USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Alias Equals Zone? Large-Scale and Stealthy Takeover of Domain Hosting Service via CNAME-Following Cross-Domain Verification

Ruixuan Li, *Tsinghua University;* Xingyu Zhao, *Zhejiang Gongshang University;* Yunyi Zhang and Baojun Liu, *Tsinghua University;* Jun Shao, *Zhejiang Gongshang University and Zhejiang Key Laboratory of Big Data and Future E-Commerce Technology*

https://www.usenix.org/conference/usenixsecurity26/presentation/li-ruixuan

This paper is included in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the 35th USENIX Security Symposium is sponsored by

جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

Alias Equals Zone? Large-Scale and Stealthy Takeover of Domain Hosting Service via CNAME-Following Cross-Domain Verification

Ruixuan Li[1] Xingyu Zhao[2] Yunyi Zhang[1] Baojun Liu[1] Jun Shao[2,3]

[1]*Tsinghua University,* [2]*Zhejiang Gongshang University,*
[3]*Zhejiang Key Laboratory of Big Data and Future E-Commerce Technology*

Abstract

CNAME records define alias relationships between domains and are widely used for service hosting and load balancing. We find that popular domain hosting providers misinterpret CNAME semantics during domain ownership verification. They accept DNS records after CNAME redirection as valid challenge tokens for alias domains, even though these domains do not configure any tokens. Based on this flaw, we propose ALIASLEAP, a novel domain takeover attack that enables hijacking hosting services of alias domains in CNAME chains. ALIASLEAP poses a serious threat in the real world: we identify four email and seven web hosting providers that are vulnerable, affecting over two million domains, including 200K in the Tranco Top 1M domain list. ALIASLEAP is highly stealthy because vulnerable CNAME chains are typically legitimate and long-lived: about half persist for more than 12 months, and up to 19,819 domains have been exposed for over 10 years. We propose mitigation strategies and responsibly disclose ALIASLEAP to 11 affected hosting providers, receiving confirmations from eight of them. We call on the Internet community to revisit the usage practices and capability boundaries of CNAME records.

1 Introduction

Domain name system (DNS) is a cornerstone of the Internet infrastructure, underpinning the operation of many higher-layer applications. Among the various DNS resource records, canonical name (CNAME) plays a critical role in delegation and indirection [33]. When a domain (the *alias*) has a CNAME record, DNS resolution is redirected to the corresponding *canonical* name, and the DNS resolver continues by retrieving the canonical name's records. With its convenient name-mapping functionality, CNAME is widely used for service migration, load balancing, and integration with third-party services. According to our measurements, at least 21,660 second-level domains (SLDs) and 8,268,312 subdomains in the Tranco top 1M domain list [43] have CNAME records configured.

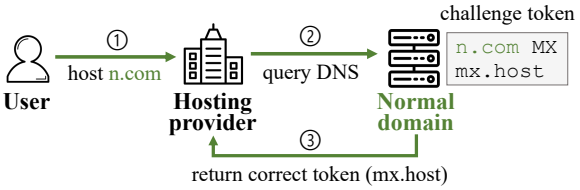

Figure 1: Theoretical domain ownership verification process.

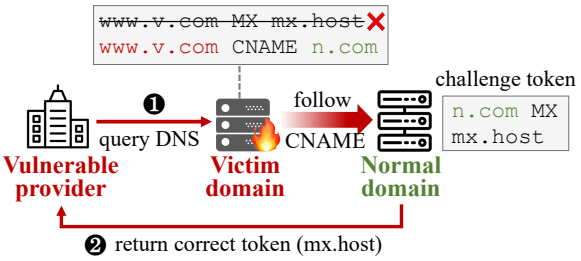

Figure 2: Flawed domain ownership verification logic.

Misunderstanding of CNAME. In domain hosting scenarios, we observe that many providers' domain ownership verification (DOV) misinterprets CNAME, granting the domain alias mechanism privileges equivalent to domain control.

Figure 1 illustrates the theoretical DOV process in an email hosting scenario. Providers require users to configure specific DNS records (i.e., *challenge tokens*) for a domain (`n.com`). Then, the provider queries the MX records of `n.com` and activates the email hosting service upon obtaining the correct token (`mx.host`). However, when CNAME records are introduced into this process, the DOV logic becomes seriously flawed. Specifically, if an alias domain can resolve to valid challenge tokens belonging to the canonical name via a CNAME record, the provider considers the domain as verified. Crucially, no challenge tokens are actually configured in the alias domain's zone.

New threat model. This paper introduces a novel domain hosting service takeover attack, termed ALIASLEAP, which

exploits flawed DOV logic that follows CNAME records to perform cross-domain verification, as shown in Figure 2. We assume that a normal domain (`n.com`) is configured with valid challenge tokens and can be properly hosted at the provider. Victim domains (`www.v.com`) are alias domains that point to `n.com` through a CNAME chain. When an attacker hosts the victim domain, the provider follows the CNAME chain to perform DOV. Upon detecting the expected DNS records (`mx.host`) at the end of the resolution process, the provider activates hosting services for `www.v.com`. After that, attackers can launch a variety of attacks, including deploying phishing websites, sending spoofed emails, and obtaining valid TLS certificates for victim domains, etc.

Using controlled domains, we conduct end-to-end experiments across popular hosting providers to demonstrate the feasibility of AliasLeap. Then, we leverage passive DNS data to collect potentially affected domains at scale. Finally, we non-intrusively detect vulnerable domains by exploiting providers' distinctive error responses for unhosted domains.

Practical risk assessment. AliasLeap poses a significant threat in the wild. We find that four email hosting providers and seven web hosting providers are vulnerable to AliasLeap (e.g., Namecheap, GitHub). The subdomain and SLD can be victims of AliasLeap. Furthermore, we identify over two million domains affected by AliasLeap, the vast majority of which are subdomains. Within the Tranco Top 1M domain list, the email services of 132,057 subdomains and the web services of 31,055 subdomains are vulnerable, including those under `cloudflare.net` and `whecloud.com`. Furthermore, 5,067 government domains and 87,560 educational domains are affected, including institutions such as the State of New Hampshire and the University of Toronto.

The CNAME chains of vulnerable domains are usually configured for legitimate services, making AliasLeap highly stealthy. The most common vulnerable CNAME chains occur between domains within the same SLD, for example, when a *www* subdomain points to its SLD. Additionally, service management, company brand change, and corporate acquisitions are important sources of vulnerable CNAME chains. The DNS configurations exploited by AliasLeap are generally stable, providing attackers with ample time to conduct attacks. We find that over 99% of vulnerable CNAME chains persist for more than one month, about half last over 12 months, and as many as 19,819 domains have been exposed to the AliasLeap threat for more than ten years.

Mitigation and disclosure. Domain hosting providers can mitigate AliasLeap by adopting DOV mechanisms that rely on randomized challenge tokens or by prohibiting cross-domain verification that follows CNAME records. We have responsibly disclosed AliasLeap to 11 affected providers, obtaining confirmations from eight. We received a total bonus of $1,100, and 10 CNVD/CNNVD vulnerability numbers.

Contributions. The contributions of this paper are as follows:

- *New threat model*. We introduce a new domain hosting service takeover attack, AliasLeap, which exploits providers' CNAME-following cross-domain ownership verification.
- *Large-scale risk assessment*. We implement a non-intrusive detection system for AliasLeap, identifying 11 providers and more than two million domains affected by the attack.
- *Mitigation and disclosure*. We propose effective mitigation strategies and responsibly disclose AliasLeap to 11 providers, receiving confirmations from eight of them.

2 Background and Preliminaries

In this section, we first introduce the function of canonical name (CNAME) records and their application scenarios. We then measure CNAME records to understand the real-world CNAME configuration. Finally, we describe common domain ownership verification (DOV) mechanisms.

2.1 CNAME Record

The CNAME record establishes an alias relationship between two domains [33]. As illustrated in Figure 3, a CNAME record maps one domain (e.g., `alias.foo.com`), referred to as the *alias*, to another domain (e.g., `canonical.foo.net`), known as the *canonical* name. When a DNS resolver encounters a CNAME record, it replaces the queried name with the canonical name and continues resolution until it reaches a DNS record type that directly maps to an IP address or service endpoint (e.g., MX records).

Importantly, CNAME records establish equivalence between two domains only at the DNS resolution layer and do not confer control over the domain zone. The management of a specific namespace is reflected in the domain zone and its corresponding authoritative name servers. Furthermore, DNS standards prohibit the coexistence of a CNAME record with other record types at the same time [33]. This restriction renders CNAME records generally defined as unsuitable for second-level domains (SLDs), which typically require multiple essential DNS records (e.g., NS records).

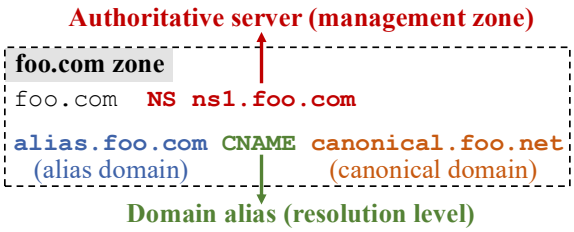

Figure 3: Definition and function of the CNAME record.

CNAME records are widely used to support various operational scenarios. A common use case is domain service hosting, where a domain delegates the implementation and management of Internet services to third-party providers. This

delegation enables service providers to manage backend infrastructure transparently while allowing domain owners to retain control over their domains. For instance, GitHub requires domains to configure CNAME records pointing to its subdomains (e.g., `user.github.io`) [26], where `user` denotes the account name. However, such CNAME-based hosting introduces the risk of dangling DNS records [46], which arise when users fail to update their DNS configurations after releasing services. For example, an attacker can register the `user` account on GitHub and thereby take over the web page of `user.github.io`.

2.2 CNAME Chains in the Wild

We conduct a large-scale measurement of domain CNAME chains to investigate the prevalence and configuration patterns of CNAME records in the wild. First, we downloaded the Tranco Top 1M domain list on December 1, 2025 [43]. Using passive DNS (PDNS) data from 114DNS [16], we then collected subdomains under the SLDs of these popular domains. To limit excessive data from highly popular domains, we restrict the collection to at most 10,000 subdomains per SLD. Next, we queried the CNAME chains for the popular domains and their subdomains through PDNS data. The query window spans from January 1, 2024, to December 1, 2025, and the maximum CNAME chain depth is set to three. To mitigate the impact of transient misconfigurations, we filter out CNAME records observed fewer than 100 times or with a lifetime shorter than one day.

We find that CNAME records are extensively used in practice. While most CNAME records are used for subdomain aliasing, a non-negligible number of SLDs are also configured with CNAME records. Specifically, 21,660 SLDs and 8,268,312 subdomains are configured with CNAME records, spanning a total of 638,962 SLDs (63.90%). Alias relationships most frequently occur under the subdomain labels *www* (49.76%), *mail* (12.98%), and *smtp* (6.37%). The *www* subdomain is typically used for web services, while the latter two are commonly associated with email services.

We further analyze the lengths of CNAME chains and find that the majority have length one. Among the 21,660 SLDs with CNAME records, 14,088 have chains of length one, and 1,971 have chains of length three. Among the 8,268,312 subdomains with CNAME records, 5,592,054 have chains of length one, and 398,615 have chains of length three.

Finally, we analyze CNAME relationship patterns among popular SLDs and their subdomains. As shown in Table 1, CNAME mappings between subdomains are the most prevalent. Among the 21,660 SLDs with CNAME records, 21,062 (97.23%) have subdomains as their canonical names. Similarly, among the 8,268,312 subdomains with CNAME records, 7,675,317 (92.82%) point to subdomains as their canonical names. Furthermore, we find that most subdomains (65.54%) point to subdomains under different SLDs via CNAME records. The most common canonical SLDs include `cloudfront.net`, `amazonaws.com`, and `edgekey.net`, which are typically used for website and CDN hosting services. In the case of subdomains pointing to SLDs via CNAME records, we observe that common SLDs are primarily providers offering blogging or posting platforms, using separate subdomains to host user-generated content, such as `write2me.nl`, `ameblo.jp`, and `blog.bg`.

Table 1: CNAME relationship patterns for popular domain names and their subdomains.

CNAME relationship pattern	Number of domains
ALL SLD CNAME*	21,660
SLD-to-SLD	707 (3.26%)
SLD-to-SLD (loop)†	182 (25.74%)
SLD-to-SUB	21,062 (97.23%)
SLD-to-SUB (same)	1,073 (5.09%)
ALL SUB CNAME	8,268,312
SUB-to-SUB	7,675,317 (92.82%)
SUB-to-SUB (same)	2,645,584 (34.46%)
SUB-to-SUB (loop)	473 (0.01%)
SUB-to-SLD	601,453 (7.27%)
SUB-to-SLD (same)	535,867 (89.09%)

* *SUB* is short for subdomain. *ALL SLD/SUB CNAME* denotes the number of SLDs/subdomains that configure CNAME records. *SLD-to-SLD* indicates that an SLD points to an SLD via CNAME records, and others follow the same convention.

† *loop* means that alias and canonical domain are identical; *same* means that belong to the same SLD.

2.3 Domain Ownership Verification

DOV is a mechanism used to verify a user's control over a domain, ensuring that domain services are provisioned only to authenticated entities. In cloud hosting scenarios, providers typically require customers to configure specific DNS records (i.e., challenge tokens) to complete DOV. Figure 1 illustrates a typical DOV workflow. When a user claims a domain for hosting, the provider issues DNS queries to verify whether the required challenge tokens have been correctly configured. Below, we introduce common DNS challenge tokens.

• **MX token**. Email hosting providers typically require the domain to configure MX records that point to their mail servers. The MX records specified by providers are usually fixed domains. For example, Namecheap Private Mail requires domains to set `mx1.privateemail.com` and `mx1.privateemail.com`.

• **TXT token**. TXT records primarily serve two purposes. First, they are used to publish sender authentication-related records required by email hosting services, most commonly

the Sender Policy Framework (SPF). SPF records define which servers are authorized to send emails on behalf of a domain. By configuring an SPF record, a domain authorizes the email hosting provider to send emails on its behalf. The SPF records specified by providers typically follow fixed templates. For example, Namecheap Private Mail requires domains to set *v=spf1 include: spf.privateemail.com ~all*.

Second, TXT records can serve as random challenge tokens: providers require users to configure a TXT record containing a randomly generated string to verify domain ownership. For example, Zoho Mail requires domains to configure a random TXT record such as *zoho-verification=<random string>* [47]. In addition, email hosting providers can also use DomainKeys Identified Mail (DKIM) records as challenge tokens. DKIM is a mechanism that uses digital signatures to ensure the integrity of email content. DKIM records are published as TXT records under domains containing specific prefixes, known as selectors, such as `selector1._domainkey.foo.com`. Therefore, providers can also use DKIM records containing unique public keys and selectors as random challenge tokens.

• **CNAME token**. CNAME records are commonly used for web service hosting. Providers typically require domains to configure a CNAME record that points to a subdomain under the provider's controlled domain. The subdomain label is randomly generated (e.g., Azure Websites [18]) or derived from user-specific information such as account names (e.g., GitHub Pages [26]) and project names (e.g., Netlify Apps [8]).

• **A token**. A records are also commonly used for web service hosting. Providers require domains to configure A records that point to their web servers. These A records may belong to a fixed set or be allocated from an IP pool. For example, to host a website on GitHub, a domain can configure its A records to *185.199.108.153*, *185.199.109.153*, *185.199.110.153*, and *185.199.111.153*. In addition, although some providers require domains to configure CNAME-based challenge tokens, during the DOV process, they validate the A records obtained after CNAME resolution (e.g., GitHub) [46].

Overall, randomized DNS challenge tokens mainly take three forms: random TXT records, randomly assigned CNAME domain labels, and sufficiently large pools of A records. They can mitigate domain hosting takeover attacks based on dangling DNS records [46], since attackers cannot configure DNS records for victim domains.

Currently, the IETF has provided some guidance regarding DOV challenge tokens. For example, RFC 8555 [36] specifies the method for generating DOV tokens during the certificate issuance process, with a particular requirement for using random characters. Additionally, the DNSOP working group in an IETF draft lists recommendations for the use of DNS challenge tokens [38], including the types of challenge tokens, the use of random tokens, etc. In particular, this document describes CNAME considerations in DOV (e.g., dangling CNAME); however, it does not cover the threat scenario presented in this paper.

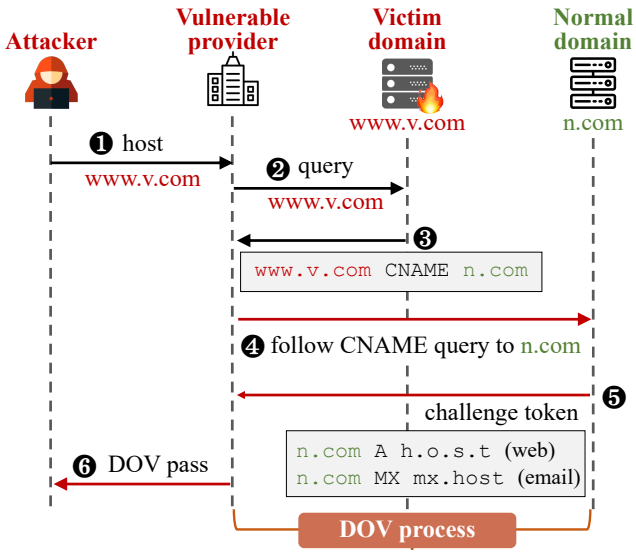

Figure 4: Process of the ALIASLEAP attack.

3 Threat Model

This paper presents a novel domain hosting service takeover attack, termed ALIASLEAP, which exploits flawed DOV mechanisms that follow CNAME records to perform cross-domain verification. This section first introduces the attack workflow and the potential impact of ALIASLEAP. Then, we compare ALIASLEAP with prior attacks.

3.1 ALIASLEAP Attack

ALIASLEAP targets the takeover of domain hosting services, such as email and website. The vulnerability arises because, during DOV, hosting providers follow CNAME records to perform DNS queries and treat the final DNS records obtained as challenge tokens for the domain being verified. As shown in Figure 2, if the canonical domain has configured valid challenge tokens, the provider erroneously assumes that the alias domain has also configured the correct tokens. In reality, the alias domain has never configured these tokens within its own domain zone. Overall, providers conflate alias equivalence at the DNS resolution layer with administrative control over the namespace, extending the capabilities of CNAME records.

Figure 4 illustrates the detailed attack process of ALIASLEAP for both email and web hosting scenarios. First, the attacker registers the victim domain (`www.v.com`) with the hosting provider (❶). The provider then performs DOV by querying the DNS records of `www.v.com` (❷). Because `www.v.com` is configured with a CNAME record pointing to `n.com` (❸), the hosting provider ultimately retrieves DNS records of `n.com` (❹). Since `n.com` is configured with valid challenge tokens (❺), the provider incorrectly concludes that `www.v.com` has passed DOV and activates hosting services

for the victim domain (❻).

In summary, hosting providers vulnerable to AliasLeap must satisfy three requirements. First, the provider adopts flawed DOV mechanisms, including non-random challenge tokens and cross-domain validation that follows CNAME records. Second, the provider offers management interfaces for domain hosting services. For email services, these interfaces mainly include mail routing and mailbox management. For web services, they mainly include website deployment and certificate application. Third, the provider supports multi-tenant shared hosting services. This allows Internet users to access the hijacked services of the victim domain through the infrastructure of the normal canonical domain.

From the attacker's perspective, the cost of launching AliasLeap is minimal. An attacker only needs to collect CNAME chains through active DNS scanning or by analyzing PDNS data, and then register the victim domain with the hosting provider. Throughout this process, the attacker does not need to control any DNS records or deploy any servers. After hijacking the email service, the attacker can send and receive emails via provider-managed mailboxes and mail routing. After hijacking the web service, the attacker can host malicious websites on the provider's servers.

The victim domain of AliasLeap can be either a subdomain or an SLD. A domain becomes a victim of AliasLeap only when two conditions are satisfied. First, the victim domain points, via a CNAME record, to a domain that is configured with valid challenge tokens, which can normally be hosted by the provider. Notably, AliasLeap can affect all alias domains along a CNAME chain. Second, the victim domain must not already be registered with the hosting provider. Furthermore, to take over an SLD's email service, the attacker needs to inject the SLD's CNAME record into the cache of the DNS resolver used by the provider; only under this condition will the provider obtain challenge tokens (e.g., MX records) by following CNAME records.

After the attacker takes over the domain hosting service, Internet users' access to the victim domain is redirected to the hosting provider via the CNAME record. The provider then delivers the hijacked service based on domain-specific fields in user requests. For web services, the `Host` header and `SNI` field in HTTP requests identify the accessed domain. In particular, the web hosting provider automatically configures valid certificates for the victim domain. For email services, the `RcptTo` field in SMTP requests specifies the target domain service. More importantly, emails sent using the victim domain can pass SPF validation. Specifically, the email sending servers of the victim domain (`www.v.com`) and the canonical domain (`n.com`) both belong to the hosting provider's multi-tenant shared infrastructure. Email sending servers are included in the SPF record of `n.com`. During SPF validation for (`www.v.com`), receivers follow the CNAME to eventually resolve to the SPF record of `n.com`, resulting in the SPF pass for emails from (`www.v.com`). For DKIM, vulnerable providers do not enforce its configuration. In practice, emails passing only SPF are still accepted by major email providers [27, 45].

3.2 Harm of AliasLeap Attack

The web is the most widely used Internet application, and email is not only a critical medium for online communication but also plays a central role in security authentication. The takeover of these domain services can therefore result in severe real-world consequences. In the following, we present five specific threat scenarios.

• **Launching phishing attacks**. After taking over the email service, an attacker can impersonate the victim domain to send phishing emails. Through hijacking web services, the attacker can leverage the victim domain's reputation to deploy phishing websites. These attacks enable many malicious objectives, such as lateral movement and extortion.

• **Obtaining valid certificates**. According to the CA/Browser Forum baseline requirements [23], certificate authorities (CA) allow applicants to receive validation links through specific email accounts, such as "admin" or "postmaster" to complete DOV and obtain certificates[1]. In addition, after successfully taking over a web service, providers typically automatically request TLS certificates for the customer's domain. Once a valid certificate is obtained, the attacker can conduct man-in-the-middle (MITM) attacks to intercept secure connections or carry out phishing attacks.

• **Thefting privacy**. The attacker can receive all emails and web requests destined for the victim domain, potentially leading to large-scale leakage of user information and the covert exfiltration of sensitive business data.

• **Hijacking account**. The attacker can abuse "forgot password" mechanisms to reset accounts on online services associated with the taken-over email domain, including payment platforms, social media services, and cloud providers.

• **Poisoning domain reputation**. Prior work by Li et al. [30] demonstrates that popular domain blocklists, such as Spamhaus [40], classify domains as malicious based on their email sending behavior. Consequently, an attacker can degrade the reputation of a victim domain by sending spam, thereby disrupting its legitimate network services. Moreover, Li et al. [30] show that many widely used domain blacklists operate at the SLD level. As a result, even if an attacker gains control over only a subdomain, the reputation of the corresponding SLD can be affected. In particular, some domain registries directly suspend an SLD based on blocklists [30, 37, 44], potentially causing the victim domain to be unusable.

3.3 Comparison with Previous Attacks

Prior studies have proposed exploiting CNAME records to achieve domain hosting service takeover. In Table 2, we com-

[1] Under the updated CA/Browser requirements effective March 31, 2026 [23], the email-based DOV method will be gradually deprecated.

Table 2: Comparison of previous CNAME-based domain hosting service takeover attacks.

CNAME-based domain takeover	Attack condition		Attack scenario	Attack cost	Impact scope
	Non-dangling*	Non-random†			
[CCS '16] Liu et al. [32]	✗	✗	Web	High	Low (10k)
[USENIX '21] Squarcina et al. [41]	✗	✗	Web	High	Low (1k)
[Sigmetrics '23] Zhang et al. [46]	✗	✗	Web	High	Low (10k)
AliasLeap	✓	✗	Email&Web	Low	High (2M)

* ✓ means that the attack does not rely on dangling DNS resources; ✗ means that it depends on dangling DNS resources.
† ✗ indicates that the attack relies on the DOV mechanism not to use randomized verification tokens.

pare AliasLeap with these existing attacks [32, 41, 46]. Below, we provide a detailed discussion from four perspectives.

• **Attack conditions**. Previous CNAME-based domain takeover attacks rely on dangling DNS records, such as expired domains or deprovisioned cloud instances. In contrast, AliasLeap exploits providers' incorrect trust in DNS records obtained by following CNAME chains during DOV and does not rely on any dangling resources. Similar to prior attacks, AliasLeap requires that the DOV mechanism not employ randomized challenge tokens.

• **Attack scenarios**. Prior studies typically examine CNAME-based hosting service takeovers only in web-related scenarios, such as websites, CDNs, and object storage services. This focus stems from the fact that web providers often require customer domains to configure CNAME records during DOV. In this paper, we demonstrate AliasLeap in both email and web scenarios, showing that the core attack idea is broadly applicable to hosting services across diverse types.

• **Attack cost**. Previous attacks depend on dangling resources, requiring attackers to monitor domain states and passively wait for victim domains to expose dangling DNS records. This dependence limits attackers' control over the timing and cost of the attack. In contrast, the CNAME chains exploited by AliasLeap are typically configured for legitimate services and are therefore more stable than dangling DNS records. Once a vulnerable CNAME chain is identified, an attacker can directly take over the hosting service of the victim domain, resulting in significantly lower attack cost.

• **Impact scope**. Prior works report that the number of domains exposed to dangling CNAME risks is on the order of tens of thousands, largely because user misconfigurations that result in dangling records are relatively uncommon in practice. In contrast, our measurement results show that AliasLeap can affect over two million domains. This disparity highlights that protocol-level misunderstandings pose a substantially broader security risk than isolated user misconfigurations.

4 Detecting AliasLeap Risks

In this section, we first introduce the key ideas behind detecting AliasLeap risks. We then present the architecture and implementation process of our detection system.

4.1 Key Insights

The detection of AliasLeap risks should enable large-scale measurement while adhering to ethical requirements [12, 29]. This necessitates minimizing manual intervention and ensuring that all probing activities are non-intrusive. In the following, we introduce two key tasks.

• **Identifying vulnerable providers**. A prerequisite for AliasLeap is that DOV does not rely on randomized challenge tokens. Based on a preliminary investigation of several hosting providers, we observe that randomized tokens are typically issued via TXT records [18, 47]. Therefore, by analyzing the prevalence of randomized TXT records among domains hosted by a given provider, we can identify providers that are unlikely to use random challenge tokens. This approach substantially reduces the need for manual testing of providers.

• **Probing domain hosting status**. AliasLeap only affects domains that are not registered with providers. We need to determine the hosting status of domains not under our control. This necessitates careful consideration of research ethics, particularly when probing email services. We propose a non-intrusive detectability method based on providers' error responses. For email services, providers return distinct non-delivery reports (NDRs) when receiving SMTP `RcptTo` command addressed to non-existent users under hosted versus non-hosted domains. Accordingly, we generate non-existent email addresses under the target domain and attempt delivery to elicit these NDRs. No real mailboxes received email content throughout the above process. For web services, we determine domain hosting status based on HTTP responses, following methodologies in prior work [46].

4.2 Detection Workflow

Figure 5 illustrates the workflow of *HostingEye*, the AliasLeap risk detection system, which comprises three main modules. First, we identify email and web hosting providers that are vulnerable to the attack. Next, we collect domains that point to these vulnerable providers at scale. Fi-

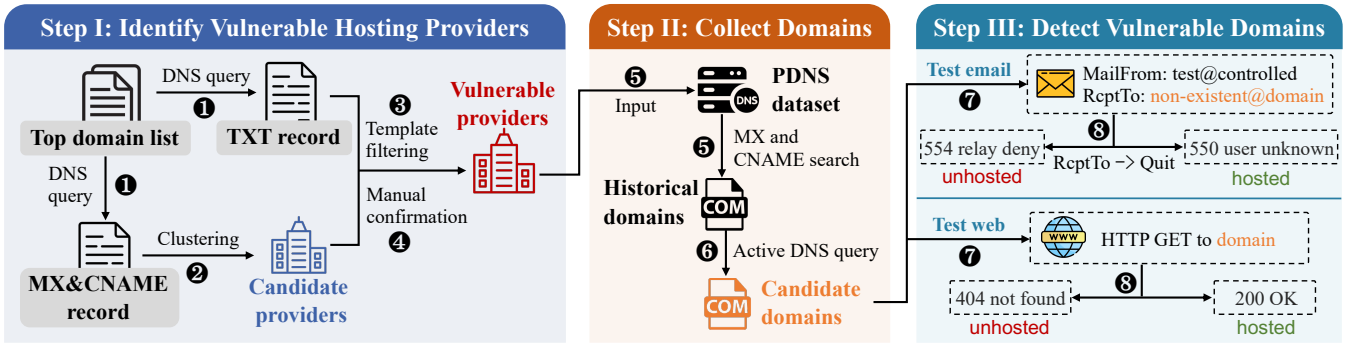

Figure 5: Workflow of the ALIASLEAP risk detection system.

nally, we conduct large-scale detection to identify domains that can be taken over by attackers. Below, we describe the workflow of the detection system in detail.

Collect and filter hosting providers. First, we compile an initial list of popular email and web hosting providers. Using active DNS scanning, we collect the MX records and *www* subdomain CNAME records of domains in the Tranco Top 1M domain list (❶) [43], and count the number of domains hosted under the SLDs of these MX and CNAME records (❷). We treat SLDs as candidate hosting providers. Our result shows that over 99% of providers host fewer than 200 domains, exhibiting a pronounced long-tail distribution. We retain only those hosting more than 200 domains, resulting in 124 MX SLDs and 165 CNAME SLDs as candidate providers for subsequent analysis. We believe that this set sufficiently covers the vast majority of popular providers.

Next, we query the TXT records of SLDs and *www* subdomains in the Tranco Top 1M domain list and further filter providers by analyzing the fraction of hosted domains that configure randomized TXT challenge tokens at each provider (❸). Randomized TXT tokens typically appear in two forms: (1) a fixed prefix followed by a sequence of random characters, often separated by "=" or ":", and (2) a standalone string of random characters. Because each provider generally adopts a fixed template for generating challenge tokens, this allows us to infer the presence of randomized tokens for providers.

Specifically, we use Shannon entropy to identify TXT records that contain random strings. If the entropy of a string is greater than 0.35 and the string length exceeds 5 characters, we consider it to be randomly generated. We adopt a relatively strict threshold to avoid filtering out too many hosting providers. To handle the first token form, entropy is computed only over the substring following the delimiter ("=" and ":"). Next, we derive TXT templates by analyzing the length of the random string, character distribution, and encoding scheme. For example, *BSI+digits(14)* denotes a 14-digit numeric string with the prefix "BSI", while "random(40)" denotes a 40-character random string. Manual inspection of these templates reveals that many domains configure randomized TXT records for non-email and non-web services (e.g., *dropbox-domain-verification=123abc*), which could interfere with our analysis. More templates for unrelated TXT records can be found in Appendix A. After excluding unrelated TXT templates, we compute the proportion of domains hosted by each provider that configure randomized TXT templates. Considering the cost of subsequent manual validation, we retain 50 email and 66 web candidate providers for which fewer than 20% of hosted domains use randomized TXT templates.

Confirm vulnerable hosting providers. We perform manual end-to-end testing on candidate providers to confirm their vulnerability (❹). Our test process is as follows[2].

1) We first host a controlled domain on the candidate provider and configure it with valid challenge tokens. This domain acts as a canonical domain for normal hosting.

2) We then set up two controlled victim domains: a subdomain and an SLD, each pointing to the normally hosted domain via CNAME records. For the victim SLD, we additionally inject its CNAME record into open DNS resolvers, as detailed in Section 5.2. Additionally, we configure CNAME chains of length three for the victim domains, ultimately pointing to the normally hosted domains, to evaluate whether CNAME chain length affects ALIASLEAP.

3) Next, we register the victim domains with the candidate providers. If the provider's website interface indicates successful registration, we further test the availability of the takeover services.

• For email services, we use Gmail accounts to verify that sending and receiving emails for the victim domains function correctly. We consider the victim's email service to have been taken over only if the email passes SPF verification and reaches the inbox rather than the spam folder.

• For web services, we deploy a web page for the victim domain and use a browser to access the victim domain. We consider the victim's web service to have been taken over

[2]In Appendix B, we use one email provider and one web provider as examples to present the detailed testing process, along with key screenshots.

only if the webpage content is displayed correctly and the certificate validation succeeds.

Among the 116 candidate providers, we find that 39 do not offer public hosting services, and 52 restrict account registration due to regional limitations or other factors. Of the remaining 25 providers where account registration is successful, we confirm that four email providers and seven web providers are vulnerable to the AliasLeap attack. There are two reasons why other 14 hosting providers are not affected by AliasLeap. Five providers do not follow the CNAME to obtain the challenge token in their DOV process. Nine providers use DKIM records containing random characters as the challenge token. Since obtaining DKIM records requires querying TXT records with a specific prefix, our TXT template filtering method cannot cover DKIM records.

Collecting candidate domains. The Internet domain namespace is vast; for example, the Tranco Top 1M domains collectively contain tens of millions of subdomains [46]. Actively enumerating domains hosted by vulnerable providers is inefficient and offers limited coverage of subdomains. Therefore, we leverage PDNS data to collect candidate victim domains for large-scale detection (❺). Our PDNS data is continuously gathered from multiple global DNS resolvers operated by 114DNS [16], one of the largest DNS providers in China.

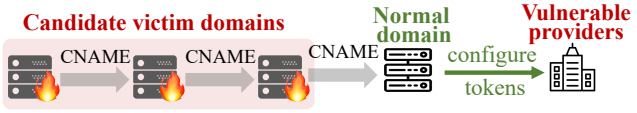

Figure 6: Collection scope of candidate victim domains.

The scope of domains we collect is shown in Figure 6. We first extract 7.8 million domains from PDNS that point to vulnerable email providers via MX records, as well as 25.2 million domains that point to vulnerable web providers via CNAME or A records[3]. The PDNS dataset spans the period from 2020 to 2025. These domains correspond to the normal domains in the Figure 6. Next, we collect 11.8 million domains from PDNS that point to normal email domains via CNAME records, and 23.9 million domains that point to normal web domains via CNAME records. We limit the length of CNAME chains pointing to normal domains to at most three. These domains constitute the set of candidate victim domains. Since we focus on domains that remain at risk, we perform active DNS queries between November 02 and November 05, 2025 to ensure that DNS record configurations for both normal domains and candidate victim domains satisfy the conditions in Figure 6 (❻). In total, we collect 2.1 million candidate victim email domains and 0.7 million candidate victim web domains.

Identifying vulnerable domains. Below, we measure the takeover risk of the candidate victim domains. All candidate domains have already been confirmed to resolve to the challenge tokens required by their respective providers; therefore, any candidate domain that is not currently hosted is vulnerable. We infer takeover potential by leveraging differences in provider responses based on domain hosting status.

As shown in Table 8 and Table 9 in the appendix, all vulnerable providers return distinct responses when handling requests for hosted versus unhosted domains. For example, when the email address in the SMTP `RcptTo` command belongs to a non-existent mailbox under a hosted domain, Namecheap responds with: *<non-exist@hosted.domain>: Recipient address rejected*. In contrast, when the email address belongs to an unhosted domain, the response is: *<non-exist@unhosted.domain>: Relay access denied*. For web providers, when an HTTP request targets a hosted domain, the provider returns a 200 OK status code. In contrast, unhosted domains return a 404/409/403 status code. Additionally, providers may include custom error pages that reveal domain hosting status; for example, GitHub returns the web page title "*Site not found · GitHub Pages*."

Finally, we sequentially send the `EHLO`, `MailFrom`, `RcptTo`, and `QUIT` SMTP commands to each candidate email domain. The sender address is under a domain we control, while the recipient address is a randomly generated 20-character username from all letters and digits under the candidate domain (❼). In general, users are unlikely to use such meaningless and complicated usernames. As such, all email probes involve incomplete SMTP connections targeting non-existent mailboxes, with only a single test required per domain. For candidate web domains, we sequentially send HTTP GET requests (❼). By analyzing SMTP error messages and HTTP responses (❽), we identify over two million domains that are not hosted on providers and thus are vulnerable to AliasLeap.

5 Evaluating AliasLeap Risks

This section first examines the usage strategy of DNS resolvers for CNAME records. After that, we conduct a comprehensive analysis of hosting providers and domains that are vulnerable to AliasLeap. Finally, we investigate the duration of the domain's exposure to the AliasLeap threat.

5.1 CNAME Usage Across DNS Resolvers

The DOV mechanism needs to use DNS resolvers to query DNS records. Therefore, how DNS resolvers follow CNAME chains to query DNS records is crucial to the AliasLeap attack. Using domains under our control, we configure CNAME records for a subdomain and an SLD, which ultimately resolve to A, MX, and TXT records. We then query 14 popular DNS resolvers, along with a self-built resolver (using Bind software), for the DNS records of the alias domains. Finally, we analyze whether these resolvers successfully return the final DNS records.

[3]Table 8 and Table 9 in the appendix list specific DNS records.

Table 3: The usage strategy of popular DNS resolvers for CNAME records.

DNS resolver	Subdomain CNAME*			SLD CNAME (no cache)†			SLD CNAME (cache)†		
	A	MX	TXT	A	MX	TXT	A	MX	TXT
Google, 8.8.8.8	✔§	✔	✔	✔	✗	✗	✔	✔	✔
Cloudflare, 1.1.1.1	✔	✔	✔	✔	✗	✗	✔	✗	✗
Quad9, 9.9.9.9	✔	✔	✔	✔	✗	✗	✔	✔	✔
OpenDNS, 208.67.220.120	✔	✔	✔	✔	✗	✗	✔	✔	✔
Level3, 4.2.2.1	✔	✔	✔	✔	✗	✗	✔	✔	✔
Yandex, 77.88.8.1	✔	✔	✔	✔	✗	✗	✔	✗	✗
ControlD, 76.76.2.0	✔	✔	✔	✔	✗	✗	✔	✔	✔
AdGuard, 94.140.14.14	✔	✔	✔	✔	✗	✗	✔	✗	✗
Dyn, 216.146.35.35	✔	✔	✔	✔	✗	✗	✔	✔	✔
DNSPod, 119.28.28.28	✔	✔	✔	✔	✗	✗	✔	✗	✗
Clean Browsing, 185.228.168.10	✔	✔	✔	✔	✗	✗	✔	✔	✔
Ali, 223.5.5.5	✔	✔	✔	✔	✗	✗	✔	✗	✗
OneDNS, 117.50.10.10	✔	✔	✔	✔	✗	✗	✔	✗	✔
Comodo Secure, 8.20.247.10	✔	✔	✔	✔	✗	✗	✔	✔	✔
Our server (Bind version 9.18.30)	✔	✔	✔	✔	✗	✗	✔	✔	✔

* Subdomain points to another domain via CNAME records.

† SLD points to another domain via CNAME, under two DNS resolver conditions: with/without cached SLD's CNAME records.

§ ✔ means that the DNS resolver can obtain the final DNS record via the CNAME record; ✗ means that it cannot.

As shown in Table 3, when querying subdomains, all resolvers follow CNAME records to retrieve the final records. However, resolver behavior differs when encountering SLD's CNAME records. Specifically, if a resolver does not have the SLD's CNAME record cached, it follows the CNAME chain only for A record queries; for MX and TXT queries, the resolver returns DNS responses with no answers.

We further examine DNS resolution behavior when resolvers have cached the SLD's CNAME record. Considering that large DNS resolvers operate multiple backend IP addresses, we send 50 queries for A and CNAME records to each resolver to populate their caches. The results indicate that resolver behavior changes significantly under this condition. Specifically, 10 resolvers (e.g., *8.8.8.8*) leverage cached CNAME records to return results for MX and TXT queries, whereas five resolvers (e.g., *1.1.1.1*) still fail to return the final MX or TXT records. OneDNS (*117.50.10.10*) uses cached CNAME records only to resolve MX records.

Additionally, we evaluate how open DNS resolvers in the wild handle CNAME records. We first collect 300K open DNS resolver IP addresses that were continuously active between October and December 2025. Measurements are then performed using the same methodology as for popular DNS resolvers. The results indicate that for subdomain CNAME records, nearly all open DNS resolvers follow the CNAME chain to resolve queries. Furthermore, 70.35% of open DNS resolvers leverage cached CNAME records of SLDs to obtain the final MX and TXT records.

Overall, when a provider relies on open resolvers for DOV, taking over a subdomain's hosting service or an SLD's web service is straightforward and only requires registering with the provider, whereas taking over an SLD's email service requires additional CNAME cache injection.

5.2 Vulnerable Hosting Providers

Email hosting providers. As shown in Table 4, we identify four email hosting providers vulnerable to the AliasLeap attack, such as Alibaba Cloud [1], Namecheap [7], and ImprovMX [5]. Namecheap does not permit customers to register any subdomains, and Anon [4] restricts registration of *www* subdomains (e.g., `www.example.com`). Furthermore, the length of the vulnerable CNAME chain does not affect the success of email service takeovers.

Compared with subdomains, attacks on the email services of SLDs are more complex. We first investigate the DNS resolvers used by the four email hosting providers during DOV. Specifically, we deploy a controlled authoritative name server and host an experimental domain on it. We then register this domain with each provider and monitor the IP addresses that query our authoritative name server. Additionally, we correlate these IPs with the published backend IP ranges of Google DNS (*8.8.8.8*) [14] and Cloudflare DNS (*1.1.1.1*) [13] to determine resolver ownership.

We find that Namecheap and ImprovMX rely on Google DNS to perform DNS queries during DOV. For Anon and

[4] After consultation with an email provider, we omitted its specific name. In addition, we anonymized victim domains.

Table 4: Email hosting providers and domains vulnerable to the ALIASLEAP attack.

Email provider (Product)	SUB*	WWW SUB	SLD	Candidate SUB	Candidate SLD	Vulnerable SUB	Vulnerable SLD	Top Vulnerable§ SUB	Top Vulnerable§ SLD
Anon (Enterprise Mail)¶	✔†	✗	✔	1,836,010	2,256	1,525,962	1956	131,877	8
Alibaba Cloud (Enterprise Email)	✔	✔	✔	175,256	332	94,170	72	156	4
Namecheap (Private Email)	✗	✗	✔	/	655	/	492	/	5
ImprovMX (Email Forwarding)	✔	✔	✔	45,206	13	29,172	12	24	0
Total	3	2	4	2,056,472	3,256	1,649,304	2,532	132,057	17

¶ According to the provider's requirements, we anonymize the specific name.
* SUB is short for subdomain. WWW SUB denotes domains whose third-level label is *www* (e.g., `www.example.com`).
† ✔ indicates that the domain can be taken over at the provider, whereas ✗ indicates that it cannot.
§ Top Vulnerable refers to victim domains that appear in the Tranco Top 1M domain list.

Table 5: Web hosting providers and domains vulnerable to the ALIASLEAP attack.

Web provider (Product)	SUB	WWW SUB	SLD	Candidate SUB	Candidate SLD	Vulnerable SUB	Vulnerable SLD	Top Vulnerable SUB	Top Vulnerable SLD
GitHub (Page)	✔	✔	✔	230,603	644	158,712	259	16,746	4
Netlify (Website)	✔	✔	✔	186,258	3,038	148,397	1,153	12,339	6
Render (Website)	✔	✔	✔	179,391	610	139,646	194	52	2
Framer (Website)	✔	✔	✔	18,754	11	17,109	11	170	3
Nuvemshop (Website)	✔	✔	✔	12,709	9	6,288	6	1,736	0
Vercel (Website)	✔	✔	✔	986	28	579	6	12	0
Hostinger (Website)	✔	✗	✔	149	12	56	5	0	0
Total	7	6	7	628,850	4,352	470,787	1,631	31,055	15

Alibaba Cloud, DNS queries originate from clusters of IP addresses. The IP addresses used by Anon belong to AS45090 (Shenzhen Tencent Computer Systems Company Limited), while those used by Alibaba Cloud belong to AS37963 (Hangzhou Alibaba Advertising Co., Ltd.) [28]. Based on the AS names, we infer that these IP addresses correspond to two public DNS providers[5]. However, because neither two DNS providers publicly discloses the backend IP ranges of their DNS resolvers, we cannot precisely identify the specific resolvers used by Anon and Alibaba Cloud.

We then evaluate whether CNAME cache injection against SLDs is feasible on the DNS resolvers used by the four email providers. The primary challenge is that we cannot directly send DNS queries to the backend IP addresses of these resolvers, and large public DNS resolvers typically select backend servers based on the client's geographic location. To overcome this limitation, we send one A-record query and one CNAME-record query for the experimental SLD to 300K open DNS resolvers that were continuously active between October and December 2025. Since many open resolvers operate as forwarders, they relay queries to upstream large DNS resolvers. This strategy effectively enables us to issue DNS queries to the target resolvers from geographically distributed vantage points. Throughout the process, we strictly control the query rate, limiting it to 20 DNS packets per second. Finally, we attempt to register the experimental SLD with the email providers to determine whether the attack succeeds.

The results show that we successfully registered the experimental SLD with all four email hosting providers, where the SLD was pointed via CNAME records to domains configured with valid challenge tokens. Notably, the attacks against Namecheap and ImprovMX succeeded within about 10 minutes after initiating the CNAME cache injection, whereas the attacks against Anon and Alibaba Cloud required about 2 hours. We conjecture that this discrepancy arises because relatively few DNS forwarders use the DNS resolvers employed by Anon and Alibaba Cloud as their upstream backends.

Web hosting providers. As shown in Table 5, we identify seven web hosting providers that are vulnerable to the ALIASLEAP attack: GitHub [3], Render [10], Nuvemshop [9], Netlify [8], Framer [2], Vercel [11], and Hostinger [4]. The impact of ALIASLEAP affects their website hosting services. Among them, Hostinger does not allow customers to register *www* subdomains. In addition, these seven providers use a fixed set of A records as challenge tokens, either through

[5] DNSPod Public DNS (119.28.28.28), Ali DNS (223.5.5.5, 223.6.6.6).

direct A records (e.g., Hostinger) or A records obtained after CNAME resolution (e.g., GitHub). Similar to email providers, the length of the vulnerable CNAME chain does not affect the success of the web service takeover.

5.3 Vulnerable Domains

Table 4 and Table 5 summarize the number of domains affected by the AliasLeap attack. For email hosting services, among the 2,056,472 subdomains and 3,256 SLDs we tested, 1,649,304 subdomains and 2,532 SLDs are vulnerable AliasLeap. For web hosting services, among the 628,850 subdomains and 4,352 SLDs tested, 470,787 subdomains and 1,631 SLDs could be taken over. All vulnerable domains cover a total of 181,827 SLDs. Furthermore, we find that the CNAME chain length between 99.42% of vulnerable domains and the normal domains hosted on providers is one.

5.3.1 Vulnerable Domain Distribution

Domain popularity. Our results indicate that AliasLeap impacts a substantial number of high-profile domains. We analyze the popularity of vulnerable domains using domain ranking lists [43]. For email domains, within the Tranco Top 1M domain list, we identify 132,057 affected subdomains spanning 284 SLDs. Within the Tranco Top 10K, we find 11,108 affected subdomains, including domains under `cloudflare.net` (rank 79), `whecloud.com` (rank 383), `dnspod.net` (rank 483), `tencent.com` (rank 708), `eu.com` (rank 1,304), `us.com` (rank 1,772), `utoronto.ca` (rank 2,408), `dnsv4.com` (rank 2,813), `org.ru` (rank 2,876), `dnspod.com` (rank 3,456), and `eu.org` (rank 3,830), etc.

In addition, we find that the email services of 17 popular SLDs can be taken over by attackers. The highest-ranked among these is the medical organization domain `d**n.com` (rank 40,391), which points via a CNAME record to another domain (`d**y.cn`) owned by the same organization and hosted on Anon. Moreover, for these popular SLDs, a large number of clients frequently query their A records. Consequently, we find that large DNS resolvers typically cache these popular SLDs' CNAME records, and attackers can directly register them without performing CNAME cache injection.

For web domains, within the Tranco Top 1M domain list, we identify 31,055 affected subdomains spanning 753 SLDs. Within the Tranco Top 10K, we find 308 affected subdomains, including domains under `cloudflare.net` (rank 79), `it.com` (rank 1,272), `dnsv1.com` (rank 1,654), `us.com` (rank 1,772), `eu.org` (rank 3,830), `cnr.it` (rank 4,142), `qiniudns.com` (rank 4,950), and `pp.ua` (rank 5,237), etc. In addition, we identify 15 popular SLDs that are vulnerable to AliasLeap. The highest-ranked among these is the technology company domain `t**a.dev` (rank 528,182), which points via a CNAME record to another domain (`t**a.at`) owned by the same company.

Moreover, we analyze the client usage scale of vulnerable domains based on their DNS query traffic. Using the PDNS dataset from 114DNS [16], we count the daily number of A record queries for all vulnerable domains between January 1 and December 1, 2025, as shown in Figure 7. Our results indicate that vulnerable domains are actively used in the real world, and their takeover could therefore have substantial practical consequences. Specifically, the daily number of A record queries for vulnerable email domains remains around two million, while for vulnerable web domains it is approximately 40K. It is important to note that these statistics represent a conservative lower bound on the actual DNS query traffic associated with vulnerable domains.

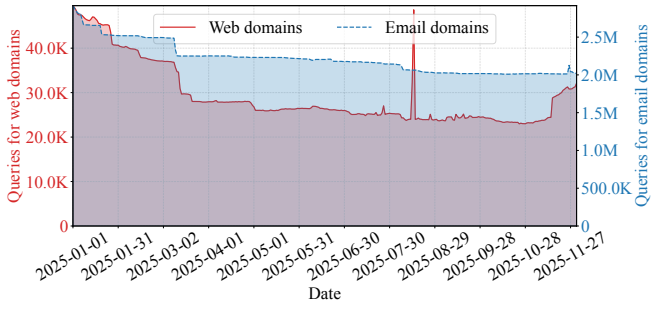

Figure 7: Daily number of A record queries for vulnerable email and web domains.

Industry category. We analyze the industry categories of vulnerable domains using VirusTotal [15]. Since VirusTotal provides limited category coverage for less popular domains, our analysis focuses on the 1,595 SLDs corresponding to popular vulnerable domains. As shown in Table 6, the majority of vulnerable domains are concentrated in the Business/Economy and Information Technology sectors, which together account for approximately half of all domains.

Furthermore, domains belonging to government agencies and educational institutions are particularly valuable due to their high perceived authority. We identify vulnerable domains in these categories (e.g., `gov.cn` for government, `edu.cn` for educational institutions) by matching them against publicly available domain suffix lists [35]. Our analysis reveals that 5,067 government domains and 87,560 educational domains are affected by the AliasLeap attack. Notable examples include subdomains under the State of New Hampshire (`nh.gov`), the Guangzhou Municipal People's Government of China (`thnet.gov.cn`), the University of Toronto (`utoronto.ca`), and Louisiana State University (`lsu.edu`).

TLD distribution. We investigate the distribution of top-level domains (TLDs) among vulnerable domains. Table 6 lists the five most common TLDs associated with 181,827 SLDs of all affected domains. We find that three generic TLDs (`com`, `org`, `net`) together account for over half of SLDs. Additionally, many vulnerable domains are distributed across country-code

Table 6: Industry category and TLD distribution of the SLDs of vulnerable domains.

Popular SLD (1,595)		**All SLD** (181,827)	
Business/Economy	24%	com	42%
Information Technology	21%	org	9%
Education	7%	co.uk	7%
Shopping	6%	ch	5%
Travel	5%	net	3%

TLDs, including `co.uk` and `ch`.

Subdomain label distribution. We examine the label distribution of vulnerable subdomains. The most frequent subdomain labels and their corresponding number of domains are: *www* (25,078), *ftp* (8,912), *mail* (7,420), *webmail* (4,532), *api* (3,794), *blog* (2,502), and *cpanel* (2,367). These labels correspond to critical Internet services, including websites, file transfer, email, and service APIs. Takeover of these subdomains would therefore pose significant security risks.

5.3.2 Vulnerable CNAME Relationship Patterns

In the following, we analyze the CNAME relationship patterns of vulnerable domains and attempt to understand the reasons behind their CNAME configurations.

We find that 81% of vulnerable subdomains use CNAME records pointing to their own SLDs. Such relationships are typically configured by domain administrators to satisfy system design requirements or operational needs. From an external perspective, the original motivations behind these CNAME configurations are difficult to accurately infer. The most common case is the *www* subdomain pointing to its SLD, which aligns with user conventions and standard web management practices. However, when the SLD hosts email services at a third-party provider, this setup enables attackers to take over the email service of the *www* subdomain. Beyond *www*, many *ftp*, *mail*, *api*, *blog*, and *cpanel* subdomains also point to their SLDs, making them susceptible to AliasLeap.

For the remaining 19% of CNAME relationships, we find that vulnerable domains often employ CNAME records pointing to other domains within the same organization. For instance, a code hosting platform, `coding.cloud.t**t.com`, points via a CNAME record to `c**g.net`. Because the email service of `c**g.net` is hosted on Anon, this configuration enables attackers to take over `coding.cloud.t**t.com`. Moreover, we examine the ten SLDs associated with the largest numbers of vulnerable subdomains for case analysis. Below, we present three common scenarios.

In the first scenario, companies offer services across multiple SLDs and use CNAME records for centralized management and scheduling. For example, the large DNS service provider offers domain resolution and hosting on subdomains under SLDs such as `d**4.net` and `n**4.cn`, all of which point via CNAME records to `d**d.net`. This setup enables the provider to manage numerous client domains simply by modifying the DNS records of `d**d.net`. However, because the email service of `d**d.net` is hosted on Anon, 29,082 subdomains are vulnerable to AliasLeap.

The second scenario arises from the company brand change, which creates CNAME redirections to connect the old and new service domains. For example, the large online data management platform migrated its service domains from `k**e.com` to `k**d.com` during the company's development. As a result, 12,744 subdomains under `k**e.com` point via CNAME records to `k**d.com`. Because the email service of `k**d.com` is hosted by Anon, these subdomains under `k**e.com` are vulnerable to AliasLeap.

The third scenario is corporate acquisition, in which the acquired company's domain is pointed to the new domain via a CNAME record. For example, after the well-known Chinese code collaboration platform `g**e.com` (similar to GitHub) was acquired by a company in 2016, many of its subdomains were redirected via CNAME records to the new code hosting platform (`c**g.net`), making 4,575 subdomains under `g**e.com` susceptible to takeover by attackers.

5.3.3 Escalated Damage by Email Authentication

Email plays a critical role in identity verification and account recovery processes. Below, we analyze the real-world impact that attackers can achieve by abusing the authentication function of the victim domain's email service.

Attackers can exploit email-based ownership verification to obtain TLS certificates for vulnerable domains [23]. By examining CA documentation and search results from Google, we identify seven CAs that support email-based verification: DigiCert [21], Sectigo [39], SSL.com [42], DNSimple [22], AWS [17], GeoCerts [25], and ComodoSSLStore [20]. After obtaining valid certificates, attackers can conduct malicious activities such as phishing and MITM attacks.

Next, we investigate email addresses associated with vulnerable domains. By intercepting emails sent to these addresses, attackers could potentially take over Internet services linked to them via "password reset" mechanisms. Given the ethical concerns of probing valid accounts by sending large volumes of emails to mail servers, we instead rely on leaked email address datasets to identify affected accounts. Using three large-scale breach datasets (Adobe, Anti Public, and Collection #1) [6, 30], we identify a total of 1,292 email addresses associated with vulnerable domains. This relatively small number is primarily attributable to the limited coverage of passive breach datasets and the fact that many vulnerable domains may have never deployed email services.

5.4 AliasLeap Risk Duration

Below, we analyze the duration for which domains are exposed to the AliasLeap attack, defined as the period during which vulnerable domains maintain CNAME chains pointing to hosting providers. Throughout this interval, attackers can launch AliasLeap to take over domain services. Our analysis is based on PDNS data from 114DNS [16], spanning 2015 to 2025. We emphasize that this exposure duration represents only a lower bound on the actual risk period.

As shown in Figure 8, vulnerable domains remain exposed to AliasLeap for extended periods, highlighting both the stealthiness and the severity of the threat. Specifically, more than 99% of vulnerable domains maintain CNAME chains with lifetimes exceeding one month. Furthermore, 64% of email domains and 45% of web domains remain at risk for over 12 months. Notably, 19,819 domains have exposure durations exceeding 10 years. For instance, a subdomain of the University of Toronto (`o**c.utoronto.ca`) has pointed to `i**c.ocadu.ca` via a CNAME record since 2016.

While the above findings may appear striking, they are also intuitive. Under the AliasLeap threat model, victim domains typically configure CNAME records for legitimate business or system design purposes, causing AliasLeap risks remain latent in the DNS ecosystem for many years. In contrast, domain hosting service takeovers caused by dangling DNS records due to user misconfiguration tend to have much shorter exposure windows. For example, Zhang et al. [46] in 2022 report that about half of the domains with dangling records remain vulnerable for no longer than 30 days.

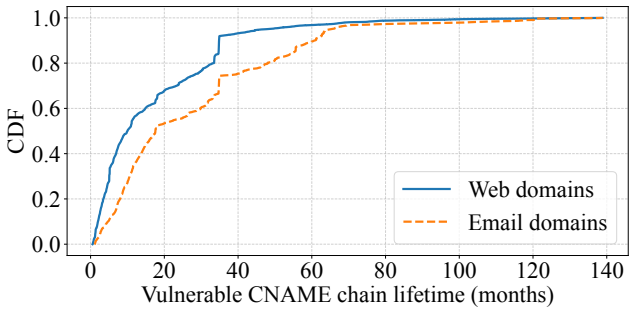

Figure 8: Distribution of risk duration of vulnerable domains.

6 Discussion

In this section, we first present mitigation strategies for AliasLeap along with the results of our vulnerability disclosure. Then, we discuss the limitations of our work.

6.1 Mitigation and Responsible Disclosure

The risk of the AliasLeap attack arises from hosting providers' incorrect reliance on DNS records obtained via CNAME redirection during DOV. Based on the prerequisites for the attack, we propose two mitigation measures. First, hosting providers can adopt DOV mechanisms that employ randomized challenge tokens, such as randomly generated TXT records. This approach ensures that an attacker would need to configure a unique DNS record for the domain on the CNAME chain of the victim domain, which is infeasible unless the attacker actually owns the domain. Second, hosting providers can prohibit cross-domain verification that follows CNAME records. A domain should be considered verified only when the requested DNS record is returned from the zone of the domain being registered.

We responsibly disclosed AliasLeap to 11 affected hosting providers, offering detailed explanations of the attack principles, reproduction videos, and recommended mitigation measures. To date, eight providers have confirmed the vulnerability. Among them, Anon, Alibaba Cloud, and Namecheap have disabled cross-domain validation following CNAME records to address the issue. Anon classified AliasLeap as the highest-severity vulnerability among its high-risk category and awarded a bounty of $800. Alibaba Cloud rated AliasLeap as a medium-risk vulnerability and rewarded us with a $200 bonus. For the remaining providers that have not yet remedied the vulnerability, GitHub and Netlify assessed AliasLeap as having limited real-world impact, while Hostinger attributed the risk to insecure DNS record configurations by domain owners. We continue to engage with these providers to address the vulnerability. In addition, we obtained 10 CNVD/CNNVD vulnerability numbers.

The systemic risks introduced by the implicit trust in CNAME semantics have not been adequately recognized by the security community, and they extend well beyond isolated incidents caused by individual misconfigurations, such as dangling DNS records. Given that CNAME records are deeply embedded in Internet architecture, we advocate for a re-examination and standardization of their use by the security community. In particular, attention should be paid to whether implementations of DNS resolvers and applications respect the intended capability boundaries of CNAME records.

6.2 Limitation

The limitations of this work primarily concern the assessment of AliasLeap risks. First, we perform end-to-end manual testing only for popular providers, and exclude many providers where account registration is unavailable.

Second, we rely on the PDNS dataset from 114DNS, a major DNS provider in China, which limits coverage of the global domain space. Additionally, we cap the maximum CNAME chain length at three, which may omit deeper chains. Since our results show that over 99% of vulnerable CNAME chains have a length of one, we believe that this cap is sufficient to capture the vast majority of vulnerable cases. In particular, our dataset and methodology have demonstrated

the significant risk of AliasLeap in the real world.

Third, due to the ethical risks associated with performing real-world domain takeovers, we infer the hosting status of domains indirectly based on error responses returned by hosting providers. Through manual validation, we confirm that these error responses reliably reflect the actual domain hosting status. The main limitation of this heuristic detection method is that hosting providers may change their response templates in the future, which would prevent us from inferring the registration status of domains. Considering that large-scale registration of uncontrolled domains may introduce ethical risks, we manually verified 14 domains with permission from Anon and found that all of them were affected by the AliasLeap attack. After completing the experiments, we immediately relinquished control of all domains.

7 Related Work

As a widely used DNS record type, CNAME is critical to the Internet ecosystem. Prior research on CNAME-related security risks has primarily focused on vulnerabilities arising from dangling CNAME records and cyclic CNAME chains.

Liu et al. [32] in 2016 conducted the first large-scale measurement of dangling DNS records in the wild, covering A, CNAME, MX, and NS records. They identified 260 dangling CNAME records among the Alexa Top 10K domains. Squarcina et al. [41] examined domain takeover threats across 26 million subdomains under the Tranco Top 50K domains and found 887 affected domains. Zhang et al. [46] developed an automated tool to detect subdomain web service takeover risks and identified 10,351 vulnerable subdomains in the Tranco Top 1M domain list over seven months. Focusing on CDN services, Lin et al. [31] discovered 1,449 dangling subdomains within the Tranco Top 1M domains. Moreover, Jens Frieß et al. [24] investigated the actual misuse of dangling DNS records by attackers in web hosting scenarios. They uncovered 20,904 cases of web service hijacking on the popular cloud platform.

Bushart et al. [19] in 2018 reported that CNAME chain loops can be exploited to launch denial-of-service attacks. Their experiments demonstrated that CNAME loop attacks could achieve an amplification factor of 8.51, and they identified more than ten DNS resolvers that could act as amplifiers. In addition, Moura et al. [34] in 2021 conducted further evaluations of CNAME loop attacks and found that most DNS resolvers are capable of detecting CNAME loops and terminating the resolution process.

In summary, prior works have focused on security risks caused by misconfigured or unconventional CNAME records. In contrast, our study demonstrates that even correctly configured CNAME chains can be exploited for domain takeover attacks due to hosting providers' incorrect assumptions about CNAME semantics.

8 Conclusion

By exploiting flawed cross-domain ownership verification that follows CNAME records, this paper presents a novel domain hosting service takeover attack, called AliasLeap. The attack has a wide impact in the real world and is highly stealthy. We identify 11 affected hosting providers and more than two million vulnerable domains. Many vulnerable CNAME chains are legitimately configured, with over half persisting for more than one month and approximately 20K lasting over 10 years. We proposed mitigation strategies and responsibly disclosed the vulnerability to the affected providers, receiving confirmations from eight of them. We hope that our findings prompt the Internet community to re-examine the implementation and security risks associated with CNAME records.

Ethical Considerations

Our study involves large-scale measurements of hosting providers and domains, which require careful consideration of potential ethical risks. We follow established research ethics guidelines [12, 29] and prior related works [24, 32, 46] to minimize ethical concerns throughout our experiments. Below, we outline our main ethical considerations.

End-to-end testing of hosting providers. In this paper, we manually test 25 hosting providers to assess their susceptibility to the AliasLeap attack. All experiments are conducted exclusively using domains under our control and through the providers' public website interfaces. In addition, with explicit authorization from Anon, we perform takeover validations on several popular domains to demonstrate the practical feasibility of AliasLeap. Upon completion of these experiments, we immediately relinquish control of all hosted domains, ensuring that no users or organizations are adversely affected by our manual testing.

Detecting domain hosting status. The primary ethical concern in identifying vulnerable domains arises from sending emails to target domains. To avoid any impact on real users, we infer the risk status of domains by leveraging hosting providers' distinctive error responses for non-hosted domains. Specifically, for each domain, we initiate at most one SMTP connection and use the randomly generated username as the recipient address. We terminate the SMTP session before sending the email body, ensuring that no email content is delivered. Overall, no real user mailboxes receive emails, and the interaction imposes negligible load on email servers.

Furthermore, we evaluate the real-world impact of the AliasLeap attack using a PDNS dataset. The data we obtained consists of DNS query logs aggregated by 4-tuples, including queried domains, answer records, query counts, and time intervals. The PDNS dataset we use does not contain any user-related sensitive information, such as client IP addresses.

Open Science

To facilitate replicability, we carefully selected publicly available datasets and tools in this study and published them at https://doi.org/10.5281/zenodo.20404540.

Risk detection system. For the HostingEye system, we release the code for identifying the email hosting and web hosting status of domains.

CNAME chains of popular domains. We release the CNAME chain dataset through PDNS to facilitate further analysis of real-world CNAME relationships.

For ethical reasons, we chose not to release the victim domain dataset. Many providers have not yet remediated the vulnerabilities, and releasing the complete dataset could enable attackers to conduct large-scale domain takeovers. In addition, some providers have requested that we not disclose their identities or the domains hosted on their platforms in order to avoid potential reputational damage and legal disputes.

Acknowledgment

We thank all anonymous reviewers for their valuable and constructive feedback. This work is supported by the National Key Research and Development Program of China (No. 2023YFB3105600), and the National Natural Science Foundation of China (Grant No. 62272413). Baojun Liu and Jun Shao are both corresponding authors.

References

[1] Alibaba cloud. https://cn.aliyun.com/.

[2] Framer. https://www.framer.com/.

[3] Github. https://github.com/.

[4] hostinger. https://www.hostinger.com/.

[5] Improvmx. https://improvmx.com/.

[6] Largest breaches. https://haveibeenpwned.com/.

[7] Namecheap. https://www.namecheap.com/.

[8] Netlify. https://www.netlify.com/.

[9] Nuvemshop. https://www.nuvemshop.com.br/.

[10] Render. https://render.com/.

[11] Vercel. https://vercel.com/.

[12] The belmont report: ethical principles and guidelines for the protection of human subjects of research. United States. National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. Department of Health, Education and Welfare, 1979.

[13] Backend ip addresses for 1.1.1.1. https://www.cloudflare.com/ips-v4, 2026.

[14] Backend ip addresses for 8.8.8.8. https://www.gstatic.com/ipranges/publicdns.json, 2026.

[15] Virustotal. virustotal.com/gui/, 2026.

[16] 114DNS. https://www.114dns.com/.

[17] Amazon. Aws certificate manager email validation. https://docs.aws.amazon.com/acm/latest/userguide/email-validation.html, 2026.

[18] Azure. Overview: Use custom domain names with azure app service. https://learn.microsoft.com/en-us/azure/app-service/overview-custom-domains.

[19] J. Bushart and C. Rossow. DNS unchained: Amplified application-layer dos attacks against DNS authoritatives. In *RAID*, volume 11050, pages 139–160. Springer, 2018.

[20] ComodoSSLStore. Domain validation using email-based validation. https://help.comodosslstore.com/support/solutions/articles/22000292638-domain-validation-using-email-based-validation, 2026.

[21] DigiCert. Use email verification to validate a domain on a pending dv tls certificate. https://docs.digicert.com/en/certcentral/manage-certificates/dv-certificate-enrollment/domain-control-validation--dcv--methods/use-the-email-dcv-method.html, 2026.

[22] DNSimple. Ssl certificate email-based domain validation. https://support.dnsimple.com/articles/ssl-certificates-email-validation/, 2026.

[23] CA/Browser Forum. Baseline requirements for the issuance and management of publicly-trusted tls server certificates. https://cabforum.org/working-groups/server/baseline-requirements/documents/CA-Browser-Forum-TLS-BR-2.2.6.pdf, 2026.

[24] J. Frieß, T. Gattermayer, N. Gelernter, H. Schulmann, and M. Waidner. Cloudy with a chance of cyberattacks: Dangling resources abuse on cloud platforms. In *NSDI*. USENIX Association, 2024.

[25] GeoCerts. Domain control validation by email verification method. https://www.geocerts.com/support/domain-control-validation-by-email-challenge-method, 2026.

[26] Github. Configuring a custom domain for your github pages site. https://docs.github.com/en/pages/configuring-a-custom-domain-for-your-github-pages-site.

[27] Gmail. Email sender guidelines. https://support.google.com/a/answer/81126, 2024.

[28] ip api. Ip geolocation api. https://ip-api.com/.

[29] E. Kenneally and D. Dittrich. The menlo report: Ethical principles guiding information and communication technology research, 2012.

[30] R. Li, C. Lu, B. Liu, Y. Zhang, G. Hong, H. Duan, Y. Lin, Q. Pan, M. Yang, and J. Shao. HADES attack: Understanding and evaluating manipulation risks of email blocklists. In *NDSS*. The Internet Society, 2025.

[31] Z. Lin, Z. Lin, R. Guo, J. Chen, M. Zhang, X. Liu, T. Yang, Z. Cao, and R. Deng. Detecting and measuring security implications of entangled domain verification in CDN. *CoRR*, abs/2409.01887, 2024.

[32] D. Liu, S. Hao, and H. Wang. All your DNS records point to us: Understanding the security threats of dangling DNS records. In *CCS*, pages 1414–1425. ACM, 2016.

[33] P. Mockapetris. Domain names - concepts and facilities. RFC 1034, November 1987.

[34] G. Moura, S. Castro, J. Heidemann, and W. Hardaker. Tsuname: exploiting misconfiguration and vulnerability to ddos DNS. In *IMC*, pages 398–418. ACM, 2021.

[35] public suffix list. https://publicsuffix.org/list/public_suffix_list.dat.

[36] D. McCarney J. Kasten R. Barnes, J. Hoffman-Andrews. Automatic certificate management environment (acme). RFC 8555, March 2019.

[37] Radix. report abuse. https://radix.website/report-abuse/, 2024.

[38] P. Wouters E. Nygren T. Wicinski S. Sahib, S. Huque. Domain control validation using dns. draft-ietf-dnsop-domain-verification-techniques-12, March 2026.

[39] Sectigo. How does the email challenge-response method work for domain control validation (dcv)? https://www.sectigo.com/knowledge-base/detail/email-challenge-response-for-domain-control-validation-dcv, 2026.

[40] Spamhaus. https://www.spamhaus.org/.

[41] M. Squarcina, M. Tempesta, L. Veronese, S. Calzavara, and M. Maffei. Can I take your subdomain? exploring same-site attacks in the modern web. In *USENIX Security Symposium*, pages 2917–2934, 2021.

[42] SSL.COM. What are the requirements for ssl.com ssl/tls certificate domain validation? https://www.ssl.com/faqs/ssl-dv-validation-requirements/, 2026.

[43] Tranco. Top 1m domains. https://tranco-list.eu.

[44] XYZ.COM. The xyz team. https://www.spamhaus.org/authors/the-xyz-team/, 2024.

[45] Yahoo. Sender requirements and recommendations. https://senders.yahooinc.com/best-practices, 2024.

[46] M. Zhang, X. Li, B. Liu, J. Lu, Y. Zhang, J. Chen, H. Duan, S. Hao, and X. Zheng. Detecting and measuring security risks of hosting-based dangling domains. *Proc. ACM Meas. Anal. Comput. Syst.*, 7(1):9:1–9:28, 2023.

[47] Zoho. Domain verification in zoho mail. https://www.zoho.com/mail/help/adminconsole/domain-verification.html.

A Domain Hosting Provider Filtering

To reduce the scope of manual testing, we use random TXT record templates to filter hosting providers. We observe that some random TXT records are specific to particular domain hosting services. Therefore, when filtering email and web hosting providers, we exclude TXT records configured on hosted domains that are unrelated to the target service, e.g., TXT records used for web hosting or file storage.

Table 7 lists 10 popular TXT record templates that are independent of email services and web services. After excluding irrelevant TXT record templates, we compute, for each provider, the proportion of hosted domains that configure random TXT record templates. The cumulative distribution of these proportions is shown in Figure 9. Finally, we retain only 116 candidate providers with proportions below 20% for manual testing.

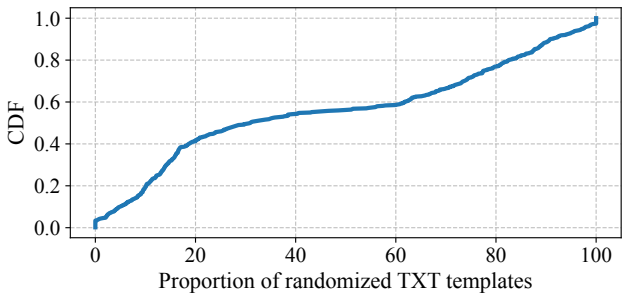

Figure 9: Proportion of domains hosted by providers that configure randomized TXT templates.

Table 7: TXT record templates and corresponding providers. Omits the random value part of the TXT record.

Provider	TXT template (partial)
Facebook	facebook-domain-verification
Apple	apple-domain-verification
Atlassian	atlassian-domain-verification
Brevo	sendinblue-site-verification
GlobalSign	_globalsign-domain-verification
Cisco	cisco-ci-domain-verification
Stripe	stripe-verification
Dropbox	dropbox-domain-verification
Adobe	adobe-idp-site-verification
1Password	1password-site-verification

B End-to-end Testing for Hosting Providers

We use ImprovMX and Render as examples to demonstrate the end-to-end testing process for email and web hosting providers, respectively. Throughout the entire process, we only used accounts and domains under our control for testing, ensuring that no third-party users were affected.

We first configured valid DOV challenge tokens for a normal domain (`normal.sec-mail.cloud`) to enable hosting on ImprovMX and Render. To host the domain on ImprovMX, `normal.sec-mail.cloud` configured MX records pointing to `mx1.improvmx.com` and `mx2.improvmx.com`, as well as a TXT record with the value "*v=spf1 include:spf.improvmx.com all*". To host the domain on Render, `normal.sec-mail.cloud` configured four A records pointing to 216.24.57.1. Figure 10 and Figure 11 show the hosting status of `normal.sec-mail.cloud` on ImprovMX and Render, respectively.

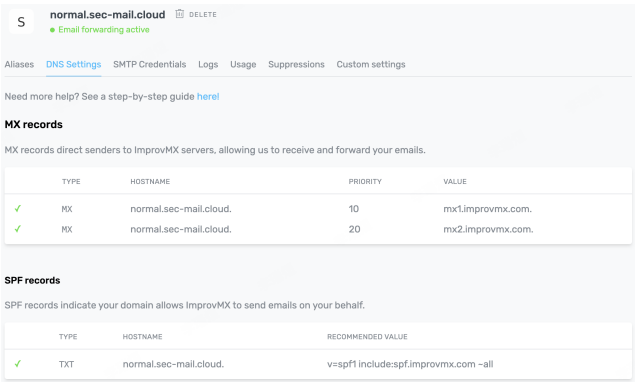

Figure 10: `normal.sec-mail.cloud` is normally hosted on ImprovMX.

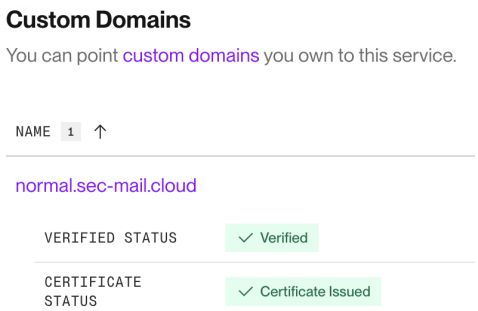

Figure 11: `normal.sec-mail.cloud` is normally hosted on Render.

We then set up a victim domain, `victim.sec-mail.xyz`, and configured its CNAME records to point to `normal.sec-mail.cloud`. Figure 12, Figure 13, and Figure 14 show the DNS query results of the MX, TXT, and A records of `victim.sec-mail.xyz`, respectively. We can observe that the victim domain already resolves to the correct DOV challenge tokens through the CNAME records.

Next, we register the victim domain through the web interfaces provided by ImprovMX and Render. As shown in Figure 15 and Figure 16, `victim.sec-mail.xyz` passed the providers' DOV and was successfully registered.

Finally, we examined whether the email service and web service of the victim domain functioned correctly. For the email service, we sent emails to our Gmail account through the mail forwarding interface provided by ImprovMX. We successfully received emails sent from the victim domain in the Gmail inbox. As shown in Figure 17, emails from `victim.sec-mail.xyz` passed Gmail's SPF verification. Therefore, we consider ImprovMX to be vulnerable to the AliasLeap attack. For the web service, we hosted a test webpage for the victim domain on Render. We then accessed the webpage of `victim.sec-mail.xyz` through Firefox browser. As shown in Figure 18, the victim domain returned the webpage deployed by us, and the certificate is valid. Therefore, we consider Render to be vulnerable to the AliasLeap attack.

```
~ dig victim.sec-mail.xyz MX

; <<>> DiG 9.10.6 <<>> victim.sec-mail.xyz MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52087
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;victim.sec-mail.xyz.           IN      MX

;; ANSWER SECTION:
victim.sec-mail.xyz.    600     IN      CNAME   normal.sec-mail.cloud.
normal.sec-mail.cloud.  56      IN      MX      10 mx1.improvmx.com.
normal.sec-mail.cloud.  56      IN      MX      20 mx2.improvmx.com.
```

Figure 12: MX records of `victim.sec-mail.xyz`.

```
~ dig victim.sec-mail.xyz TXT

; <<>> DiG 9.10.6 <<>> victim.sec-mail.xyz TXT
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27697
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;victim.sec-mail.xyz.           IN      TXT

;; ANSWER SECTION:
victim.sec-mail.xyz.    502     IN      CNAME   normal.sec-mail.cloud.
normal.sec-mail.cloud.  502     IN      TXT     "v=spf1 include:spf.improvmx.com
 ~all"
```

Figure 13: TXT records of `victim.sec-mail.xyz`.

```
~ dig victim.sec-mail.xyz A


; <<>> DiG 9.10.6 <<>> victim.sec-mail.xyz A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54386
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;victim.sec-mail.xyz.           IN      A

;; ANSWER SECTION:
victim.sec-mail.xyz.    600     IN      CNAME   normal.sec-mail.cloud.
normal.sec-mail.cloud.  600     IN      A       216.24.57.1
```

Figure 14: A records of `victim.sec-mail.xyz`.

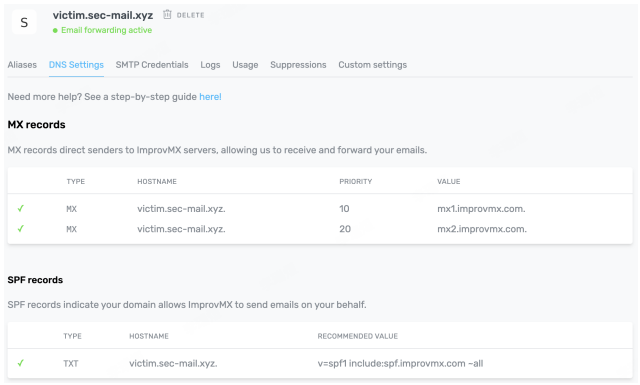

Figure 15: `victim.sec-mail.xyz` successfully registered on ImprovMX.

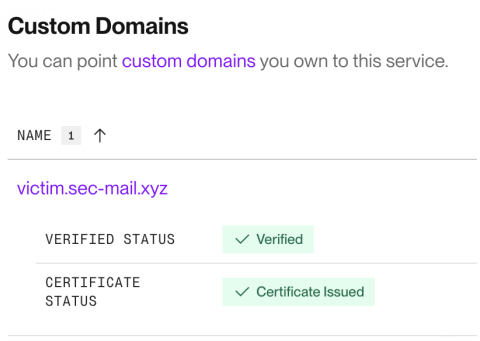

Figure 16: `victim.sec-mail.xyz` successfully registered on Render.

```
Received-SPF: pass (google.com: domain of bounces-
imx+d0b249c2ee09cc081f1f42599686e600ed9efea8@test.sec-mail.xyz designates
2001:41d0:340:e700::50 as permitted sender) client-
ip=2001:41d0:340:e700::50;
Authentication-Results: mx.google.com; dkim=pass header.i=@improvmx.com
header.s=dkimprovmx1 header.b=LnxYeNsv; arc=pass (i=1 spf=pass
spfdomain=improvmx.com dkim=pass dkdomain=improvmx.com dmarc=pass
fromdomain=improvmx.com); spf=pass (google.com: domain of bounces-
imx+d0b249c2ee09cc081f1f42599686e600ed9efea8@test.sec-mail.xyz designates
2001:41d0:340:e700::50 as permitted sender) smtp.mailfrom=bounces-
imx+d0b249c2ee09cc081f1f42599686e600ed9efea8@test.sec-mail.xyz; dmarc=pass
(p=QUARANTINE sp=REJECT dis=NONE) header.from=improvmx.com
```

Figure 17: Gmail validation results for the email from `victim.sec-mail.xyz`.

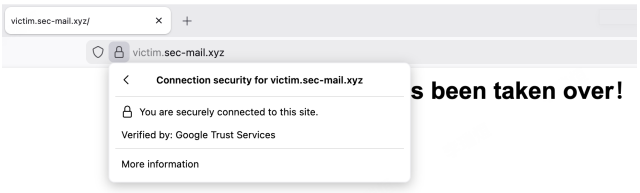

Figure 18: Access `victim.sec-mail.xyz` in Firefox browser.

Table 8: Challenge tokens and NDRs for vulnerable email hosting providers.

Email provider	Challenge token	NDR for non-existent user*	
		Hosted domain	Unhosted domain
Anon	**MX**: m**2.q**q.com **MX**: m**1.q**q.com **TXT**: v=spf1 include: spf.mail.q**q.com ~all	Mailbox unavailable or access denied [<code>= IP: <sender_ip>]. https://open.work.w**n.q**q.com/help2/pc/20057.	Bad address syntax. http://service.e**l.q**q.com/cgi-bin/help?subtype=1&&id=20022&&no=1000730
Alibaba Cloud	**MX**: mx1.qiye.aliyun.com **MX**: mx2.qiye.aliyun.com **MX**: mx3.qiye.aliyun.com **TXT**: v=spf1 include: spf.qiye.aliyun.com -all	Rcpt Ok	RCPT (<sender_address>) dosn't exist
Namecheap	**MX**: mx1.privateemail.com **MX**: mx2.privateemail.com **TXT**: v=spf1 include: spf.privateemail.com ~all	<rcpt_address>: Recipient address rejected: unverified address: Mailbox might be disabled, full, or may not exist on the server. Reason: JFE030050	<rcpt_address>: Relay access denied
ImprovMX	**MX**: mx1.improvmx.com **MX**: mx2.improvmx.com **TXT**: v=spf1 include: spf.improvmx.com ~all	SMTP error, RCPT TO: Host <sender_domain>(<rcpt_ip>) RCPT TO said 550 5.1.1 The email account you tried to reach does not exist. Please double-check the recipient's email address for typos or unnecessary spaces. - ImprovMX v2025.11.21	host <sender_domain>[<rcpt_ip>] said: 550 5.1.3 Relay not permitted. (#id-5.9.2) - ImprovMX v2025.11.21 (in reply to RCPT TO command)

* <code> denotes a provider-defined string; <rcpt_address> denotes the recipient's email address; <rcpt_ip> denotes the IP address of the recipient mail server; <sender_ip> denotes the IP address of the sender mail server; <sender_address> denotes the sender's email address; and <sender_domain> denotes the sender's domain.

Table 9: Challenge tokens and HTTP responses for vulnerable web hosting providers.

Web provider	Challenge token*	HTTP GET response	
		Hosted domain	Unhosted domain
GitHub	**A**: 185.199.108.153 **A**: 185.199.109.153 **A**: 185.199.110.153 **A**: 185.199.111.153 **CNAME**: <user>.github.io†	**Status**: 200 OK	**Status**: 404 Not Found **Title**: Site not found · GitHub Pages
Netlify	**A**: 75.2.60.5 **CNAME**: <project>.netlify.app	**Status**: 200 OK	**Status**: 404 Not Found
Render	**A**: 216.24.57.1 **CNAME**: <project>.onrender.com	**Status**: 200 OK	**Status**: 409 Conflict **Title**: DNS resolution error | <domain>| Cloudflare
Framer	**A**: 31.43.160.6 **A**: 31.43.161.6 **CNAME**: sites.framer.app	**Status**: 200 OK	ERR_SSL_PROTOCOL_ERROR§
Nuvemshop	**A**: 185.133.35.21 **A**: 185.133.35.22 **CNAME**: <project>.lojavirtualnuvem.com.br	**Status**: 200 OK	**Status**: 404 Not Found **Title**: 404: NOT_FOUND
Vercel	**A**: 216.198.79.1 **CNAME**: <project>.vercel-dns-017.com	**Status**: 200 OK	**Status**: 404 Not Found **Title**: 404: NOT_FOUND
Hostinger	**A**: 153.92.8.254	**Status**: 200 OK	**Status**: 403 Forbidden **Title**: 403 Forbidden

* Domains can complete the DOV of web providers by configuring A records or CNAME records.

† <user> represents the user account name; <project> represents the project name created by the customer; <domain> represents the domain requested by the user.

§ Requesting HTTP resources of unhosted domains to Framer will receive an SSL error. ERR_SSL_PROTOCOL_ERROR is an error code given by the Chrome browser.