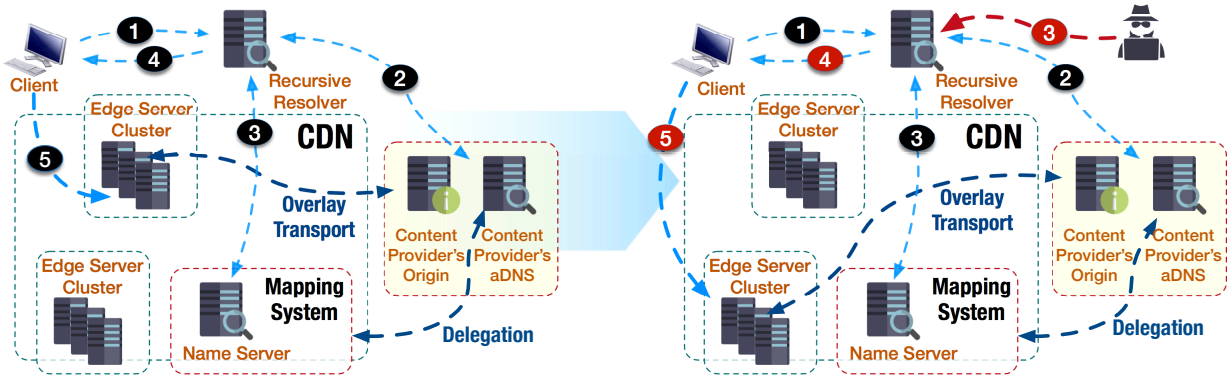

Figure 1: Illustration of a Redirection Hijacking Attack. (An adversary replays and injects a legitimate record associated with suboptimal or non-responsive edge servers, resulting in the maneuvered end user redirection while still passing the DNSSEC validation.)

to a non-optimal edge server chosen by the adversary, typically heavily loaded and geographically distant from the clients, or even an unresponsive edge server to interrupt the client accesses to the service hosted by CDNs.

We further note that such an attack can be successful even under the environments with strong security settings. Due to the nature of replay attacks in the redirection hijacking, neither the client end nor resolver signature validations can detect the manipulation.

4 ATTACK ASSESSMENT

To assess the magnitude of redirection hijacking in CDNs, we present the characterization of CDN's request routing and conduct a detailed threat analysis to demonstrate the vulnerability of DNS-based CDNs to the malicious manipulation of end-user redirection, even with DNSSEC. Then, we quantitatively measure the practical impacts, such as performance degradation and caching popularity, and explore the more serious threats posed by the redirection hijacking, which nullify CDN's load balancing and DoS protection.

4.1 Methodology

In order to identify the CDN platforms that are vulnerable to redirection hijacking, we measure the popular commercial CDNs across the Internet to characterize their configurations and operations. To do so, we set up a group of geographically distributed vantage points on the machines in different Amazon EC2 regions (us-east-1, us-west-2, ap-northeast-1, ap-southeast-2, ap-south-1, eu-central-1, eu-west-1, and sa-east-1, as shown in Figure 2) to retrieve the DNS resolution results for customer websites hosted in each CDN provider. Then, we examine the strategies of request routing and analyze the practical impacts, including the more serious threats than performance degradation.

More specifically, we empirically investigate the patterns of content delivery for CDN vendors by taking the steps as follows:

- First, we simply search the official blog articles, technical documents, and announcements published by each CDN vendor, as well as the external technical blogs (e.g., [1, 2]), to learn the details of content delivery mechanisms.
- We then verify our findings by studying DNS configurations and resolution results from distributed vantage points for a list of customers of each CDN provider, which are gathered by available utilities (e.g., [3, 4]) and the customer list/case studies presented in CDN's websites. For example, an identical A RRset should be fetched from different locations when global anycast routing is utilized, and diverse A RRsets would be observed when DNS-based dynamic mapping is used.
- Finally, we crosscheck the domain/host names and IP addresses obtained from the DNS resolution by retrieving their information from publicly available passive DNS databases [5, 25] to validate the resolution results.

Figure 2: Vantage Points for Resolution

4.2 Characterization Overview

Request routing in CDNs mainly consists of two consecutive steps:[3] *domain delegation* and *surrogate selection*. In the domain delegation, the Content Providers (CPs) delegate the domain resolution to CDN vendors. In the surrogate selection, CDNs redirect a client's request to a proximal edge server. In essence, these two steps determine how CDNs enable their service infrastructures to be located and accessed by end users. Since CDNs have different ways to perform

[3]The higher level techniques of request routing [30] such as application-level request routing are only suitable for large file delivery due to extra latency [32], and thus we only consider those techniques when discussing countermeasures (see Section 5.4).

Table 1: Characterization of CDNs' Request Routing and DNSSEC Provision. The "●" indicates that adversaries may be able to manipulate the end-user redirection that results in more serious damage (§4.4). The "○" indicates that the record suffers from limited form of dynamic vulnerability that may not cause serious threats such as service interruption.

CDN	Domain Delegation	Surrogate Selection	DNSSEC	Dynamics	
			A	CNAME	A
Akamai	CNAME Chain	DNS-based Mapping (ECS)	×		●
Cachefly	CNAME/NS Hosting	Anycast Routing	Feasible		
CDN.net	CNAME	DNS-based Mapping	×		●
CDN77	CNAME	DNS-based Mapping (ECS)	×		●
CDNetworks	CNAME	DNS-based Mapping (ECS)	×		●
CDNlion	CNAME	DNS-based Mapping	×		●
CDNsun	CNAME	DNS-based Mapping	×		●
ChinaCache	CNAME/CNAME Chain	DNS-based Mapping (ECS)	×		●
CloudFlare	CNAME/NS Hosting	Anycast Routing	✓		
CloudFront (Amazon)	CNAME/NS Hosting	DNS-based Mapping (ECS)	×		●
EdgeCast (Verizon)	CNAME/CNAME Chain	Hybrid Type I	Feasible		○
Fastly	CNAME	Hybrid Type II	×		●
Highwinds	CNAME	Anycast Routing	Feasible		
Incapsula	CNAME	Hybrid Type I	Feasible		○
KeyCDN	CNAME Chain	DNS-based Mapping (ECS)	×	●	●
LeaseWeb	CNAME	DNS-based Mapping	×		●
Limelight	CNAME	DNS-based Mapping	×		●
MaxCDN/NetDNA	CNAME	Anycast Routing	Feasible		
Rackspace	CNAME Chain	DNS-based Mapping (ECS)	×		●
cedexis (*MultiCDN*)	CNAME Chain	N/A	×	●	

domain delegation and surrogate selection, we characterize CDNs' request routing with respect to their two redirection steps. Table 1 summarizes the request routing and DNSSEC provision in popular CDN vendors.

Domain Delegation. The domain delegation is used to forward each client's request from the origin of Content Providers (CPs) to a CDN's platform. The most common domain delegation mechanisms are CNAME redirection and NS hosting.

- **CNAME Redirection**: The CNAME record enables a domain name to be resolved via an alias. By pointing a CP's domain to a domain provisioned by CDN via CNAME, a client's request will subsequently be redirected to a CDN's domain name and resolved by the CDN's nameservers.
- **NS Hosting**: An alternative approach of domain delegation is to designate CDN-provided authoritative nameservers in NS records of a DNS referral response, which is generated by the CP's authoritative nameservers and then is received by clients. Consequently, the DNS resolution of the CP's domain will be fully operated by CDN.

From Table 1, we can see that all CDN vendors provide CNAME redirection to enable the CPs to delegate their DNS resolution to CDNs. By contrast, only three CDN vendors also support NS hosting for domain delegation. Given the prevalent use of CNAME in CDNs, however, we note that the integrity of CNAME records has been widely disregarded on the Internet. This is because (1) typically, the first-level front-end CNAME redirection occurs at the CP's authoritative DNS infrastructure, which is mainly out of control of CDNs, (2) the CP's authoritative nameservers lack of motivations to sign CNAME records, due to the dynamic mapping in the following surrogate selection, (3) in some cases, dynamic CNAME mapping exists in CDN's platforms (see Section 4.3.1), and (4) many CDN providers leverage multiple CNAME records (i.e., CNAME chain in Table 1) to facilitate their platform management (e.g., enabling customers to employ different services by being mapped to different CNAMEs), which causes that traversing signed CNAME records are significantly expensive for recursively validating DNSSEC signature for each CNAME record. We will discuss the technique of "CNAME Flattening" in Section 5.3 to mitigate the security threat of CNAME in CDNs.

Surrogate Selection. The surrogate selection falls into two fundamental approaches: DNS-based and anycast-based. Table 1 shows that the DNS-based mapping is still dominant in CDNs and ECS has been widely supported, especially for those vendors operating a large-scale infrastructure, such as Akamai and Amazon. However, more recent vendors are more likely to build their platforms with anycast routing to leverage its easy and robust deployment. We also observe that some CDN vendors have employed the different hybrid system design by leveraging both DNS-based mapping and

www.dell.com.	3600	IN	CNAME	www1.dell-cidr.akadns.net.
www1.dell-cidr.akadns.net	3600	IN	CNAME	cdn-www.dell.com.edgekey.net.
cdn-www.dell.com.edgekey.net.	21600	IN	CNAME	cdn-www.dell.com.edgekey.net.globalredir.akadns.net.
cdn-www.dell.com.edgekey.net.globalredir.akadns.net.	3600	IN	CNAME	e28.x.akamaiedge.net.
e28.x.akamaiedge.net.	20	IN	A	104.117.80.33

Figure 3: An Example of DNS-based End-User Redirection by CNAME (Akamai**)**

anycast routing to improve the performance of their global content deliveries. In the following, we will elaborate on those different patterns for the operations of CDNs' request routing and analyze the security threat of redirection hijacking caused by the dynamic surrogate selection and the ineffectiveness of DNSSEC via case studies.

4.3 Threat Analysis

4.3.1 DNSSEC (Live Signing) is NOT a Solution: Case Studies. DNSSEC is proposed as a foundational system-wide solution to DNS vulnerabilities, especially for the record injection by MitM attacks. Here we depict detailed case studies to analyze the vulnerability under different CDN deployment patterns. We demonstrate the infeasibility of providing pre-computed DNSSEC signatures in the dynamic context of DNS-based CDNs. As we discussed in Section 2.2, the root cause is that the traditional RSA-based signature algorithm cannot achieve on-demand signature in real-time due to its high computational cost.

Subsequently, for the case studies, we also examine the scenarios when all necessary signature operations can be efficiently performed. To do so, we assume that (1) the CNAME records would be secured by adding corresponding signatures, and (2) CDNs are able to generate on-demand DNSSEC signatures to sign the dynamic mapping records efficiently, such as the ECDSA-based implementation that has been used in CloudFlare's platform [12].

Case Study of End-User Mapping: Akamai. Exemplified by Akamai, Figure 3 shows a typical resolution chain by CNAME redirection and the end-user mapping system rolled-out by ECS [32]. Specifically, the domain of the content provider is first translated to a domain provisioned by Akamai's CDN via CNAME. Afterwards, the CDN's nameservers take over the resolution and finally an A record is dynamically generated by the end-user mapping subsystem to assign an edge server with optimized performance such as responsiveness and capacity, based on the location estimation of the end-user's IP address carried in ECS extension.

Due to the diversity of mapping records and more than 233,000 servers within more than 1,600 networks in Akamai's CDN [7], it is inefficient and impractical to predetermine or predict the server assignment for each customer and provide a pre-computed DNSSEC signature, resulting in the fundamental vulnerability to record injection attacks. An adversary is able to exploit this vulnerability to hijack redirection and mislead end users to a different domain controlled by the adversary. We note that such a threat can be mitigated by employing ECDSA-based signature, as ECDSA is capable of dynamically signing the records. However, given the adoption of ECDSA, the dynamic mapping is still vulnerable to the redirection hijacking attack as mentioned in Section 3.2.

It is worth noting that, including Akamai, some DNS-based CDN vendors also provide DNS-hosting services with anycast routing and optional DNSSEC signature (e.g., Akamai's Fast DNS [8]). However, this type of service aims to protect the DNS infrastructure itself only; if a customer enables the content delivery, dynamic A records are still used to direct end users to edge servers and thus cannot be protected by DNSSEC.

Case Study of Anycast: CloudFlare. Anycast announces the same IP address(es) from multiple locations and relies on the BGP routing protocols to perform the front-end redirection. Therefore, the content providers leveraging anycast-based CDNs would have identical A record(s), which are static, and thus the anycast-based CDNs are able to secure the integrity of RRsets with either ECDSA-based or pre-computed RSA-based signatures. This makes the anycast-based CDNs immune to redirection hijacking.

The examples below show the configurations of CloudFlare with the domain delegation of CNAME and NS hosting, respectively. In both cases, the returned signed A records are with the global anycast addresses, and so there is no risk of redirection hijacking. However, we also note that although the Content Provider enables DNSSEC in the CloudFlare's CDN, the integrity of its CNAME record has been disregarded, which still leads to the risk of domain hijacking.

$ DNS resolution for domain using **NS Hosting**

filippo.io.	NS	beth.ns.cloudflare.com.	
filippo.io.	NS	jim.ns.cloudflare.com.	
filippo.io.	DS	...	
filippo.io.	RRSIG	DS	*[ECDSA signature]*
blog.filippo.io.	A	104.20.145.15	
blog.filippo.io.	A	104.20.144.15	
blog.filippo.io.	RRSIG	A	*[ECDSA signature]*

$ DNS resolution for domain using **CNAME**

www.martindale.com.	CNAME	www.martindale.com.cdn.cloudflare.net.	
www.martindale.com.cdn.cloudflare.net .	A	104.18.60.26	
www.martindale.com.cdn.cloudflare.net.	A	104.18.61.26	
www.martindale.com.cdn.cloudflare.net.	RRSIG	A	*[ECDSA signature]*

Note that ECDSA provides CloudFlare the solution to sign their records "on-the-fly" at the edge, but its invulnerability to the end-user manipulation is mainly due to anycast routing rather than ECDSA signing.

Case Study of Hybrid Type I – Regional Anycast: Incapsula. Incapsula enables a hybrid strategy for the request routing, where the DNS-based mapping is used to preliminarily determine the geographic area of end users and a *regional anycast* address is used to serve a specific region. That is, using the regional anycast, a world-wide network is divided into different regions (typically 5-7 regions based on the continents), and within each region, identical anycast addresses are advertised and used to direct end users in this region to a close PoP (Point-of-Presence).

On one hand, such a type of hybrid strategy mostly leverages the anycast routing and is able to use DNSSEC to secure the certain number of static DNS records within each region. On the other hand, it still introduces the dynamic DNS records as DNS-based mapping

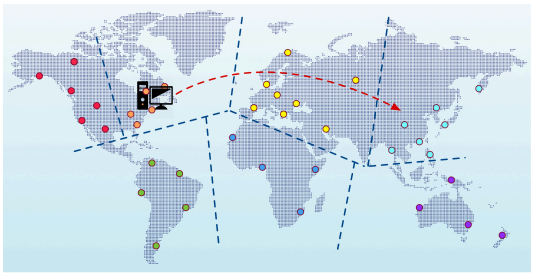

Figure 4: Illustration of Redirection Hijacking with Regional Anycast. (The global platform is divided into different regions, each of which leverages the anycast routing within the region. A redirection hijacking can force end users to access the suboptimal or unresponsive edge servers located within a remote region.)

is used, and thus it is vulnerable to the redirection hijacking for maneuvering the end-user access. Figure 4 illustrates an example of a global network using regional anycast and its susceptibility to redirection-hijacking. Even with the adoption of DNSSEC, similar to the case of DNS-based redirection, an adversary can inject a legitimate anycast record associated with the end users from a different region, directing the victim users to the edge servers that serve the clients in another continent.

Case Study of Hybrid Type II – Separate Anycast/Unicast: Fastly. Instead of adding the ECS support, Fastly addresses the problem of location discrepancy in a different hybrid strategy: (1) in a normal case, the traditional NS-based mapping is utilized to direct end users to close PoPs; (2) anycast addresses are used to answer the queries from public DNS resolvers. Under such a strategy, the end users behind ISPs leveraging centralized DNS infrastructures will still suffer from the problem of location discrepancy. Moreover, those clients that do not use public DNS services are vulnerable to redirection hijacking, as in the case of DNS-based mapping.

Case Study of Dynamic CNAME: KeyCDN. Unlike other DNS-based CDNs, KeyCDN leverages CNAME to map the CP's domain to a close PoP first, and then assign an appropriate edge server within the PoP via A records.

$ DNS resolution from us-west

ja.onsen.io.	CNAME	jaonsenio-4ecf.kxcdn.com.
jaonsenio-4ecf.kxcdn.com.	CNAME	p-usse00.kxcdn.com.
p-usse00.kxcdn.com.	A	76.164.234.2

$ DNS resolution from us-east

ja.onsen.io.	CNAME	jaonsenio-4ecf.kxcdn.com.
jaonsenio-4ecf.kxcdn.com.	CNAME	p-uswd00.kxcdn.com.
p-uswd00.kxcdn.com.	A	107.182.231.101

The dynamic CNAME mapping introduces another potential attack vector for redirection hijacking via CNAME record. Similar to hijacking a dynamic A record, an adversary could inject a legitimate CNAME record associated with a remote non-optimal PoP to degrade the user-perceived performance, even under the availability of DNSSEC live signing enabled by ECDSA.

On the other hand, with the usage of ECDSA, the redirection hijacking for dynamic A records would not cause significant performance degradation because all valid A records are being mapped to the IP addresses within the nearby PoP assigned by CNAME. However, the adversaries can still leverage legitimate records to redirect users to the IP addresses of unresponsive edge servers within PoP to nullify the DoS protection and interrupt the end-user accesses for the victim service.

Case Study of Multiple-CDN Deployment: cedexis. We then investigate the deployment with multiple CDN providers (*a.k.a. CDN Brokers* [61, 62]). A typical deployment pattern of multiple CDNs leverages Global Traffic Management (GTM) as the first-level redirection, where the GTM platform directs end users to a selected appropriate CDN provider:

$ DNS resolution from us-east

www.lequipe.fr.	CNAME	2-01-273c-0023.cdx.cedexis.net.
2-01-273c-0023.cdx.cedexis.net.	CNAME	lequipe-fr.lequipe.netdna-cdn.com.
lequipe-fr.lequipe.netdna-cdn.com.	A	94.31.29.248

$ DNS resolution from ap-northeast

www.lequipe.fr.	CNAME	2-01-273c-0023.cdx.cedexis.net.
2-01-273c-0023.cdx.cedexis.net.	CNAME	www.lequipe.fr.edgekey.net.
www.lequipe.fr.edgekey.net.	CNAME	e7130.g.akamaiedge.net.
e7130.g.akamaiedge.net.	A	104.116.83.6

In the example above, the cedexis's GTM platform [10] is responsible for choosing an appropriate CDN vendor according to the location of a client and the real-time performance of CDNs in this area. First, under such a deployment, the diversity of A records depends on the strategy of each CDN's request routing. The clients accessing the website via NetDNA would not be vulnerable to redirection hijacking for A records, due to the use of global anycast (assuming signed anycast A records), but the clients directed by Akamai will suffer the risk of hijacked redirection mappings.

Since the selection of CDN providers is performed via CNAME redirection, it introduces dynamic CNAME mappings. Thus, DNSSEC live signing cannot prevent the redirection-hijacking attacks, in which legitimate records are injected to redirect users to arbitrary non-optimal CDN providers, nullifying performance improvements offered by both GTM and CDN platforms.

Summary. The vulnerability of CDNs to redirection hijacking stems from the dynamic characteristics of DNS records used for the request routing, which gives adversaries a chance to maneuver CDN's user redirection by injecting crafted but legitimate DNS records. We summarize the features of dynamic mapping for CNAME and A records in Table 1, respectively. The DNS-based CDNs are widely vulnerable to redirection hijacking, but the CDNs using global anycast for the request routing are immune to redirection hijacking, due to the static mapping of DNS records. Specifically, CloudFlare is the only CDN vendor providing DNSSEC signatures for A records to its customers, by leveraging its global anycast routing and ECDSA-based DNSSEC implementation. Also, we consider other CDN vendors with anycast routing of being capable of supporting DNSSEC signatures without DNS dynamics, labeled as "Feasible" in Table 1.

Note that the DNSSEC provision summarized in Table 1 involves only the capacity of signing the CDN-issued records for the request routing; the CPs may still be able to sign their records for origin sites, but the request routing would not be protected by their signatures since the mapping records will be provided by CDNs. We argue that this has been a foundational obstacle for the DNSSEC adoption

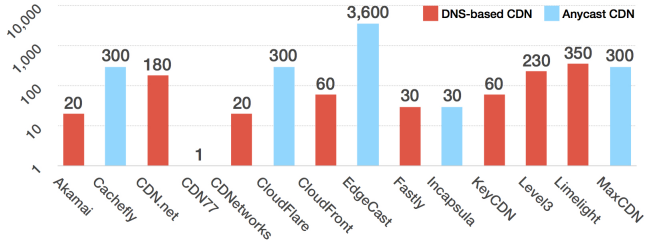

Figure 5: TTL

on the Internet, especially for the top websites leveraging CDNs to provide worldwide services.

4.3.2 TTL. We list the TTL values of the DNS records for surrogate assignment in Figure 5. The DNS-based CDNs use shorter TTL values in their dynamic A records for fast traffic redirection and load balancing, typically less than 300 seconds. Most of anycast CDNs have the TTL values of A records at 300 seconds, while Edgecast has a larger value at one hour and Incapsula leverages a very short value at 30 seconds.

The length of TTL in a normal DNS record has a significant impact on the possibility of DNS poisoning, because the short TTLs force the recursive resolver to more frequently perform DNS lookups, which grants adversaries more chances (i.e., more frequent "windows of opportunity") to perform the record injections [48]. With DNSSEC enabled, we will craft records based on the legitimate records with valid signatures that are re-used or replayed. Thus, the prevalent use of short TTL values in normal DNS records indeed increases the possibility of injecting replayed records.

Passive Analysis is NOT effective. Since CDNs typically utilize short TTL values in dynamic mapping records and adversaries usually intend to use larger TTLs in injected records to cause more damage, intuitively, a dynamic record with large TTL value may indicate that it is highly likely to be a crafted mapping. However, popular large-scale passive DNS databases do not enable their sensor servers to capture the TTL information in the traces so that such a manipulation might not be detected via passive DNS databases.

4.3.3 Performance Impact. We analyze the performance impact caused by redirection hijacking, in which adversaries inject the records to deliberately direct end users to a geographically distant non-optimal site.

Performance matters. User experience is extremely important to the business of CPs, especially eCommerce sites [9, 32]. Thus, the performance benefits provided by CDNs become critical to the CPs. A prior work [27] observes that even little difference in CDN's performance could cause significant financial gain/loss.

Performance metrics. Similar to the study [32], we measure the following metrics to characterize the potential performance impact when an end user is diverted from optimal edge servers by redirection hijacking.

- Round-Trip-Time (RTT): the RTT measures the propagation delay when the packets traverse the networks, which indicates the quality of the selected network path and is significantly dominated by the distance between two endpoints.

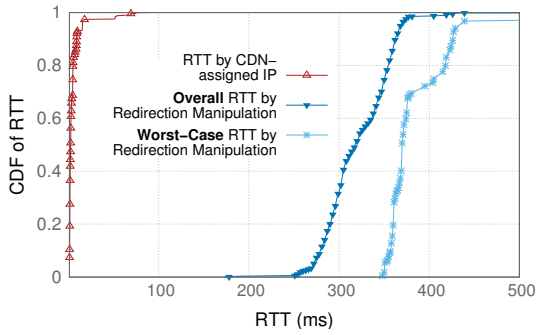

Figure 6: CDF for the Round-Trip Time

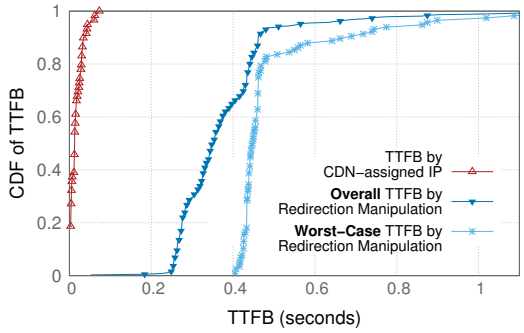

Figure 7: CDF for the TTFB

- Time-to-First-Byte (TTFB): the TTFB measures the amount of time between when the first byte of requested content is received and when the client issues the request.
- Content Download Speed: unlike the study [32] measuring the page download time, we use the download speed measured by the `curl` utility because the `curl` does not support concurrent connections for embedded contents.

Methodology. We leverage the DNS records obtained via the probes from distributed Amazon regions, as shown in Figure 2, and use the same technique for launching a cache penetrating attack presented in [75], where the `curl` utility is used to bypass CDN's server assignment by replacing the normal host header with a (distant) non-optimal IP address in HTTP request. A recent work [33] verifies that such a technique still works for all CDNs in their study. For example, to fetch a content object from an edge server located in Asia as the representation of end users in east coast of Unite States, we issue the following request at a host in the Amazon region of us-east-1:

```
curl -H Host:i.dell.com -O http://104.78.87.26/sites/
     imagecontent/products/...inspiron-15-7000-gaming-pdp-
     polaris-01.jpg
```

Our experiments are specifically performed based on Akamai's CDN platforms. We manually obtain a list of content objects from popular CDN-hosted sites (dell.com, apple.com, and walmart.com), including static web pages (.html and .css), dynamically generated web pages (embedded search keywords in URLs), images, documents, and medium-sized download files, with a variety of sizes

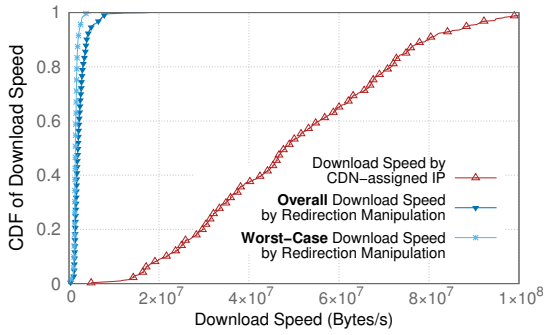

Figure 8: CDF for the Download Speed

from 500K to 50M. We download those web contents by using the `curl` utility to evaluate the performance impact experienced by end users under redirection hijacking.

For each metric presented above, we report the measured results associated with the optimal surrogate assignment and the redirected non-optimal surrogates, respectively. In addition, we identify a redirected site with the most significant performance degradation for each vantage point, and plot it as the worst case in Figures 6 - 8.

Round-Trip-Time. RTT is a purely underlying network latency and the most simple and straightforward performance metric of a network connection and user experience. Figure 6 shows that for the optimal assignment of the CDN's mapping system, the RTTs are mostly less than 30ms; but the hijacked redirections typically significantly increase the RTT latency to around 300ms, and in the worst case, the RTTs are increased to around 350 - 450ms.

Time-to-First-Byte. Since TTFB involves both the network latency and the aspects that are not affected by the mapping decisions, such as the construction and compression of a web page, we only include the measured results for web pages. Figure 7 illustrates similar impacts of TTFB in comparison to RTT. Note that our results show lower TTFBs than the results reported in [32], probably due to less dynamics of the web pages we requested.

Download Speed. Figure 8 shows the measured speeds for the file downloads. The results from optimal mapping decisions vary, but the cases under redirection hijacking show a significant decrease in their file download performance.

4.3.4 Scope of Impact. As discussed before, both the CNAME and `A` records for the CDN's request routing could be exploited by redirection hijacking. We then study whether hijacking a single record can cause collateral damage for other domains. Table 2 summarizes the scope of impact for those CDNs vulnerable to redirection hijacking. If CNAME records are unsigned, hijacking a CNAME record itself will just affect the domain associated with this record in all cases, since in these CDNs there is no canonical name being reused among CPs. In other words, there is no shared name appeared in the "left-side" of a CNAME record. However, if CNAME could be signed, only KeyCDN's dynamic CNAME poses the threat of hijacking a single domain. Meanwhile, in some CDNs there could be multiple (sub)domains being mapped to the same CNAME alias, i.e., a shared name appears in the "left-side" of an `A` record, and thus hijacking such `A` records would have collateral damages for those "co-resident" (sub)domains.

Table 2: Impact of a single record hijacking (*CDNs with global anycast that are invulnerable to the redirection hijacking have been excluded.*)

CDN	CNAME		A (signed)	
	Single Domain (unsigned)	Single Domain (signed)	Single Domain	Multiple Domain
Akamai	✓		✓	
CDN.net	✓		✓	
CDN77	✓		✓	
CDNetwork	✓		✓	
CDNlion	✓		✓	
CDNsun	✓		✓	
ChinaCache	✓		✓	✓
CloudFront	✓		✓	
EdgeCast	✓		✓	✓
Fastly	✓		✓	✓
Incapsula	✓		✓	
KeyCDN	✓	✓	✓	✓
LeaseWeb	✓		✓	
Limelight	✓		✓	✓
Rackspace	✓		✓	

4.3.5 Domain Sharding. The *domain sharding* (or *content segregation*) [6] technique is typically used to increase the amount of simultaneous connections by utilizing multiple domains. For example, www.dell.com is directed to e28.x.akamaiedge.net but all embedded images are served via i.dell.com, which is directed to e28.g.akamaiedge.net. Although this technique also distributes the connections to different domains among multiple edge servers, in such a case poisoning a portal domain (i.e., www.dell.com) is sufficient to affect the accessibility of most end users.[4]

4.3.6 Impact of CDN Caching. In addition to the issues discussed above, we are aware of that redirection hijacking may also have a subtle impact upon the caching system of CDNs. The caching system is the important building block of a CDN's infrastructure, providing accelerated access for static and popular contents. The cache hit ratio is a critical metric to CDN's performance, since a cache miss may cause extra latency for fetching the requested content from a remote origin server, as well as induce more network traffic and server workload.

The popularity of requested contents on the Internet shows a strong localization. In other words, the redirected end-user groups may be highly likely to have totally different interests on the web contents. Thus, the manipulated redirection would cause the previously cached contents to be rapidly expelled and the limited caches at edge server to be frequently updated, consequently resulting in much degraded performance and user experience. Also, the decreased cache hit ratio will significantly increase the bandwidth

[4]Note that the domain sharding would become unnecessary under the adoption of HTTP/2 (SPDY) that supports concurrent requests.

costs of CPs for delivering contents to numerous clients [20]. Finally, the increased back-end connections to origin servers for fetching requested contents will further slow down servers' responsiveness.

4.4 More Serious Threats

We further explore the more serious threats of redirection hijacking for maneuvering the end-user's access in CDNs. Technically, CDNs have natural capability to absorb and diffuse attack traffic with the geographically distributed edge networks, and thus become an ideal infrastructure to integrate the enhanced security mechanisms, where the edge servers can (1) act as reverse proxies to inspect incoming traffic and apply the rules of Web Application Firewalls (WAFs) to filter out malicious traffic and (2) perform the load balancing and DoS protection by diverting users from overwhelmed edge servers via DNS-based dynamic mapping or anycast routing.

Adversaries could exploit redirection hijacking to launch a (parts of) DoS attack by directing the requests from a large number of clients to a single IP address of the victim edge server. The WAFs cannot discard those legitimate traffic from real end users. By selectively injecting the DNS records associated with different popular contents, more clients are connecting to the victim edge server, and then the server has to maintain more back-end connections to different origin servers to fetch the contents. However, the cached contents are quickly being replaced, due to high volume of traffic for massive popular contents. Sooner or later, the victim edge server become overloaded and unresponsive to client requests. More importantly, the load balancing cannot appropriately distribute the traffic since the clients are bypassing the mapping system, and subsequently all the clients that are redirected to those overloaded edge servers will not be able to access the contents or services hosted in CDNs anymore.

Furthermore, the adversaries can leverage the system failure or outage to significantly amplify their attacks. For example, we sent the `ping` probes to monitor the liveness of edge servers for two weeks whose IP addresses have been obtained from our experiments for DNS resolution presented in Section 4.1. We found that 4.5% of IP addresses become unresponsive during the tests, around half of which do not come back online by the end of our experiments. With the easy detection for unresponsive edge servers, the adversaries do not have to perform the actual DoS attack and can simply interrupt end users' accessibility by replaying legitimate mapping records associated with those unresponsive edge servers to resolvers.

5 COUNTERMEASURES

In this section, we discuss the practical factors affecting the vulnerability and the countermeasures for detecting or mitigating the redirection hijacking attacks.

5.1 ECS Considerations

The introduction of EDNS-Client-Subnet provides DNS-based CDNs an attractive scheme to improve the accuracy of their mapping systems and the user-perceived performance for clients using public DNS or the resolvers distant from their locations. As mentioned before, the presence or absence of ECS option does not affect the vulnerability we studied in this paper. The standardized document [35] does not discuss the difficulty of signing the dynamic mapping records. Also, according to the document, the EDNS0 extension does not change the behavior of data authentication, i.e., the ECS data will not be signed by DNSSEC.

On the other hand, ECS indeed provides another attack vector for DNS abuse. For example, the *scope netmask* carried in ECS indicates the specific IP block associated with a reply. An adversary may be able to selectively poison a resolver's cache to only impact a specific IP range [50], via a fraudulent record directing clients to a malicious address. However, such an activity can be detected if the record is signed by DNSSEC (assuming that either ECDSA is used or only a limited number of mapping records exist so that the signatures can be pre-computed). Furthermore, if adversaries exploit redirection hijacking to maneuver the end-user mapping for tussling CDN's performance or interrupting a service, they could arbitrarily designate the ECS data to impact more clients by using a less detailed network prefix.

Countermeasures. As discussed in Section 4.3, the root cause that even the DNSSEC with live-signing is not effective against redirection hijacking, lies in that the resolvers cannot detect a legitimate but replayed mapping that is supposedly used for a different group of clients. Thus, considering the ECS enabled, one potential mitigation is to include the ECS data in DNSSEC when signing the RRsets. With ECDSA, the records generated by the end-user mapping can be dynamically signed on demand. Although signing a dynamic record by ECDSA still suffers from the mapping manipulation, due to the ECS data being signed, the resolvers can validate the integrity of the end-user mapping. In other words, adversaries cannot craft a valid record to manipulate the end-user mapping anymore. This is because the signed ECS can guarantee that the data field (i.e., IP addresses) is assigned to the specified user group (ECS data), which eliminates the risk of obscuring the mapping between end users and edge servers.

Limitations. ECS is suggested to be enabled only when clear advantages can be seen by resolvers [35], e.g., open DNS resolvers or a centralized DNS infrastructure serving clients from a variety of geographically distributed networks. Meanwhile, in current practice, CDN vendors typically enable ECS by whitelisting the resolvers that explicitly support ECS, and vice versa. Thus, as only limited adoption of ECS can be expected, signing the RRsets with ECS just authenticates the records in the resolvers that enable ECS.

5.2 DNSSEC Considerations

The inclusion of ECS extension as additional information when signing a record with DNSSEC provides an effective countermeasure against the record replay in redirection hijacking, but its effectiveness is limited by the deployment of ECS. Inspired by this, we then consider a more general scheme that leverages existing additional data elements in DNSSEC.

Note that adversaries cannot generate a valid signature since they are unable to obtain the private key. Moreover, the replay attack of redirection hijacking can be successful because the validity period of DNSSEC signatures is typically long enough to be reused by adversaries to launch the record injection. However, only using a much shorter validity period is not sufficient since the signature inception and expiration could also be fabricated by adversaries. Consequently, we consider that one possible mitigation is to secure

the validity period by including additional timestamp information when signing a record. Combined with a short validity period in RRSIG (e.g., only slightly longer than the TTL of mapping records), this would significantly increase the difficulty of record injection as the validity period cannot be altered and adversaries only have a short time window to perform the record injection.

Therefore, a straightforward approach is to include the validity period (i.e., signature inception and expiration) when signing a record. However, since the validity period is associated with the RRSIG record rather than the record being signed, it breaks the normal operations of signing a record (but in a harmless manner): the inception and expiration timestamp will be generate first, and then the signature of RRSIG is computed according to both the responded RRset and the validity period associated with the RRSIG record itself. Correspondingly, the resolver's software needs to be modified to include the validity period when computing the message digest. An alternative approach is to define a new extension representing the validity period in the additional section of DNS messages and sign the RRsets, including such extension data.

Note that the mechanisms we discussed here have the similarities to TSIG/SIG(0) [36, 64], which sign complete DNS request/response with timestamps. However, TSIG requires a symmetric key and thus is most commonly used for authorizing dynamic updates and zone transfers. The SIG(0)'s functionality has been fundamentally replaced by DNSSEC. We argue that it may be worth to enhance the operations of DNSSEC to mitigate the threat of replay attacks under the prevalence of dynamic mapping in CDNs.

5.3 CNAME Flattening

One of foundational obstacles for CDN vendors to achieve the integrity of redirection records is the prevalent use of CNAME records, specially the dynamic CNAME mapping and chained CNAME records. A possible solution is to hide the CNAME chain from resolvers and leave the CNAME traversing to CDN's authoritative nameservers, i.e., the *CNAME Flattening* [13].[5]

The CNAME Flattening implemented by CloudFare was originally designed to enable the CNAME at the root domain while complying RFC's DNS specification [60], which requires that there should be no other record types if the type of a record is CNAME. With CNAME flattening, the CDN's authoritative nameserver acts as a resolver by recursively resolving the CNAME chain and finally constructs an `A` record to substitute the original CNAME record.

We therefore suggest that the CNAME flattening should also be leveraged by CDNs for security purposes. That is, instead of iteratively replying with multiple CNAME records, the CDN's authoritative nameserver takes the full responsibility of CNAME resolution, typically within the CDN's mapping infrastructure, and finally returns an `A` record, which can be signed with DNSSEC (live signing). This significantly reduces the computational overhead of signing CNAME records as well as the cost of multiple times of signature validation.

Note that the CNAME flattening is mainly associated with the records for the redirection operated by CDNs. The first level CNAME delegation occurs at the CP's authoritative nameservers, which may be out of control of CDNs. However, the CPs can easily secure the CNAME redirections by enabling (traditional) DNSSEC signatures at their authoritative nameservers, since those records are typically static mappings for domain delegation. Also, when enabling the CNAME flattening in DNS-based CDNs, the CDN's authoritative nameservers may need to employ ECS when retrieving the mapping result as the representation of client's networks.

Overall, the CNAME flattening provides CDN vendors with a potential solution to secure the CNAME records at an acceptable cost by avoiding iterative signature validation for multiple CNAME records, while retaining the flexibility of using a CNAME chain to facilitate the platform management.

5.4 Request Re-Mapping

In addition to performing the request routing via DNS or anycast, CDNs also leverage the high-level re-mapping mechanism to remedy the non-optimal server assignment in some cases. For example, when a request for content objects arrives at an edge server assigned by the mapping system, the edge server first performs an RTT measurement to the client. If the RTT is acceptable, the edge server immediately serves the content to the client based on normal content retrieval strategies; otherwise, the edge server requires the mapping system to reassign an optimal server and direct the client to the different server (e.g., via HTTP status code 3xx for redirection). Due to the extra server selection and redirection operations, the re-mapping introduces additional high latency penalty. Moreover, it is worth to note that, with the wide support of ECS, the accuracy of DNS-based mapping has been significantly improved for those clients impacted by the location discrepancy of LDNSes. That is, the clients are rarely being assigned to a non-optimal edge server. Thus, the request re-mapping is typically only suitable for large file transfer, such as video streaming and software distribution [17, 32].

Nevertheless, CDNs can still enable their Real User Measurement (RUM) system to monitor the performance from a large set of clients and aggregate the monitoring results with geographic locality or client-LDNS pairing to recognize the group of clients affected by anomalous redirections. In general, a more fine-grained performance monitoring and more active request re-mapping could be useful to mitigate serious performance degradation in some cases. However, any high-level re-mapping mechanism still suffers from the threat of nullifying load balancing and DoS mitigation when unresponsive edge servers are exploited in redirection hijacking by adversaries, as discussed in §4.4.

5.5 DNS Encryption and Transport-layer DNS

DNSCurve [15] and DNSCrypt [14] use ECC to encrypt DNS packets. Google Public DNS offers DNS-over-HTTPS [19] interface to enable the DNS resolution over encrypted HTTPS connections. Subsequently, Connection-Oriented DNS (T-DNS) [46, 84] is proposed to fundamentally address the weakness of DNS connectionless transmissions in security and privacy. Unlike DNS-Crypt and DNSCurve leveraging the ECC, T-DNS is established over the existing Transport-Layer Security (TLS) framework and is carefully implemented to make the latency and resource needs induced by

[5] A similar functionality has also been implemented by DNS-hosting providers, such as the ANAME record [16]. Here we focus on the discussion of such a feature provided by CDNs.

T-DNS manageable. Using TLS, the channels between stub and recursive resolvers, as well as between recursive resolvers and authoritative servers, would be protected from eavesdropping and MitM attacks.

It is clear that the encrypted DNS and transport-layer DNS indeed address most security and privacy issues of DNS, including the vulnerability we presented in the paper, because adversaries would be unable to know the content of DNS queries. However, due to high performance penalty and expensive cost for deployment, there is only very limited adoption on the current Internet. Moreover, their negative impacts upon the scalability of DNS, especially at the root and top-level domains, remains unclear.

6 RELATED WORK

Disrupting CDN's server assignment has been recently proposed to circumvent the Internet censorship [45, 85], where arbitrary edge servers, rather than the optimal servers assigned by the CDN's mapping system, are used to bypass the DNS-based/IP-based censorship and obtain the censored content. The focus of such censorship circumvention is to retrieve the censored content from edge servers with acceptable performance. By contrast, we explore the attack scenarios where an end-user's access would be significantly degraded or interrupted, resulting in potential financial losses for both CDN providers and content providers.

DNS and CDN. The discrepancy of location proximity between end-users and their LDNSes has been observed for more than a decade [59, 69]. Pang *et al.* [65] characterized the responsiveness of DNS-based network controls according to the behaviors of end-systems and LDNSes. Huang *et al.* [47] proposed a solution called FQDN extension, where the clients obtain a location-aware cluster identifier and add this identifier to hostnames, to tackle the client-LDNS mismatch problem in Global Traffic Management (GTM). In order to improve the efficiency of content delivery, Krishnamurthy *et al.* [53] proposed a method by which the HTTP interactions are piggybacked on DNS responses. Krishnan *et al.* [54] built a system to diagnose the inflated latencies using active measurements to improve the effectiveness of CDN's indirection and user performance. Scott *et al.* [68] built a tool chain for understanding the web deployment and footprints of CDNs by collecting DNS resolution results and probing the IPv4 address space. In addition, Pearce *et al.* [66] developed a tool to measure and study the global DNS manipulation in Internet censorship.

Ager *et al.* [26] compared the local DNS resolvers against public DNS resolvers (Google Public DNS and OpenDNS) to study the responsiveness and diversity of resolvers. Subsequently, Otto *et al.* [63] examined the performance cost when clients use public DNS services to access CDNs. With the emergence of EDNS-Client-Subnet, Streibelt *et al.* [74] and Calder *et al.* [31] leveraged the ECS with specified client prefixes to infer and profile the large-scale service infrastructure on the Internet such as Google. Kintis *et al.* [50] investigated the potential privacy risk of ECS for surveillance, and revealed a cache poisoning threat for highly selective group of clients.

Cache Poisoning and DNSSEC. Schomp *et al.* [67] assessed the vulnerabilities of diverse record injection attacks, particularly Kaminsky's attack and Bailiwick attack. Duan *et al.* [38] proposed a "Hold-On" period before accepting a reply to mitigate the DNS poisoning by allowing a legitimate reply to also arrive. Lian *et al.* [56] measured the practical impact of DNSSEC deployment and found that DNSSEC-signed domains may create collateral damage in the resolutions of valid domains. Van Rijswijk-Deij *et al.* [76, 77] studied the ECDSA deployment in CloudFlare and the .nl TLD, and examined the computational overhead induced by the validation of ECC-based signatures. Yan *et al.* [83] proposed a revised DNSSEC signature that constructs a hash chain to limit the replay vulnerability windows when the master server has failed. Their study tackles the malicious slave servers and has a different scope than our study.

Recent studies revealed the pervasive mismanagement of DNSSEC. Shulman *et al.* [71] developed a validation engine to identify vulnerable keys in DNSSEC-signed domains. Chung *et al.* [34] performed a longitudinal measurement study into how well DNSSEC's PKI is managed.

Security Issues in CDN. Liang *et al.* [57] studied the practical impact of CDN's HTTPS deployment. Composing HTTPS with CDN introduces the complexity of authentication delegation since CDN cuts the secure communication paths offered by HTTPS. Similarly, Wählisch *et al.* [81] investigated the Resource Public Key Infrastructure (RPKI) deployment on the routing layer and reported that CDNs are the main cause for the insufficiency of RPKI deployment. While the focus of these work is on the vulnerability of CDN's backend, our study explores the issue of the frontend of CDN's service delivery.

Chen *et al.* [33] presented the forwarding-loop attacks, where malicious customers may be capable of creating the forwarding loops inside one CDN or across multiple CDNs to launch potential DoS attacks. The root cause of this threat is that CDNs lack the control over customers' (mis)configurations. Vissers *et al.* [80] studied the "origin-exposing" attacks to identify the address of a service origin and bypass the cloud-based security infrastructure, typically provided by CDNs.

7 CONCLUSION

In this paper, we present a new vulnerability of CDNs, redirection hijacking, which stems from the dynamic characteristics of DNS records used for CDN's request routing. In a redirection hijacking attack, adversaries can easily maneuver CDN's mappings between end users and edge servers by injecting crafted but legitimate DNS records. We reveal that DNSSEC is ineffective to address such a hijacking attack, even with the new ECDSA-based signatures that are capable of achieving live signing for dynamically generated DNS records. This is mainly due to the reusability of signed legitimate records, which can be exploited by adversaries to override CDN's surrogate assignment and redirect client requests to inappropriate edge servers. We assess the magnitude of this vulnerability in the wild by characterizing the operations of the request routing for popular CDN vendors and analyzing the threats via multiple case studies. We quantify the practical impacts of redirection hijacking, especially on performance, and present more serious threats that could nullify CDN's load balancing and DoS protection. Finally, we discuss the countermeasures against redirection hijacking in CDNs from different aspects.

REFERENCES

[1] https://www.cdnplanet.com/blog/which-cdns-support-edns-client-subnet/.
[2] https://www.cdnoverview.com.
[3] https://trends.builtwith.com/cdns.
[4] https://wappalyzer.com/categories/cdn.
[5] https://www.bfk.de/bfk_dnslogger_en.
[6] Akamai, *Inc.* Customized Caching Rules. https://developer.akamai.com/learn/Caching/Customized_Caching_Rules.html.
[7] Akamai, *Inc.* Facts & Figures. www.akamai.com/us/en/about/facts-figures.jsp.
[8] Akamai, *Inc.* Fast DNS. https://www.akamai.com/us/en/solutions/products/cloud-security/fast-dns.jsp.
[9] CDN.net. Why low latency CDN is important for eCommerce stores. https://cdn.net/low-latency-cdn-important-ecommerce-stores/.
[10] Cedexis. https://www.cedexis.com/.
[11] CloudFlare, *Inc.* DNSSEC Complexities and Considerations. https://www.cloudflare.com/dns/dnssec/dnssec-complexities-and-considerations/.
[12] CloudFlare, *Inc.* ECDSA: The missing piece of DNSSEC. https://www.cloudflare.com/dns/dnssec/ecdsa-and-dnssec/.
[13] CloudFlare, *Inc.* Introducing CNAME Flattening: RFC-Compliant CNAMEs at a Domain's Root. https://blog.cloudflare.com/introducing-cname-flattening-rfc-compliant-cnames-at-a-domains-root/.
[14] DNSCrypt. https://dnscrypt.org/.
[15] DNSCurve. https://dnscurve.org/.
[16] DNSMadeEasy. Breakthrough in DNS Records. https://www.dnsmadeeasy.com/services/anamerecords/.
[17] E. Zhang. Intelligent User Mapping in the Cloud. https://blogs.akamai.com/2013/03/intelligent-user-mapping-in-the-cloud.html.
[18] F. Assolini. Massive DNS poisoning attacks in Brazil. https://securelist.com/blog/incidents/31628/massive-dns-poisoning-attacks-in-brazil-31/.
[19] Google Public DNS. DNS-over-HTTPS. https://developers.google.com/speed/public-dns/docs/dns-over-https.
[20] Imperva, *Inc.* The Essential Guide to CDN: CDN Caching. https://www.incapsula.com/cdn-guide/cdn-caching.html.
[21] J. Kirk. Google's Malaysia site latest to be felled in DNS attacks. http://www.pcworld.com/article/2054120/googles-malaysia-site-latest-to-be-felled-in-dns-attacks.html.
[22] J. Spring and L. Metcalf. Probable Cache Poisoning of Mail Handling Domains. https://insights.sei.cmu.edu/cert/2014/09/-probable-cache-poisoning-of-mail-handling-domains.html.
[23] P. Gilmore. Serving at the edge: Good for performance, good for mitigating DDoS. https://blogs.akamai.com/2013/04/serving-at-the-edge-good-for-performance-good-for-mitigating-ddos-part-ii.html.
[24] S. Friedl. An Illustrated Guide to the Kaminsky DNS Vulnerability. http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html.
[25] VirusTotal. https://www.virustotal.com.
[26] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Comparing DNS Resolvers in the Wild. In *ACM IMC*, 2010.
[27] S. M. N. Alam and P. Marbach. Competition and Request Routing Policies in Content Delivery Networks. In *CoRR*, 2006. http://arxiv.org/abs/cs/0608082.
[28] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. *IETF RFC 4033*, 2005.
[29] D. Atkins and R. Austein. Threat Analysis of the Domain Name System (DNS). *IETF RFC 3833*, 2004.
[30] A. Barbir, B. Cain, R. Nair, and O. Spatscheck. Known Content Network (CN) Request-Routing Mechanisms. *IETF RFC 3568*, 2003.
[31] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan. Mapping the Expansion of Google's Serving Infrastructure. In *ACM IMC*, 2013.
[32] F. Chen, R. K. Sitaraman, and M. Torres. End-User Mapping: Next Generation Request Routing for Content Delivery. In *ACM SIGCOMM*, 2015.
[33] J. Chen, J. Jiang, X. Zheng, H. Duan, J. Liang, K. Li, T. Wan, and V. Paxson. Forwarding-Loop Attacks in Content Delivery Networks. In *NDSS*, 2016.
[34] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. A Longitudinal, End-to-End View of the DNSSEC Ecosystem. In *USENIX Security*, 2017.
[35] C. Contavalli, W. van der Gaast, D. Lawrence, and W. Kumari. Client Subnet in DNS Queries. *IETF RFC 7871*, 2016.
[36] D. Eastlake 3rd. DNS Request and Transaction Signatures (SIG(0)s). *IETF RFC 2931*, 2000.
[37] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, and W. Lee. Increased DNS Forgery Resistance Through 0x20-bit Encoding: Security via Leet Queries. In *ACM CCS*, 2008.
[38] H. Duan, N. Weaver, Z. Zhao, M. Hu, J. Liang, J. Jiang, K. Li, and V. Paxsonh. Hold-On: Protecting Against On-Path DNS Poisoning. In *SATIN*, 2012.
[39] R. Gieben and W. Mekking. Authenticated Denial of Existence in the DNS. *IETF RFC 7129*, 2014.
[40] S. Goldberg, M. Naor, D. Papadopoulos, L. Reyzin, S. Vasant, and A. Ziv. NSEC5: Provably Preventing DNSSEC Zone Enumeration. In *NDSS*, 2015.
[41] A. Herzberg and H. Shulman. Security of Patched DNS. In *ESORICS*, 2012.
[42] A. Herzberg and H. Shulman. Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org. In *IEEE CNS*, 2013.
[43] A. Herzberg and H. Shulman. Socket Overloading for Fun and Cache-poisoning. In *ACSAC*, 2013.
[44] P. Hoffman and W. Wijngaards. Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC. *IETF RFC 6605*, 2012.
[45] J. Holowczak and A. Houmansadr. CacheBrowser: Bypassing Chinese Censorship Without Proxies Using Cached Content. In *ACM CCS*, 2015.
[46] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. Specification for DNS over Transport Layer Security (TLS). *IETF RFC 7858*, 2016.
[47] C. Huang, I. Batanov, and J. Li. A Practical Solution to the Client-LDNS Mismatch Problem. *ACM SIGCOMM Comput. Commun. Rev.*, 42(2), Mar. 2012.
[48] A. Hubert and R. van Mook. Measures for Making DNS More Resilient against Forged Answers. *IETF RFC 5452*, 2009.
[49] D. Kaminsky. It's The End Of The Cache As We Know It. *BlackHat*, 2008.
[50] P. Kintis, Y. Nadji, D. Dagon, M. Farrell, and M. Antonakakis. Extended Abstract: Understanding the Privacy Implications of ECS. In *DIMVA*, 2016.
[51] A. Klein, H. Shulman, and M. Waidner. Internet-Wide Study of DNS Cache Injections. In *IEEE INFOCOM*, 2017.
[52] O. Kolkman, W. Mekking, and R. Gieben. DNSSEC Operational Practices, Version 2. *IETF RFC 6781*, 2012.
[53] B. Krishnamurthy, E. Krishnamurthy, R. Liston, and M. Rabinovich. DEW: DNS-Enhanced Web for Faster Content Delivery. In *WWW*, 2003.
[54] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. Moving Beyond End-to-end Path Information to Optimize CDN Performance. In *ACM IMC*, 2009.
[55] B. Laurie, G. Sisson, R. Arends, and D. Blacka. DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. *IETF RFC 5155*, 2008.
[56] W. Lian, E. Rescorla, H. Shacham, and S. Savage. Measuring the Practical Impact of DNSSEC Deployment. In *USENIX Security*, 2013.
[57] J. Liang, J. Jiang, H. Duan, K. Li, T. Wan, and J. Wu. When HTTPS Meets CDN: A Case of Authentication in Delegated Service. In *IEEE Security & Privacy*, 2015.
[58] B. M. Maggs and R. K. Sitaraman. Algorithmic Nuggets in Content Delivery. *ACM SIGCOMM Comput. Commun. Rev.*, 45(3), 2015.
[59] Z. M. Mao, C. D. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A Precise and Efficient Evaluation of the Proximity between Web Clients and their Local DNS Servers. In *USENIX ATC*, 2002.
[60] P. Mockapetris. Domain Names - Implementation and Specification. *IETF RFC 1035*, 1987.
[61] M. K. Mukerjee, I. N. Bozkurt, B. Maggs, S. Seshan, and H. Zhang. The Impact of Brokers on the Future of Content Delivery. In *ACM HotNets*, 2016.
[62] M. K. Mukerjee, I. N. Bozkurt, D. Ray, B. Maggs, S. Seshan, and H. Zhang. Redesigning CDN-Broker Interactions for Improved Content Delivery. In *ACM CoNEXT*, 2017.
[63] J. S. Otto, M. A. Sánchez, J. P. Rula, and F. E. Bustamante. Content Delivery and the Natural Evolution of DNS: Remote DNS Trends, Performance Issues and Alternative Solutions. In *ACM IMC*, 2012.
[64] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington. Secret Key Transaction Authentication for DNS (TSIG). *IETF RFC 2845*, 2000.
[65] J. Pang, A. Akella, A. Shaikh, B. Krishnamurthy, and S. Seshan. On the Responsiveness of DNS-based Network Control. In *ACM IMC*, 2004.
[66] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson. Global Measurement of DNS Manipulation. In *USENIX Security*, 2017.
[67] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman. Assessing DNS Vulnerability to Record Injection. In *PAM*, 2014.
[68] W. Scott, T. Anderson, T. Kohno, and A. Krishnamurthy. Satellite: Joint Analysis of CDNs and Network-Level Interference. In *USENIX ATC*, 2016.
[69] A. Shaikh, R. Tewari, and M. Agrawal. On the Effectiveness of DNS-based Server Selection. In *IEEE INFOCOM*, 2001.
[70] H. Shulman and M. Waidner. Fragmentation Considered Leaking: Port Inference for DNS Poisoning. In *ACNS*, 2014.
[71] H. Shulman and M. Waidner. One Key to Sign Them All Considered Vulnerable: Evaluation of DNSSEC in the Internet. In *NSDI*, 2017.
[72] G. Sisson and B. Laurie. Derivation of DNS Name Predecessor and Successor. *IETF RFC 4471*, 2006.
[73] S. Son and V. Shmatikov. The Hitchhiker's Guide to DNS Cache Poisoning. In *SecureComm*, 2010.
[74] F. Streibelt, J. Böttger, N. Chatzis, G. Smaragdakis, and A. Feldmann. Exploring EDNS-client-subnet Adopters in Your Free Time. In *ACM IMC*, 2013.
[75] S. Triukose, Z. Al-Qudah, and M. Rabinovich. Content Delivery Networks: Protection or Threat? In *ESORICS*, 2009.
[76] R. van Rijswijk-Deij, K. Hageman, A. Sperotto, and A. Pras. The Performance Impact of Elliptic Curve Cryptography on DNSSEC Validation. *IEEE/ACM Trans. Netw.*, Sept. 2016.
[77] R. van Rijswijk-Deij, M. Jonker, and A. Sperotto. On the Adoption of the Elliptic Curve Digital Signature Algorithm (ECDSA) in DNSSEC. In *CNSM*, 2016.

[78] R. van Rijswijk-Deij, A. Sperotto, and A. Pras. DNSSEC and Its Potential for DDoS Attacks: A Comprehensive Measurement Study. In *ACM IMC*, 2014.
[79] R. van Rijswijk-Deij, A. Sperotto, and A. Pras. Making the Case for Elliptic Curves in DNSSEC. *ACM SIGCOMM Comput. Commun. Rev.*, 45(5), Oct. 2015.
[80] T. Vissers, T. Van Goethem, W. Joosen, and N. Nikiforakis. Maneuvering Around Clouds: Bypassing Cloud-based Security Providers. In *ACM CCS*, 2015.
[81] M. Wählisch, R. Schmidt, T. C. Schmidt, O. Maennel, S. Uhlig, and G. Tyson. RiPKI: The Tragic Story of RPKI Deployment in the Web Ecosystem. In *ACM HotNets*, 2015.
[82] S. Weiler and J. Ihren. Minimally Covering NSEC Records and DNSSEC On-line Signing. *IETF RFC 4470*, 2006.
[83] H. Yan, E. Osterweil, J. Hajdu, J. Acres, and D. Massey. Limiting Replay Vulnerabilities in DNSSEC. In *NPSec*, 2008.
[84] L. Zhu, Z. Hu, J. Heidemann, D. Wessels, A. Mankin, and N. Somaiya. Connection-Oriented DNS to Improve Privacy and Security. In *IEEE Security & Privacy*, 2015.
[85] H. Zolfaghari and A. Houmansadr. Practical Censorship Evasion Leveraging Content Delivery Networks. In *ACM CCS*, 2016.