Your Remnant Tells Secret: Residual Resolution in DDoS Protection Services

Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton
University of Delaware
Newark, DE, USA
Email: {linjin, haos, hnw, ccotton}@udel.edu

***Abstract*—The increasing prevalence of Distributed Denial of Service (DDoS) attacks on the Internet has led to the wide adoption of DDoS Protection Service (DPS), which is typically provided by Content Delivery Networks (CDNs) and is integrated with CDN's security extensions. The effectiveness of DPS mainly relies on hiding the IP address of an origin server and rerouting the traffic to the DPS provider's distributed infrastructure, where malicious traffic can be blocked. In this paper, we perform a measurement study on the usage dynamics of DPS customers and reveal a new vulnerability in DPS platforms, called *residual resolution*, by which a DPS provider may leak origin IP addresses when its customers terminate the service or switch to other platforms, resulting in the failure of protection from future DPS providers as adversaries are able to discover the origin IP addresses and launch the DDoS attack directly to the origin servers. We identify that two major DPS/CDN providers, Cloudflare and Incapsula, are vulnerable to such residual resolution exposure, and we then assess the magnitude of the problem in the wild. Finally, we discuss the root causes of residual resolution and the practical countermeasures to address this security vulnerability.**

I. Introduction

Distributed Denial of Service (DDoS) attacks have posed a serious threat to Internet users for decades, and their intensity and prevalence are even still growing. Typically, a DDoS attacker exploits a large number of compromised machines (e.g., botnets) and orchestrates a significant amount of traffic being sent to a victim either directly or indirectly by leveraging the reflectors, resulting in service interruptions on the victim side due to exhausted resources. Nowadays, the volume of the aggregated attack traffic can easily reach hundreds of Gbps [1]. Also, more advanced and stealthy methods [2], [3] have been exploited, making DDoS attacks more powerful and difficult to defend. The recent Dyn DDoS attack [4] leveraged the emerging Internet-of-Things (IoT) botnets [5]–[7] and set a record for the largest DDoS attack (1.2Tbps [8]). This attack took down Dyn's nameservers and thus disconnected Dyn's customers from their naming services, causing numerous popular websites (e.g., Twitter and Netflix) inaccessible and significant financial losses for those services. In addition, the emergence of DDoS-as-a-Service [9] even lowers the technical hurdles and costs to launch a large-scale DDoS attack.

As DDoS attacks have become more powerful, it is very challenging to combat them within traditional on-site DDoS defense systems. In order to survive the battle, web service providers resort to the dedicated DDoS Protection Services (DPS), which are the core function offered by the security features of Content Delivery Networks (CDNs).[1] DPS platforms reroute the traffic to their highly distributed network infrastructures, where the traffic is examined and the malicious is blocked. Also, the high network capacity of DPS enables it to handle high-volume traffic.

The key feature of DPS is to reroute the traffic through the DPS's platform (typically the infrastructure of CDNs) to hide the actual IP address of an origin server. In the case of widely adopted DNS-based request rerouting, it requires the customers to change their DNS configurations to enable DPS as their service front-ends. However, if adversaries are able to acquire the origin IP address by exploiting various attack vectors studied in [10], they can launch the DDoS attack directly against the origin, thereby completely circumventing the traffic rerouting mechanism and bypassing the defense provided by DPS. Therefore, it is crucial for a DPS provider and its customers to keep the origin IP addresses hidden and unpredictable.

In this paper, we conduct a large-scale measurement study to thoroughly investigate the DPS usage dynamics and its security implications. In particular, we first examine the top 1 million websites for their adoption of DPS and usage behaviors. We focus on five DPS usage behaviors, including leave, join, pause, resume, and switch. We identify these usage behaviors and then verify the practical operations of an origin IP address (e.g., whether a website would change the already-exposed origin IP address after joining the DPS protection or resuming the service). Furthermore, we reveal and study a new vulnerability in DPS platforms, called *residual resolution*. It exploits stale DNS records stored in DPS providers to obtain the origin IP addresses. Among the popular DPS providers, we uncover that two major DPS providers (Cloudflare and Incapsula) sometimes do expose the origin IP addresses after customers leave their services or switch to another service provider, i.e., the residual resolution exposure. Unfortunately, this residual resolution exposure may nullify a website's DDoS protection provided by future DPS providers.

[1]Note that DPS is also a core function of the Cloud-Based Security Providers (CBSPs) [10]. However, according to [10] and [11], the border between CBSPs and CDNs has been blurred due to their similar functionalities and shared infrastructures. In this paper, we do not differentiate CBSPs and CDNs, while focusing on DPS provided by mainstream CDN vendors.

In order to evaluate the magnitude of residual resolution, we retrieve the A records of the top 1 million websites from the nameservers of Cloudflare, and we collect and resolve the CNAMEs of Incapsula's customers. We filter out the A records that can be publicly resolved through the name resolution. The rest are hidden records that can only be retrieved from DPS's nameservers. We find 3,504 hidden records from Cloudflare and 42 from Incapsula, and verify that of those, 24.8% and 69% point to the real origins, respectively. Those websites are at the high risk of being DDoS attacked.

The major contributions of our work are summarized as follows:

- We investigate the DPS usage dynamics based on measurement and analyze its security implications.
- We observe and verify a new vulnerability, residual resolution, in two large DPS platforms, and assess the magnitude of residual resolution in the wild.
- We discuss the root causes of residual resolution and provide guidelines for addressing this security vulnerability.

The remainder of this paper is organized as follows. Section II introduces the DDoS protection services and security vulnerabilities of origin exposure. In Section III, we describe the threat model of residual resolution. We present our measurement study to track the DPS usage dynamics and analyze the potential security risks in Section IV, and then we assess the magnitude of residual resolution in the wild in Section V. We discuss the root causes and countermeasures of residual resolution in Section VI. We survey the related work in Section VII, and finally, we conclude the paper in Section VIII.

II. Background

In this section, we first provide background on the DDoS Protection Services (DPS) and request rerouting techniques used by DPS providers. Also, since the CDN's infrastructure is naturally able to absorb and divert the attack traffic and the popular DPS platforms are typically co-hosted with CDNs, we then discuss the overlapping functionality between DPS and CDN providers. Finally, we present the attack vectors studied in a prior study.

A. DDoS Protection Services

1) Service Model: The effectiveness of DPS highly relies on hiding the IP address of an origin server and rerouting the web traffic to the DPS provider's network infrastructure that consists of tens or hundreds of Points-of-Presence (PoPs), at each of which a scrubbing center is deployed. The scrubbing center refers to a cleansing station, which includes a large number of edge servers. It is responsible for cleaning the traffic and blocking the malicious on its way to the origin. The total capacity of such networks can reach several Tbps [12]–[14], which is sufficient to absorb the world's largest DDoS attack.

In order to launch a DDoS attack on a website, adversaries need to first perform the name resolution to obtain the IP address of the target, and then send the malicious traffic to the obtained IP address directly or indirectly via reflectors (e.g., NTP servers or DNS open resolvers). However, when DPS is in effect, adversaries would only obtain an edge server's IP address, and the malicious traffic would be rerouted to the DPS platform. Then, the scrubbing centers start examining the traffic and blocking the malicious.

In general, there are two types of rerouting mechanisms used in DPS: the DNS-based rerouting mechanism [15] and the BGP-based rerouting mechanism [16]. In this study, we focus on the websites adopting the DPS with DNS-based rerouting mechanism, which is currently dominant on the Internet.

2) DNS-Based Rerouting Mechanism: Leveraging different components of the DNS ecosystem, there are various mechanisms used to reroute web traffic through a DPS's platform. The most common DNS-based rerouting techniques include A-based, CNAME-based, and NS-based rerouting.

- *A-Based Rerouting*: The A record maps a hostname to an IP address. When A-based rerouting is used, the DPS provider assigns an IP address to a customer and requires the customer to update its A record to the assigned IP address so that the DNS resolution of its website will return the DPS's IP address rather than its origin IP address.
- *CNAME-Based Rerouting*: The CNAME record provides an alias for a domain name. With CNAME-based rerouting, the DPS provider generates a canonical name for a customer. The customer creates a CNAME record in which the customer domain name points to the canonical name given by the provider. After that, the customer domain name would be resolved to this canonical name, and thus the DPS provider would subsequently take the control of name resolution, and finally return an IP address of its edge server.
- *NS-Based Rerouting (NS Hosting)*: The NS record indicates a nameserver that is responsible for the authority of a domain. When adopting NS-based rerouting, the DPS provider assigns the nameservers to host the customer's DNS records. The customer configures these nameservers as its authoritative nameservers via its domain control panel. Then, the DPS provider's nameservers are in charge of the name resolution of the customer's (sub)domains.

3) DPS on CDN: CDN is a geographically distributed network with a large number of edge servers (a.k.a., surrogates) deployed at different edges of the Internet. CDN is built to lower the web origin's workload burden and shorten the latency of fetching web contents by diverting the web requests to edge servers, instead of origin servers. As such, each edge server acts as a reverse proxy, fetching and caching the web contents, leading to a natural evolution of the CDN platforms to deploy security extensions atop their infrastructures. Nowadays, with the increasing demand of the DPS market, more and more CDN providers offer the built-in DPS features, such as cleansing traffic, in their CDN infrastructures. To this end, our

TABLE I: Attack Vectors of Origin Exposure

Origin Exposure	Threat Descriptions
IP History	Historical DNS record databases may contain possible origin IP addresses.
Subdomains	Subdomains that are not protected by DPS are hosted in the same machine as the origin.
DNS Records	Other records like MX record may still point to the origin.
Temporary Exposure	The domain may be resolved to the origin if the DPS is paused.
SSL Certificates	The subject name in the certificate indicates the domain; request certificates from IP space may reveal the origin IP address.
Sensitive Files	Sensitive files stored in the origin may have the origin IP address.
Origin in Content	The webpage file, such as HTML, may have the origin IP address.
Outbound Connection	The origin IP address is exposed when it actively initiates an outbound connection.

study focuses on the major DPS providers, which are typically the popular CDN providers with security extensions.

B. Origin Exposure

As discussed above, the effectiveness of DPS relies on keeping an origin IP address hidden and unpredictable. If the origin IP address of a server has been exposed, adversaries can bypass the name resolution process and launch the DDoS attack directly on the victim server. Previous research has identified several origin exposure vectors that can be exploited to figure out the origin IP address of a server. Table I presents eight identified attack vectors studied in [10]. In this paper, we reveal a new vulnerability called residual resolution that exploits stale caches in DPS nameservers to obtain an origin IP address.

III. Threat Model

A. Residual Resolution

In order to enable the DPS protection that is typically provided by the security features of CDN providers, a website administrator normally needs to indicate the IP address of its origin server in the DPS's configuration portal. Then, the contents of the website would be served via a large number of edge servers distributed across the Internet. As shown in Figure 1(a), with either the CNAME-based or NS-based rerouting, the nameservers of the DPS provider are responsible for providing the mapping results (i.e., the IP address associated with an assigned edge server) for the name resolution (❶). To this end, the actual IP address of the origin server is invisible from the clients including adversaries, and malicious traffic would be rerouted and absorbed by DPS's infrastructures when under attacks (❷).

However, it is common for website owners to decide to leave an in-progress DPS adoption, or switch to another DPS provider. Unfortunately, in this paper we reveal a new security vulnerability that stems from such DPS dynamics and could leak an origin IP address, resulting in the nullification of future DPS protection against DDoS attacks. Figure 1(b) illustrates how the residual resolution may lead to origin exposure. In order to enable the new DPS protection, the website administrators should change the DNS configuration to delegate their name resolution to the new DPS provider. Technically, after such a delegation happens, the name resolution should be entirely handled by the new DPS provider (DPS/CDN 2 in Figure 1(b)), as well as all traffic heading toward the origin server. However, we uncover that an adversary could obtain the origin IP address by directly issuing the DNS queries to the nameservers of the previous DPS provider (DPS/CDN 1 in Figure 1(b)) (❸). In other words, the previous DPS provider may respond to such requests with the previously recorded origin IP address, posing a high risk of origin exposure, which can be exploited by adversaries to launch a DDoS attack directly toward the exposed origin server (❹).[2]

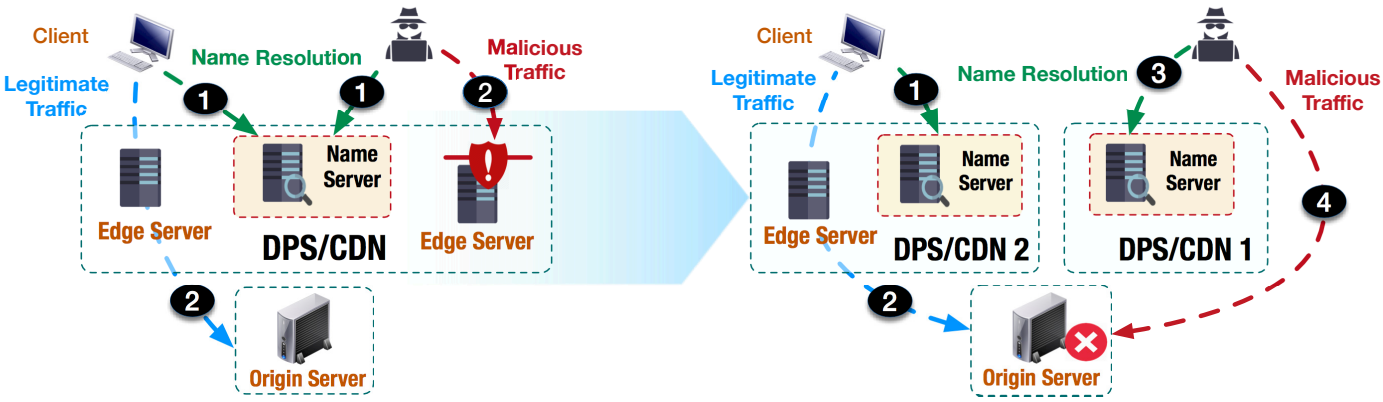
B. Attacker Model

To obtain the "residual" `A` record of a website from its previous DPS provider, an adversary needs to leverage different policies regarding different rerouting mechanisms of DPS's platforms. In particular, if the NS-based rerouting is used, the adversary could acquire the `A` record by directly sending a DNS request to the nameserver of the previous DPS provider. When the CNAME-based rerouting is adopted, the adversary would first need to collect the CNAME record associated with the previous DPS provider. This is because (1) CDNs typically assign a CNAME in a random or unpredictable manner and (2) the CNAME will be updated or deleted if the website terminates its DPS. Once the previous CNAME is collected, the adversary could make many attempts to resolve this CNAME to obtain the `A` record and check if the origin IP address is exposed. Note that if `A`-based rerouting is used, there will be no domain delegation and thus no risk of residual resolution, since the origin IP addresses are not stored in the nameservers of DPS providers.

IV. DPS Usage Dynamics and Security Implications

In this section, we first conduct a large-scale measurement study on DPS usage dynamics for the top 1 million websites within a time period of six weeks. Then, we analyze the

[2]For customers who intentionally leave the DPS protection, the residual resolution (i.e., origin exposure) may be a potential risk only when they rejoin the DPS in the future.



(a) Origin Server under Protection of DPS

(b) Attacks Bypassing DPS by Residual Resolution

Fig. 1: Illustration of the Residual Resolution. Note that (1) the vulnerability is only associated with the previous provider (DPS/CDN 1) and (2) the figure refers to the NS-based or CNAME-based rerouting. With the A-based rerouting, there is no such threat since the (previous) DPS providers are not involved in the process of name resolution.

security implications of the origin exposures based on the DPS usage dynamics.

A. Data Set

The apex domain list used in our experiments is obtained from the Alexa top 1 million list[3]. To retrieve the DNS resolution result of each website, we leverage the presence of the most commonly used portal domain, the `www` subdomain [17]. We select 11 popular DPS providers in our study, which are shown in Table II.

B. DPS Usage Dynamics

1) DNS Record Collector: We study the DPS usage dynamics based on the DNS measurement. In particular, we set a recursive DNS resolver inside Amazon EC2 at the `us-east-1c` zone as our DNS record collector, and send DNS queries for the tested domains to obtain their A, CNAME, and NS records. This measurement repeats every day and lasts for six weeks. Since the TTL of DNS records, especially NS records, may be longer than one day, we purge the DNS cache of the resolver before performing each experiment to ensure that the newly collected records are independent from the previous ones.

2) DPS Adoption: For a given website and its A, CNAME, and NS records acquired from the DNS record collector, we further infer its DPS adoption, including the DPS provider, the DPS status, and the rerouting mechanism. In order to do so, we first define the matching process for A, CNAME, and NS records as follows:

- A-*matching*: We collect the AS numbers of a selected provider and extract its associated IP ranges from the RouteView database[4]. We match the IP addresses from the collected A records with the IP ranges of each provider[5] to determine an "A-matched" DPS provider (see Table II).
- *CNAME-matching*: We collect the unique strings that are used in the second-level domain of the CNAMEs by each provider to determine a "CNAME-matched" DPS provider.
- *NS-matching*: Similar to the CNAME matching, we collect and search the unique strings used in a hostname of nameservers (i.e., NS records) of each provider to determine an "NS-matched" provider.

Determine DPS status. We first define the DPS status ON, OFF, or NONE, as described in Table III. Note that none of the selected providers offer the web hosting services. Therefore, the "A-matching" result indicates if the traffic rerouting is in effect. Specifically, if an A record matches with one DPS provider, it implies edge servers of this DPS provider are receiving the traffic for the origin so that the origin is being protected by this DPS provider, thereby indicating that the DPS status is ON. An "OFF" status means that the domain has been delegated to the DPS provider (CNAME-matching or NS-matching), but the DPS protection is not in effect (non A-matching).[6] Meanwhile, if none of the records is matched, then the DPS status is "NONE", implying that no DPS information has been detected.

Determine DPS providers. Within the same process above, based on the results of our A/CNAME/NS matching, we also determine the adopted DPS provider for each examined website. Figure 2 shows the breakdown of DPS adoption for each DPS provider, with an average per day. Overall, we

[3]http://www.alexa.com/topsites

[4]http://archive.routeviews.org/bgpdata/

[5]We manually collect the IP ranges of the studied DPS providers and make the dataset publicly available at [18].

[6]Note that the edge servers from some providers such as Akamai and CDNetworks may also hold IP addresses from other organizations (e.g., ISPs), which may cause an OFF status to be recorded. We identify that 1.5% of cases in Akamai and CDNetworks fall into this category. We eliminate those cases when we determine the adoption status.

TABLE II: DPS Provider Information [18]

Provider	CNAME Substring	NS Substring	AS Number†	Rerouting Method
Akamai	akamai edgekey edgesuite	akam	32787 12222 20940 16625 35994	A / CNAME
Cloudflare	cloudflare	cloudflare	13335	NS / CNAME
Cloudfront	cloudfront	-	-¶	CNAME
CDN77	cdn77	cdn77	60068	CNAME
CDNetworks	cdnga cdngc cdnetworks	cdnetdns panthercdn	38107 36408	CNAME
DOSarrest	-	-	19324	A
Edgecast	edgecastcdn alphacdn‡	edgecastcdn alphacdn	15133 14210 14153	CNAME
Fastly	fastly	fastly	54113 394192	CNAME
Incapsula	incapdns	incapdns	19551	CNAME
Limelight	llnw lldns	llnw lldns	22822 38622 55429	CNAME
Stackpath	stackpath netdna hwcdn♯	netdna hwcdn	54104 20446	CNAME

†We collect AS numbers from http://www.cidr-report.org/as2.0/autnums.html and only show the major ASes in the table.
¶Cloudfront does not have a dedicated AS number since it builds on the Amazon AWS. Instead, we leverage the data of IP ranges, which can be accessed from http://d7uri8nf7uskq.cloudfront.net/tools/list-cloudfront-ips.
‡Edgecast utilizes a set of substrings named with Greek alphabet in its CNAME/NS records, e.g., alphacdn, betacdn, *etc*.
♯Both MaxCDN (including NetDNA) and Highwinds have been acquired by Stackpath.

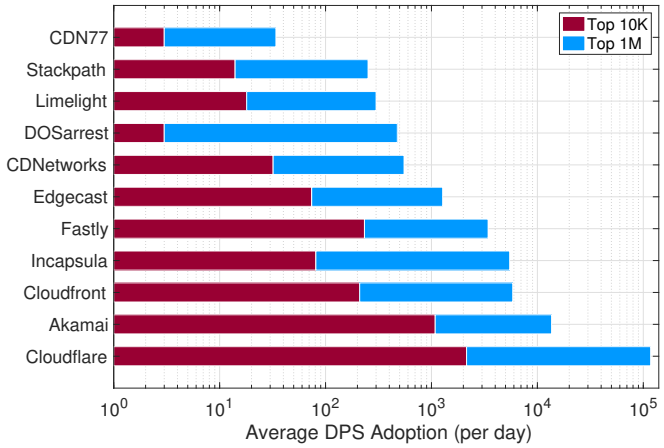


Fig. 2: DPS Adoptions

identify that 14.85% of top 1 million websites employ the DPS services, and Cloudflare dominates the market share.[7] Among the top 10 thousand websites, which usually are the most popular websites, the DPS adoption rate reaches to 38.98%, indicating that the DPS services are more welcome and needed at the popular websites. In addition, we observe an overall increase of 1.17% for DPS adoption in our six-week measurement period.

Rerouting mechanism. We summarize the rerouting mechanisms for each DPS provider in Table II by searching the official technical blogs or documents and crosschecking the

[7]We think this is because the Cloudflare provides the free DPS and DNS service so that many small enterprises can use it.

TABLE III: DPS Status

Status	Explanation
ON	A record points to a DPS's IP (A-matched)
OFF	Domain has been delegated to DPS ("CNAME-matched" with all providers or "NS-matched" with Cloudflare) and A record points to a non-DPS IP (typically the origin IP)
NONE	Domain has not been delegated to DPS and A record points to a non-DPS IP

results of A/NS/CNAME matching. Meanwhile, we label each customer website with its rerouting mechanism according to Table II. For the DPS providers supporting multiple rerouting mechanisms such as Akamai and Cloudflare, we further examine the results of CNAME-matching for customer websites to determine the specific rerouting mechanism applied upon each individual website. In particular, the existence of CNAME-matching indicates that the CNAME-based rerouting is applied for customer websites, while non-existence of CNAME-matching implies that the rerouting mechanism is A-based for Akamai customers and NS-based for Cloudflare customers, respectively.

3) DPS Usage Behavior: We then study the DPS usage behaviors of the examined websites by comparing the DPS adoption data of two consecutive days. Combined with the DPS status presented in Table III, we first define five DPS usage behaviors, as described in Table IV. Since we conduct our experiment daily, we may not identify one pair of two reversed usage behaviors (e.g., LEAVE and JOIN or PAUSE and RESUME) if both just happen in the interval of the two experiments. Also, we assume that when a website joins a

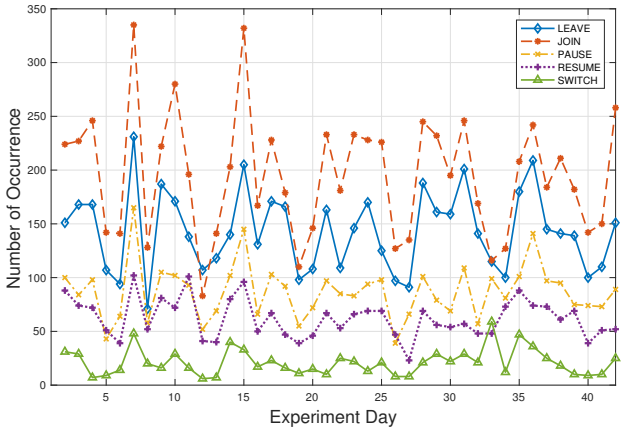


Fig. 3: DPS Behaviors

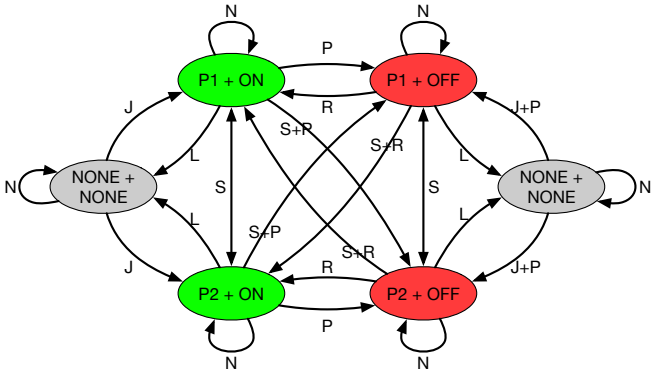


Fig. 4: DPS Finite State Machine

DPS service, the service status is ON by default.

Note that we filter out those websites that use the multiple-CDN platform such as Cedexis[8]. The multiple-CDN service works as front-end redirection and dynamically selects an appropriate CDN for its customers. This dynamic selection feature makes it difficult to identify the accurate usage behaviors.

Figure 3 shows the measurement results for each usage behavior. We can see that the average number of JOIN behaviors (195 per day) is higher than that of LEAVE behaviors (145 per day), indicating the increasing adoption of DPS. Interestingly, the average number of RESUME behaviors (62 per day) is less than that of PAUSE behaviors (87 per day). It implies that the administrators may not have the strong motivation to resume the DPS service after they pause it. One possible reason is that DDoS attacks may not last very long so that temporarily pausing the DPS service is acceptable. In addition, we observe an average of 21 SWITCH behaviors per day, which is the least happened usage behavior.

Moreover, we observe synchronization among behaviors (i.e., different behaviors increase or decrease in the same day). However, we do not find any special event associated with this phenomenon. We infer that such synchronization could be partially aggregated by uneven experiment intervals. We conduct our experiments daily, but the experiment intervals vary from 20 to 30 hours. Indeed, we observe that the longer experiment intervals result in the higher spikes, since the usage behaviors are time-sensitive (i.e., the longer time elapses, the more usage changes happen). To confirm this, we conduct additional experiments with the same interval, and we observe the significantly reduced spikes.

In order to clearly describe the DPS usage dynamics, we design a Finite State Machine (FSM) as shown in Figure 4. The state consists of the DPS provider and its DPS status, where "P1" and "P2" stand for two different DPS providers. The transitions represent DPS usage behaviors. It is worth noting that two usage behaviors could happen in one day. For example, a website joins a DPS provider and pauses its service at the same day, resulting in J + P in the transition.

[8]https://www.cedexis.com/.

TABLE IV: DPS Usage Behavior

Behaviors	Explanation	Status
LEAVE (**L**)	a domain leaves a DPS's platform	ON / OFF → NONE
JOIN (**J**)	a domain joins a DPS's platform	NONE → ON
PAUSE (**P**)	a domain pauses or disables the DPS protection temporarily but does not leave the platform	ON → OFF
RESUME (**R**)	a domain resumes the paused DPS service	OFF → ON
SWITCH (**S**)	a domain switches from one DPS provider to another	-
NULL (**N**)	no action is identified	-

C. Security Implications of DPS Usage Dynamics

The effectiveness of DPS requires that origin IP addresses should remain hidden and unpredictable. However, as we discussed before, the DPS usage dynamics may introduce security problems due to the (mis)configurations or undesired behaviors. In the following, we analyze the security implications of the DPS usage behaviors defined above.

1) PAUSE Behavior: An administrator may temporarily pause the DPS service for various reasons such as maintenance. According to our definition of PAUSE behavior, only when a customer changes its DPS status to OFF, can such an action be identified as a PAUSE behavior, and the OFF status means that the DPS provider exposes origin IP addresses. In our experiment, we find that a PAUSE behavior only happens on the customers of Incapsula and Cloudflare. Therefore, the nameservers of these two providers are configured to return the origin IP addresses in the name resolution when the customers pause the service, which poses the risk of origin exposure.

The longer the pause period is, the higher risk a customer has. In order to analyze the threats posed by PAUSE behav-

iors, we extract the websites that have the PAUSE behaviors ever, and calculate the pause periods that are also the exposure windows. Figure 5 shows the duration of exposure windows caused by PAUSE. The "Overall" result consists of every pause period, including the cases where a website pauses the service at Cloudflare and resumes the service at Incapsula, and vice versa. Less than half of the customers would resume DPS in one day after they pause it. More importantly, around 30% of the pause periods are longer than 5 days, which gives the adversaries enough time to collect the origin IP addresses. The result for Cloudflare or Incapsula only consists of the pause periods when PAUSE and RESUME behaviors happen at the same provider. We can see that the Incapsula's customers have a slightly shorter pause period than the Cloudflare's customers.

2) LEAVE/SWITCH Behavior: Furthermore, the web services may also leave or switch DPS providers for financial or performance considerations. Along with the observation from PAUSE behaviors, we also identify that Cloudflare and Incapsula answer DNS queries with the origin IP addresses in some cases after customers leave their platforms or switch to another provider.[9] In particular, we collect the websites with LEAVE/SWITCH behaviors and then acquire the A records of these websites from their previous DPS providers. For the websites using NS-based rerouting offered by Cloudflare, we directly send the DNS queries to the Cloudflare's nameservers to obtain the A records. For the websites using CNAME-based rerouting in Incapsula, we retrieve the corresponding A records of CNAMEs. Note that if a DPS customer intentionally leaves the DPS platform, the residual resolution from its previous provider may not be a real threat if the customer would never return to a DPS platform again in the future. However, it is still inappropriate for DPS providers to reveal their previous customers' origin addresses.

Residual Resolution Verification. To this end, we conclude that Cloudflare and Incapsula do respond DNS queries with the previously recorded origin IP addresses after their customers are involved with the corresponding DPS usage behaviors, resulting in the vulnerability of residual resolution. To verify the problem of residual resolution in DPS usage dynamics, we sign up the services of Cloudflare and Incapsula, and enable the DPS protection for our own website. We then terminate[10] the DPS service and verify that their nameservers will respond with the origin IP address of our website when we request the A record from their nameservers directly. We believe the existence of residual resolution in Cloudflare and Incapsula is due to the fact that those cached and still valid NS/CNAME records of a terminated website across the Internet are still pointing to the nameservers of Cloudflare/Incapsula, and both

[9]Some customers may not explicitly notify the previous DPS provider of their leaving or switching (e.g., via management portal). If the previous DPS provider is not aware of a customer's leaving, it may not change the configurations for the customer, and hence it will not return the origin IP address.

[10]When a customer terminates its DPS service, it means that the customer explicitly informs the DPS provider of its leaving or switching to another DPS service.

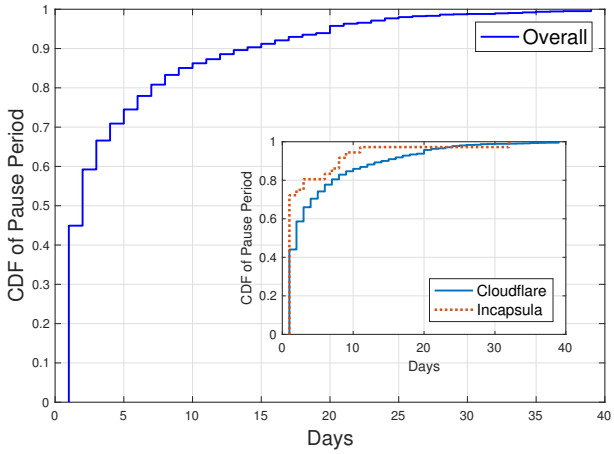


Fig. 5: CDF for the Pause Period

decide to continue answering the DNS queries for this website in order to avoid service disruption at the website.

3) JOIN/RESUME Behavior: In fact, when a website newly enables a DPS or resumes a paused DPS protection, one of the best practices for its administrator is to assign a new IP address to the origin and notify its (new) DPS provider with this address [19], [20], which can significantly reduce the risk of exposing origin addresses. Otherwise, it may leave a backdoor for adversaries to bypass the DPS services by exploiting the residual resolution presented before. Thus, we perform an experiment to explore whether this practice has been widely adopted.

To do so, we focus on the websites with the JOIN and RESUME behaviors. Note that we exclude the SWITCH behavior here because switching to another provider is typically not required to change the origin IP address; however, it does introduce the problem of residual resolution. We examine the unchanged rate of origin IP in the following steps.

- For any website with the JOIN or RESUME behavior, we extract its origin IP address before either of the actions has been taken, and mark it as IP1.
- For the same website, we retrieve the IP addresses responded by the DPS provider after the JOIN or RESUME behavior. Normally they are IP addresses of edge servers in the DPS platform, and we mark them as IP2.
- **HTML Verification.** We obtain the corresponding URL of the landing page of a website and download the HTML file of the landing page by sending an HTTP GET request to each IP2.[11] Then, we send another HTTP GET request to IP1 with the obtained URL, trying to download the HTML file of the landing page. We then verify that if these two HTML files are from the same host by comparing their titles and meta tags. We notice that some

[11]An edge server with IP2 will retrieve the HTML file of the origin and return it to us. Meanwhile, the HTTP response contains the corresponding URL of the landing page.

TABLE V: Origin IP Unchanged Rate

Provider	Join & Resume	IP Unchanged	Percentages
Cloudflare	7,302	4,342	59.5%
Akamai	412	239	58.0%
Cloudfront	443	155	35.0%
Incapsula	492	312	63.4%
Fastly	119	68	57.1%
Edgecast	45	30	66.7%
CDNetworks	46	34	73.9%
DOSarrest	58	24	41.8%
Limelight	6	4	66.7%
Stackpath	40	29	72.5%
CDN77	32	30	93.8%
Total	**8,995**	**5,267**	**58.6%**

attributes in the meta tags are dynamically changed based on different factors (e.g., time and location) of the HTTP requests, and the origin server could be configured to only respond to the requests from the DPS. Therefore, we may miss some exposed origin IP addresses, and the number of the origin IP addresses we can verify are the lower bound of exposed origin IP addresses.

Table V summarizes the results of origin IP unchanged rates. We can see that the traditional CDN providers that do not intentionally highlight their security features, such as CDN77 and CDNetworks, have the highest unchanged rates. The DOSarrest, a security-driven company, has a relatively low percentage. The Amazon's Cloudfront has the lowest percentage (35%) of the unchanged IPs. This is because its customers are mainly from Amazon's cloud platforms where the IP addresses are highly dynamic, especially when users shut down VMs and reboot. In summary, more than half (58.6%) of the DPS customers do not have strong security awareness to change their origin IP addresses after joining and resuming DPS services, resulting in the serious threat of origin exposure.

V. Residual Resolution in the Wild

The residual resolution allows adversaries to acquire the origin IP address of a website from its previous DPS provider, even if the website is under the protection of another DPS provider. Given that 82.6% of customers that use DPS in this study are from Cloudflare and Incapsula, we assess the magnitude of this problem in these two platforms through measurement-based case studies.

A. Case Study: Cloudflare

Cloudflare serves 79% of the customers in our study. It leverages both CNAME-based and NS-based rerouting to deliver its DPS services. We show the breakdown of the customer adoption for Cloudflare in Figure 6. Note that the CNAME-based rerouting is exclusive to those customers with the business or enterprise plans [21], and thus its adoption is significantly less popular than the adoption with NS-based rerouting (10.05% vs. 89.95%). Therefore, we focus on the customers with NS-based rerouting in Cloudflare.

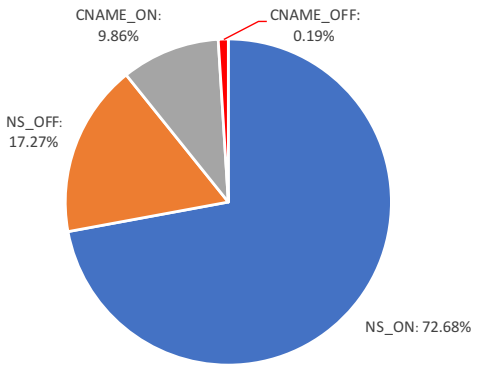


Fig. 6: Cloudflare Average Adoption Breakdown

Fig. 7: Cloudflare PoPs and Vantage Points

1) Cloudflare Nameserver System: Cloudflare builds the nameserver system within its global anycast CDN infrastructure (over 100 PoPs distributed across the world). It stores DNS records in a central database and distributes the records via its anycast-based DNS system [22], [23]. Leveraging the global anycast routing, the DNS requests sent to the same IP address of nameservers will hit different physical machines if the hosts issuing these requests are located at different PoPs. In addition, every nameserver is able to answer queries for all its customers.

As such, in order to reduce the impact of traffic upon our experiments, we set up five geographically distributed vantage points (within the Google Cloud Platform and Amazon EC2) on machines in different regions (Oregon, London, Sydney, Singapore, and Tokyo, as shown in Figure 7) to distribute the total traffic load to five PoPs of Cloudflare. Moreover, we observe that (1) the nameservers used for NS-based rerouting are different from those used for CNAME-based rerouting and (2) all the customers using NS-based rerouting are equipped with the nameservers including a unique string

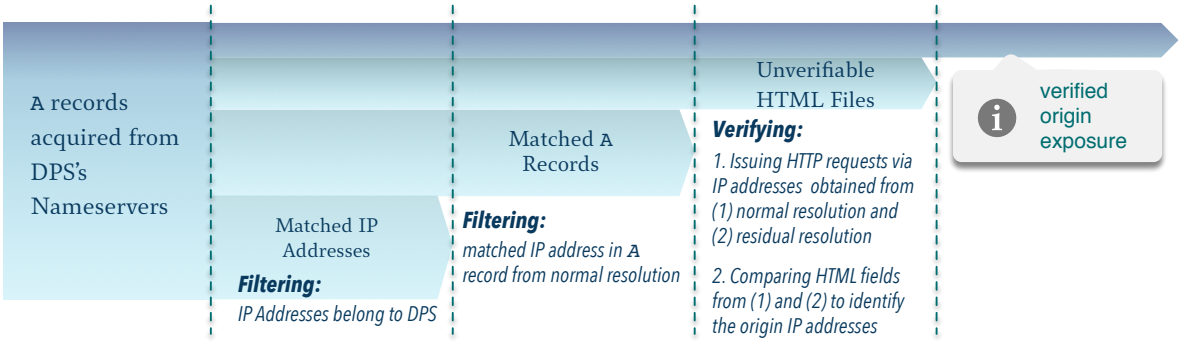


Fig. 8: Filtering Procedure

"ns.cloudflare.com".[12] Consequently, in total we extract 391 Cloudflare nameservers that are exclusively used for the NS-based rerouting customers.

2) Experimental Approach: To investigate the residual resolution problem in Cloudflare, we first retrieve the A records of the top 1 million websites (i.e., the www subdomains) stored in Cloudflare by sending the DNS queries directly to randomly-chosen nameservers acquired from the previous step. The nameserver will respond to a query with the A records of the requested website if it holds the records. Otherwise, it will ignore the query. We perform the experiment once a week for a period of six weeks.

We design a filter-based approach to analyze the A records retrieved from Cloudflare's nameservers and verify the exposed origin IP addresses. Figure 8 depicts the different stages of our detection scheme. Note that the same filtering and verification procedures are also applied to Incapsula.

- **IP-matching Filter.** We filter out the A records with the IP addresses in the IP ranges of the Cloudflare DPS provider, because those websites are under the Cloudflare DPS protection now and there is no residual resolution at this time window. We group the rest of A records after filtering into a set of A_{IP}.
- **A-matching Filter.** Based on A_{IP}, we collect another set of A records by performing the *normal* name resolutions and mark the results as a set of A_{nor}. In particular, for each website whose A record is in A_{IP}, the normal name resolution retrieves the corresponding A records from the website's current authoritative nameserver(s). We refer to the difference between A_{IP} and A_{nor} as a new set of A_{diff}, where
  $A_{diff} = A_{IP} - A_{nor} = \{x: x \in A_{IP} \text{ and } x \notin A_{nor}\}$.
  Note that the A records in set A_{diff} can only be retrieved from the DPS nameservers and are not visible through normal resolutions. We refer to these records as *hidden* records. The hidden records are essentially exposed by the residual resolution.
- **HTML Verification Filter.** The hidden records can be exploited only when the corresponding IP addresses are currently pointing to the origin servers. If the origin IP addresses have already been changed by administrators, the exposure of hidden records is harmless. Therefore, by leveraging the IP addresses from both hidden records and normal resolutions, we conduct the HTML verification as presented in Section IV to uncover the origins exposed by the DPS providers. If a hidden record passes this HTML verification filter, we conclude that this website is vulnerable to the origin exposure caused by the residual resolution.

[12]Those nameservers are named as [girl/boy's name].ns.cloudflare.com.

TABLE VI: Residual Resolution in the Wild

	Hidden Records	Verified Origins	Percentage
Cloudflare			
Week 1	1,449	326	22.5%
Week 2	1,480	365	24.7%
Week 3	1,464	361	24.7%
Week 4	1,893	467	24.7%
Week 5	1,356	315	23.2%
Week 6	1,435	300	20.9%
Total	**3,504**	**868**	**24.8%**
Incapsula			
Total	**42**	**29**	**69.0%**

3) Experimental Results: As shown in Table VI, for each experiment we can identify 1,512 hidden records on average and verify that approximately 24% of them are pointing to the origin addresses. In total, we identify that 24.8% (868 out of 3,504) of the hidden records suffer from origin exposure caused by the residual resolution. Note that there are some overlapping hidden records as well as exposed origins at each week's experiment, resulting in that the total numbers do not match with the sum of numbers from each week.

We then explore the duration of an exposed origin. The results are illustrated in Figure 9, where the repeatedly appeared instances of exposed origins are labeled with a same pattern in different experiments. For each experiment from week 2 to week 6, on average we find 114 newly exposed origins caused

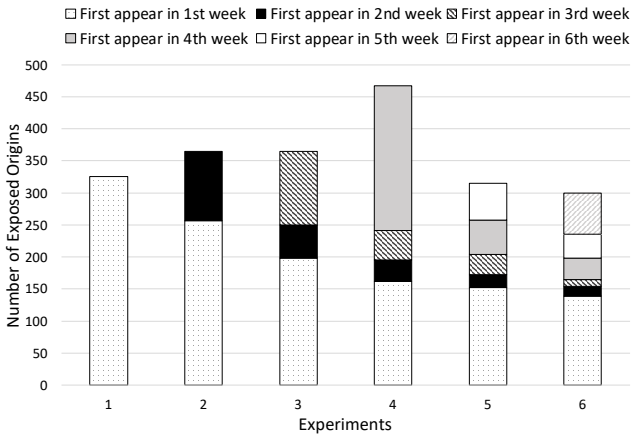


Fig. 9: Exposure Observations

by residual resolution in Cloudflare. Meanwhile, we observe that 139 origins are always exposed during the entire period of our experiments, implying that their exposure durations are at least 5 weeks. Moreover, there are 388 exposed origins, whose starting and ending exposure times are both within our experiments. (i.e., we observe both their exposure appearance and disappearance), indicating that either the administrators have changed their origin IP addresses or Cloudflare has purged their records.

In order to figure out how long Cloudflare may purge the stale DNS records, we sign up its free DPS service with our own website and terminate the service at the same day. We then find that our `A` record is purged at the 4th week after the day of termination. We conduct the same trial for three times and observe the same result, while the time interval between any two trials is 3 weeks. Thus, we speculate that the long exposure duration (more than 3 weeks) in Figure 9 may be due to the adoption of different DPS service plans.

B. Case Study: Incapsula

Incapsula serves 3.7% of customers in our study. It only employs CNAME-based rerouting for its DPS.

1) Experimental Approach: In order to study the residual resolution problem in Incapsula, we extract the CNAMEs of its customers from the DNS records we collected in Section IV. We then keep tracking of the `A` records associated with those CNAMEs for three weeks. We identify the exposed origin IP addresses by using the same filtering and verification procedures as shown in Figure 8.

2) Experimental Results: The results are shown in Table VI. We identify 42 hidden records in total, and verify that 69% of them are the origin IP addresses. Although Incapsula has a relatively smaller number of hidden records as well as the verified origins (mostly due to the limited set of customers we identified), the high percentage of verified origins indicates that the problem is still serious.

C. Limitations

While our approach is able to uncover the exposed origin IP addresses, we cannot verify whether they have already been exploited. Moreover, our study currently only covers the websites with `www` subdomain. The residual resolution problem could be universal across any subdomain that adopts the DPS service offered by Cloudflare or Incapsula. Also, as we mentioned before, our HTML verification provides the lower bound number of the verified origins. Therefore, the exposed origins could be much more prevalent than the results we presented.

VI. Discussion

A. Causes of Residual Resolution

Since the residual resolution problem is mainly due to the intentional configuration by the DPS providers, we attempt to speculate the reasons behind such configuration decisions. We believe that the allowance of residual resolution is mainly under the consideration of service continuity. More specifically, when a customer adopts a DPS service, it requires the customer to change the DNS setting so that the DPS's nameservers are involved in the name resolution and responsible for providing `A` records. However, after a customer terminates the service and changes its DNS setting again, the NS/CNAME records of the customer website cached in DNS resolvers across the Internet may still point to previous DPS's nameservers, especially given the fact that the TTLs of NS records are relatively long [24], [25]. Those stale NS/CNAME records[13] would direct DNS queries to the previous DPS provider, which is no longer able to provide the legitimate address of its edge server since the service is terminated. To this end, the DPS providers (i.e., Cloudflare and Incapsula) respond to those queries with the origin IP addresses to ensure the continuous access to the web services. Unfortunately, as a side effect of such a configuration, a backdoor is left open, allowing adversaries to figure out the origin IP addresses, especially of those websites that are being protected by other DPS's platforms.

B. Countermeasures

1) Actions from DPS providers: As discussed above, the problem of residual resolution stems from the (mis)configuration of DPS's nameserver systems. Therefore, it can be completely eliminated if DPS providers do not respond to DNS queries with origin IP addresses. On the other hand, if the providers would like to minimize the impact of service discontinuity while avoiding the risk of residual resolution, they should keep tracking of the `A` records of a customer who has recently terminated its service. As such, if the current IP address of the customer acquired from a normal DNS resolution does not match the IP address stored in the DPS's nameserver system, which implies that the customer's origin is being protected by another DPS or the customer has

[13]We mainly discuss NS/CNAME records since `A` records offered by DPS providers are typically with much shorter TTL values.

changed its origin IP address, the DPS provider should stop responding to the DNS queries on this customer.

2) Actions from DPS customers: The successful exploitation of residual resolution depends on (1) the DPS providers would respond to DNS queries with the previously stored `A` records and (2) an origin IP address is left unchanged after a customer adopts a different DPS provider. Thus, the customers may intentionally leave a fake `A` record before they terminate the DPS service so that the DPS provider will not be able to reveal the actual origin IP addresses after service termination. In addition, customers can completely circumvent residual resolution by changing their origin IP addresses after adopting another DPS, which is also one of the best practice recommended by the majority of DPS providers. More importantly, it not only eliminates the residual resolution attack vector, but also mitigates the major attack vectors associated with the origin exposure attack.

VII. Related Work

A. DDoS Attacks and Protections

DDoS attacks and protections have received broad attention in both academia and industry for decades, and many attack variants have been developed. In recent studies, link-flooding attacks [26] have been exploited to cut off network connections by flooding the network links between adversaries [27] or concentrating the attack flows on a small set of carefully chosen links [2]. Also, modern DDoS attacks have been widely involved with the amplification vulnerabilities in network protocols. Rossow [1] analyzed the capability of mounting potential amplification DDoS attacks for 14 UDP-based protocols. Kührer *et al.* [28] performed Internet-wide scans to identify potential amplifiers for seven network protocols and observed a significant decrease in the number of NTP amplifiers after a large-scale security notification campaign.

To thwart DDoS attacks, different approaches have been proposed. Wang *et al.* [29] presented a filtering technique to weed out spoofed IP packets at a victim server by checking the number of hops an IP packet takes to reach the victim (i.e., the IP-to-Hop-Count mapping). Krupp *et al.* [30] proposed a novel scheme to trace back the sources of amplification DDoS attacks by leveraging honeypots as potential amplifiers. Smith *et al.* [31] presented a defense system to mitigate the transit-link DDoS attacks, regardless of the amount of attack traffic, by leveraging BGP advertisements and allowing an AS to achieve the traffic isolation from a critical upstream AS.

Vissers *et al.* [10] systematically studied and discussed eight origin exposure attack vectors to bypass the emerging CBSP (e.g., DPS provider) and analyzed the global risk of the origin exposure. They found that more than 70% of the evaluated websites are vulnerable to at least one of the attack vectors. Recently, Jonker *et al.* [11] performed a large-scale measurement on the status of DPS adoption and found that the DPS adoption had grown by a factor of 1.24 during their measurement period of 1.5 years. We improve their method for determining the DPS adoption by considering the combination of `A`/CNAME/NS matching results. Jonker *et al.* [32] then further developed a framework to comprehensively characterize the DoS ecosystem, including attack events, attack targets, and DPS services. More importantly, they found that the attack intensity is the major factor that contributes to the DPS migration, while the repeated attacks and attack duration do not strongly correlate with the DPS migration.

B. CDN Security

Liang *et al.* [33] investigated the deployment issue of HTTPS in CDN infrastructures and proposed a DANE-based solution, which stores a certificate in each DNS record and leverages DNSSEC to ensure the integrity of the certificate for front-end authentication. Chen *et al.* [34] presented four different types of the forward-loop attacks in CDNs, where adversaries can configure the forwarding paths to create loops inside one CDN or across multiple CDNs to massively consume CDN bandwidth resources. Gilad *et al.* [35] developed a software-based DDoS defense system by creating an on-demand CDN system that manages cloud resources in an elastic way to provide equivalent functions offered by a typical CDN provider with a relatively low cost. To launch potential DDoS attacks, Triukose *et al.* [36] presented a method to force the requests penetrating CDN caches and reaching origin servers.

C. DNS Security and Stale Information Exploitation

The integrity of DNS is critical to the Internet ecosystem and its exploitation has been widely studied (e.g., parked domains [37], [38], domain squatting [39], [40], and domain shadowing [41]). Recently, stale records has been extensively exploited to manipulate the integrity of the DNS ecosystem. Lever *et al.* [42] studied the residual trust of expired domains, which have been re-registered and exploited by adversaries. Lauinger *et al.* [43] analyzed the post-expiration domain ownership changes, and found that the expired domains are pre-released during the auto-renewal period. Liu *et al.* [44] demonstrated that adversaries can exploit unsafe dangling DNS records to hijack domains by harnessing three different attack vectors. Our study reveals a new vulnerability of residual resolution in DPS, where stale DNS records in the nameservers of DPS could be exploited for origin exposure.

VIII. Conclusion

The growth of Distributed Denial of Service (DDoS) attacks in both power and prevalence has led to the wide and increasing adoption of DDoS Protection Services (DPS) on the Internet. In this paper, we investigated the DPS usage dynamics and uncovered a new vulnerability in DPS platforms, called residual resolution, which could be abused by adversaries to bypass the DPS protection. In particular, we conducted a large-scale measurement study on the DPS usage dynamics and its security implications. We found that 58.6% of websites did not change their origin IP addresses after joining or resuming a DPS service, which poses the security risk of exposing origin addresses and nullifying DPS. In addition, we analyzed the pause periods in DPS, and found

that approximate 30% of the pause periods are longer than 5 days, which is a rather long time for adversaries to collect the origin IP addresses. Meanwhile, the residual resolution exposure exists when a customer leaves a DPS provider or switches to another DPS provider. We observed and verified that two major DPS providers, Cloudflare and Incapsula, suffer from residual resolution exposure. We evaluated the magnitude of this problem in the wild, and identified 897 exposed origin IP addresses in Cloudflare and Incapsula. Finally, we discussed the root causes of this security vulnerability and provided guidelines for effective countermeasures.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their insightful and valuable comments, which help us improve the quality of this work. This work was partially supported by NSF grant CNS-1618117 and ONR grant N00014-17-1-2485.

REFERENCES

[1] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Network and Distributed System Security Symposium (NDSS)*, 2014.

[2] M. S. Kang, S. B. Lee, and V. D. Gligor, "The Crossfire Attack," in *IEEE Symposium on Security and Privacy (S&P)*, 2013.

[3] R. Rasti, M. Murthy, N. Weaver, and V. Paxson, "Temporal Lensing and its Application in Pulsing Denial-of-Service Attacks," in *IEEE Symposium on Security and Privacy (S&P)*, 2015.

[4] S. Hilton, "Dyn Analysis Summary Of Friday October 21 Attack," https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/.

[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *USENIX Security Symposium*, 2017.

[6] B. Herzberg, D. Bekerman, and I. Zeifman, "Breaking Down Mirai: An IoT DDoS Botnet Analysis," https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html.

[7] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," in *IEEE Computer 50(7): 80-84*, 2017.

[8] R. Graham, "Mirai and IoT Botnet Analysis," in *RSA Conference*, 2017.

[9] M. Karami and D. McCoy, "Understanding the Emerging Threat of DDoS-As-a-Service," in *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2013.

[10] T. Vissers, T. van Goethem, W. Joosen, and N. Nikiforakis, "Maneuvering Around Clouds: Bypassing Cloud-based Security Providers," in *ACM Conference on Computer Communication Security (CCS)*, 2015.

[11] M. Jonker, A. Sperotto, R. van Rijswijk-Deij, R. Sadre, and A. Pras, "Measuring the Adoption of DDoS Protection Services," in *ACM Internet Measurement Conference (IMC)*, 2016.

[12] CloudFlare, "Advanced DDoS Protection and Mitigation," https://www.cloudflare.com/ddos/.

[13] Incapsula, "Imperva. Global Network Map," https://www.incapsula.com/incapsula-global-network-map.html.

[14] Fastly, "A new architecture for the modern internet," https://www.fastly.com/network-map/.

[15] A. Barbir, B. Cain, R. Nair, and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms," *IETF RFC 3568*, 2003.

[16] Incapsula, "Infrastructure DDoS Protection," https://www.incapsula.com/infrastructure-ddos-protection-services.html.

[17] Bitquark, "The most popular subdomains on the Internet," https://bitquark.co.uk/blog/2016/02/29/the_most_popular_subdomains_on_the_internet.

[18] "DPS/CDN Information," https://doi.org/10.5281/zenodo.1216249.

[19] N. Sullivan, "DDoS Prevention: Protecting The Origin," https://blog.cloudflare.com/ddos-prevention-protecting-the-origin/.

[20] I. Zeifman, "How to Prevent Origin Exposing Attacks (Cloud-Piercer Study)," https://www.incapsula.com/blog/cloudpiercer-origin-ddos-attack.html.

[21] Cloudflare, "How do I do CNAME setup?" https://support.cloudflare.com/hc/en-us/articles/200168706-How-do-I-do-CNAME-setup, 2017.

[22] T. Arnfeld, "How We Made Our DNS Stack 3x Faster," https://blog.cloudflare.com/how-we-made-our-dns-stack-3x-faster/, 2017.

[23] Cloudflare, "Fast, Powerful, and Secure DNS," https://www.cloudflare.com/dns/.

[24] H. Gao, V. Yegneswaran, Y. Chen, P. Porras, S. Ghosh, J. Jiang, and H. Duan, "An Empirical Reexamination of Global DNS Behavior," in *ACM SIGCOMM Conference*, 2013.

[25] S. Hao, H. Wang, A. Stavrou, and E. Smirni, "On the DNS Deployment of Modern Web Services," in *IEEE International Conference on Network Protocols (ICNP)*, 2015.

[26] D. Gkounis, V. Kotronis, C. Liaskos, and X. Dimitropoulos, "On the Interplay of Link-Flooding Attacks and Traffic Engineering," in *ACM SIGCOMM Computer Communication Review (CCR)*, 2016.

[27] A. Studer and A. Perrig, "The Coremelt Attack," in *European Symposium on Research in Computer Security (ESORICS)*, 2009.

[28] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks," in *USENIX Security Symposium*, 2014.

[29] H. Wang, C. Jin, and K. G. Shin, "Defense Against Spoofed IP Traffic Using Hop-count Filtering," *IEEE/ACM Transactions on Networking*, 2007.

[30] J. Krupp, M. Backes, and C. Rossow, "Identifying the Scan and Attack Infrastructures Behind Amplification DDoS Attacks," in *ACM Conference on Computer Communication Security (CCS)*, 2016.

[31] J. M. Smith and M. Schuchard, "Routing Around Congestion: Defeating DDoS Attacks and Adverse Network Conditions via Reactive BGP Routing," in *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[32] M. Jonker, A. King, J. Krupp, C. Rossow, A. Sperotto, and A. Dainotti, "Millions of Targets Under Attack: a Macroscopic Characterization of the DoS Ecosystem," in *ACM Internet Measurement Conference (IMC)*, 2017.

[33] 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 Symposium on Security and Privacy (S&P)*, 2014.

[34] 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 *Network and Distributed System Security Symposium (NDSS)*, 2016.

[35] Y. Gilad, A. Herzberg, M. Sudkovitch, and M. Goberman, "CDN-on-Demand: An Affordable DDoS Defense via Untrusted Clouds," in *Network and Distributed System Security Symposium (NDSS)*, 2016.

[36] S. Triukose, Z. Al-Qudah, and M. Rabinovich, "Content Delivery Networks: Protection or Threat?" in *European Symposium on Research in Computer Security (ESORICS)*, 2009.

[37] S. Alrwais, K. Yuan, E. Alowaisheq, Z. Li, and X. Wang, "Understanding the Dark Side of Domain Parking," in *USENIX Security Symposium*, 2014.

[38] T. Vissers, W. Joosen, and N. Nikiforakis, "Parking Sensors: Analyzing and Detecting Parked Domains," in *Network and Distributed System Security Symposium (NDSS)*, 2015.

[39] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Felegyhazi, and C. Kanich, "The Long "Taile" of Typosquatting Domain Names," in *USENIX Security Symposium*, 2014.

[40] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis, "Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse," in *ACM Conference on Computer Communication Security (CCS)*, 2017.

[41] D. Liu, Z. Li, K. Du, H. Wang, B. Liu, and H. Duan, "Don't Let One Rotten Apple Spoil the Whole Barrel: Towards Automated Detection of Shadowed Domains," in *ACM Conference on Computer Communication Security (CCS)*, 2017.

[42] C. Lever, R. Walls, Y. Nadji, D. Dagon, P. McDaniel, and M. Antonakakis, "Domain-Z: 28 Registrations Later Measuring the Exploitation of Residual Trust in Domains," in *IEEE Symposium on Security and Privacy (S&P)*, 2016.

[43] T. Lauinger, A. Chaabane, A. S. Buyukkayhan, K. Onarlioglu, and W. Robertson, "Game of Registrars: An Empirical Analysis of Post-Expiration Domain Name Takeovers," in *USENIX Security Symposium*, 2017.

[44] D. Liu, S. Hao, and H. Wang, "All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records," in *ACM Conference on Computer Communication Security (CCS)*, 2016.