Zombies in Alternate Realities: The Afterlife of Domain Names in DNS Integrations

Sulyab Thottungal Valapu
USC/Information Sciences Institute
Los Angeles, United States
sulyab.tv@usc.edu

John Heidemann
USC/Information Sciences Institute
Los Angeles, United States
johnh@isi.edu

Mattijs Jonker
University of Twente
Enschede, The Netherlands

Raffaele Sommese
University of Twente
Enschede, The Netherlands

Abstract

DNS integrations leverage the discovery, trust, and uniqueness of the global Domain Name System with a *linkage* to another naming ecosystem, so the DNS name can help identify resources such as a cryptocurrency wallet or software component. While DNS ownership is verified at linkage creation, many ecosystems do not track subsequent DNS changes. The result is *zombie linkages*, where the DNS ownership has expired or changed, but the mapping to the linked resource persists. We define a threat model for DNS integrations, identifying five classes of attacks that leverage or exploit zombie linkages. We measure zombie occurrence across three DNS integrations—Web PKI; ENS, a blockchain naming system; and Maven Central, a Java software repository. We show that zombies exist in every ecosystem, but at very different fractions—zombies make up roughly 3% of TLS certificates for new domains, 24% of ENS on-chain imports, and 15% of Maven Central namespaces. We evaluate how integration design choices affect outcomes, with validate-once integrations (ENS on-chain, Maven Central) accumulating long-lasting zombies, linkages with expiration (Web PKI) limiting damage, while integrations that validate on every use (ENS gasless) are zombie-free by design. We look for specific attacks, finding attacks actively available for exploitation in both Web PKI and Maven Central. Finally, we recommend steps to reduce zombie occurrence.

1 Introduction

DNS, the Internet's Domain Name System [38], provides a universal namespace for the Internet. The importance of the DNS has given rise to a rich ecosystem of registries that coordinate use of top-level domains (TLDs), and registrars that sell names to the public. Together, they allocate domain names to individual owners, giving each each owner authority over resources under that domain.

DNS integration is when a new, independent naming ecosystem links itself to DNS names [45]. They do so to leverage the fact that domain names in DNS are globally unique, human-friendly, and already widely recognized. The Web Public Key Infrastructure (Web PKI), one of the most widely deployed integrations, binds TLS certificates to DNS names to secure web communications [7]. Other integrations include the Ethereum Name Service (ENS), which lets users associate DNS names with cryptocurrency wallets [33]; Maven Central, which derives Java package namespaces from DNS names, and Bluesky, which supports using DNS names as social media handles [15].

For all these benefits, integrating with DNS comes with a cost: *DNS names change hands, and linked ecosystems must keep up.* Domains expire, get transferred, or are re-registered by different parties, and no mechanism exists for ecosystems to subscribe to such changes in DNS state. While some periodically re-validate the DNS linkage, not all do.

When integrations become desynchronized, linkages become zombies, creating new risks as the linkages outlive the DNS ownership epoch that created them (§3.3). Consequences of zombie linkages range from user confusion to exploitable security vulnerabilities (§3.4). If an adversary squats on a linked name backed by a DNS name that is later re-registered, they may trick users into sending cryptocurrency to the wrong party. If an adversary acquires an established DNS name that expired, they may take over established linked resources such as software packages and mislead users into installing malware.

We provide the first systematic examination of the security threats that emerge when DNS integrations fail to track DNS ownership changes. Prior work has studied synchronization challenges in individual integrations [29, 24], (see also §2). Different ecosystems take different approaches: proactively limiting integration lifetime with built-in expiration, reactively removing stale records on discovery, or neither. Our goal is to understand what defenses are used and how effective they are. As the number of DNS integrations grows, exploring these questions grows more urgent.

This paper makes four contributions: First, we define the problem and develop a threat model when DNS integrations become desynchronized (§3). Second, we measure the prevalence and persistence of exploitable zombie linkages across three diverse integrations (§4): Web PKI (web security), ENS (cryptocurrency), and Maven Central (software package repository), and report the results (§5). Third, we examine whether zombies translate into exploitable attack surface in practice (§6). Finally, drawing on these findings, we recommend steps to reduce zombie occurrence (§7).

We measure zombies across every ecosystem we study, finding sharply different rates, determined by ecosystem design (§5).

Draft (Under Submission), Los Angeles, CA, USA

validate-once integrations accumulate the highest fraction of zombies: 24% of ENS On-chain imports and 15% of Maven Central namespaces are zombies, with none ever cleared in ENS On-chain and all permanent in Maven Central. Web PKI's bounded certificate lifetimes keep the zombie fraction at 2.7% for newly created domains, and the rarity of revocation (4.3% zombie certificates revoked) suggests lower certificate lifetimes as the most effective defense. We find that zombie formation is driven by routine DNS name expiration in ENS and Maven Central, whereas in Web PKI we see extremely short-lived linkages consistent with abuse.

We look for attacks involving zombies that are actively available for exploitation (§6), and find over 192k Web PKI zombie certificates still served months after DNS name death, 7.3k of which continue to be served past DNS re-registration, and 214 Maven Central zombie namespaces with publishing activity after DNS re-registration. Our findings show that zombie linkages are actively available for domain impersonation and software supply chain attacks.

Ethical Considerations and Data Availability: This paper used a range of non-personal data (DNS zone files, RDAP queries, TLS certificates, Maven Central metadata, and Ethereum records), most of it available publicly. Our work uses no personally identifiable information that is not already public. We assess security threats, but as general types of attacks, so our work does not require coordinated disclosure. We discuss ethics in detail in §A.

This paper uses data from third parties and newly collected data. Data from third-parties (dns.coffee [12] and OpenINTEL [52, 48]) is used with their permission and must be obtained from them. Data we have collected and pointers to third-party data is listed on our website [51].

2 Related Work

To our knowledge, only one group has previously explored the general challenge of DNS integrations, but several groups have explored specific integrations in the specific ecosystems we study (domain re-registration, Web PKI, blockchain naming systems, software package management).

To our knowledge, only Verisign has studied DNS integrations as class of security risks [44, 23, 24]. They categorize DNS-to-ecosystem name imports as synchronized, not synchronized, or ambiguous and study ENS On-chain, Bluesky, and GitHub Organizations. Our work was inspired by their study, but goes further to describe the threat model, characterizing zombie lifetimes and formation patterns, and begin the search for evidence of exploitation.

Several studies have examined the security risks of expired and re-registered domains. Problems here include residual trust and its abuse in re-registered DNS domains [27], and residual traffic in re-registered domains [47]. Dropcatching is acquiring domains when they expire, either to auction, or to provide advertising or malware [37, 26]. These prior works focus on DNS-specific implications of re-registration, we extend this work to consider *external ecosystems* that integrate with DNS.

In Web PKI, Ma et al. study stale TLS certificates over a ten-year period, finding that infrastructure changes leave third parties with TLS private keys for domains they no longer control, suggesting shorter certificate lifetimes [29]. Our results complement theirs: we focus on zombie certificates caused by DNS ownership changes and measure how long they continue to be served after DNS name death and re-registration, also finding revocation unreliable as a defense.

In blockchain naming, researchers have studied ENS adoption and identify squatting and malicious content [53], and study dropcatching of ENS-native `.eth` names to misdirect funds [39]. These studies address ENS's own namespace; we expand on both of these attacks and consider DNS integration as a broader risk. Other work has examined TLD collisions between blockchain naming systems and DNS [41, 21], work orthogonal to ours.

In software supply chains, security researches identified DNS reallocation as a supply-chain risk for Maven with "MavenGate" [40]. We expand on and quantify this attack in §6.3. While they identify vulnerable namespaces from public domain availability, we use DNS registration history to identify namespaces that have *already* undergone ownership changes and exhibit post-re-registration publishing activity. Maven is just one software packaging system; Gu et al. study six ecosystems and identify twelve attack vectors [17]; but they do not consider DNS ownership changes as an attack class, our focus.

Overall, our work is first work to develop a unified threat model for DNS integrations, characterize zombie linkages across multiple ecosystems, consider how ecosystem design choices produce different zombie behavior, and identify attack patterns through analyzing the zombie lifecycle.

3 Problem Statement

Our goal is to understand how linkages created in non-DNS ecosystems on the basis of DNS ownership can outlive that ownership, and how that can lead to abuse.

3.1 How Ecosystems Integrate DNS

We use *ecosystem* to refer to any non-DNS system that maintains its own namespace. Table 1 shows the three ecosystems we study: Web PKI [7], ENS (Ethereum Name Service) [33], and Maven Central [31].

A *DNS integration* is a mechanism by which an ecosystem uses a DNS name as an identifier within its own namespace [45]. DNS names are attractive for this purpose because they are human-friendly, unlike native identifiers in some ecosystems, and because they ground new namespaces in the most widely deployed namespace on the Internet.

Each integration binds a *DNS name* to an identifier in the ecosystem, which we call the *linked name*. Linked names often support additional information or capability, which we call the *linked resource*. For example, in ENS, a DNS name is linked to an Ethereum wallet address (the linked name), and the linked resource is the contents of wallet itself.

The linkage from DNS to the ecosystem is established through a *linkage process* in which the ecosystem verifies that the requesting user controls the DNS name. The exact mechanism for this linkage process differs by ecosystem, but it typically involves a challenge to demonstrate control of the DNS name. For example, the ecosystem may require that the DNS name owner demonstrate control by creating a specific DNS TXT record, and then it verifies the presence of this record to establish the linkage.

Completing a linkage successfully creates a *linkage entry*: a persistent artifact in the ecosystem that maps a DNS name to a linked

Term	Web PKI	ENS	Maven Central
DNS name	`example.com`		
Linked name	Public key	Wallet address	`com.example` namespace
Linked resource	Authenticated website	Ethereum wallet (balance, txn history)	Packages published under the namespace
Resource controlled by	TLS private key holder	Wallet private key holder	Account holder that completes the linkage process
Linkage process	CA issues certificate after domain control validation	*On-chain:* User submits Ethereum transaction with DNSSEC proof to ENS smart contract *Gasless:* User turns on DNSSEC; sets up DNS TXT record	User proves DNS ownership via DNS TXT record
Linkage entry	Certificate	*On-chain:* Blockchain transaction *Gasless:* DNS TXT record	Database record
Linkage entry lives in	Web server, CT logs	*On-chain:* Ethereum blockchain *Gasless:* DNS	Maven Central's internal database

Table 1: Terms applied to each DNS integration we study.

name. In some ecosystems, such as Web PKI, linkage entries automatically expire after some included time-to-live. In others, such as ENS On-chain, linkage entries remain valid indefinitely until explicitly removed.

Table 1 relates each term to the three integrations we study.

3.2 Background About Studied Ecosystems

We study three ecosystems here, and identify several other ecosystems for future study.

Web PKI. Web PKI is the cryptographic infrastructure used to secure the web and other TLS-based communication [7]. Its linkage entries are X.509 certificates binding DNS names to public keys (linked name) via the certificate's Common Name and Subject Alternative Name fields. In the ecosystem, certificates are signed by other certificates in a hierarchy rooted in trusted Certificate Authorities provided with the operating system or web browser. We focus on domain-validated certificates, the most common, making up 95% of CT logs [4].

ENS. ENS uses Ethereum to map human-readable names to blockchain addresses [33]. While ENS has its own `.eth` names, we focus on its linking DNS names to Ethereum wallet addresses (linked name) and wallets (the linked resource).

ENS supports two linkage methods. Since 2018, *On-chain* linkages write an entry to the Ethereum blockchain, including a DNSSEC-proof with the `DNSRegistrar` smart contract. Since 2024, *Gasless* linkages no longer modify the blockchain, but instead validate a DNSSEC-signed TXT record on each use [16]. We describe both as *ENS*, adding *On-chain* and *Gasless* when needed.

Maven Central. Maven Central is a public repository of Java software components [31], used automatically by build tools such as Apache Maven [2]. Anyone can publish new components (the linked resource) after creating an account on the web portal and registering a namespace (the linked name) verified through a DNS TXT record.

Maven Central's components are immutable once published [49], although they occasionally remove malware [50]. DNS integration and package immutability results in attacks we study in §6.3.

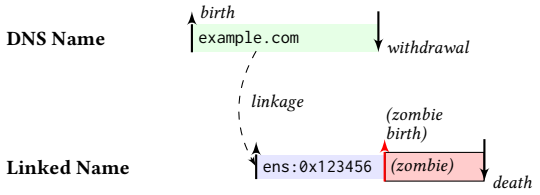


Figure 1: Timing during name integration via linkage.

3.3 How Linkages Become Zombies

DNS name ownership is not permanent. Domains expire, are transferred, or are re-registered by different parties. We define a *DNS ownership epoch* as a period of continuous control of a DNS name by a single user.

A linkage becomes a *zombie* when the DNS ownership epoch during which it was created has ended, but its linkage entry remains valid in the ecosystem. A zombie linkage continues to map a DNS name to the original linked name, an assertion that is no longer valid.

How long a zombie persists depends on two properties of the ecosystem's design: the issued lifetime of the linkage entry, and the ecosystem's revocation mechanism.

If a linkage entry is issued with a finite validity period, that period acts as an upper bound on zombie duration: a zombie TLS certificate cannot outlive its expiration date. Ecosystems that issue linkage entries without lifetime constraints lack this natural bound.

Within this bound, zombie duration depends on whether the ecosystem revokes zombies, and how quickly. Some ecosystems, such as ENS On-chain, have no revocation mechanism at all, leaving zombies to persist indefinitely. Others do revoke, but the delay between DNS ownership change and revocation determines the window of exposure.

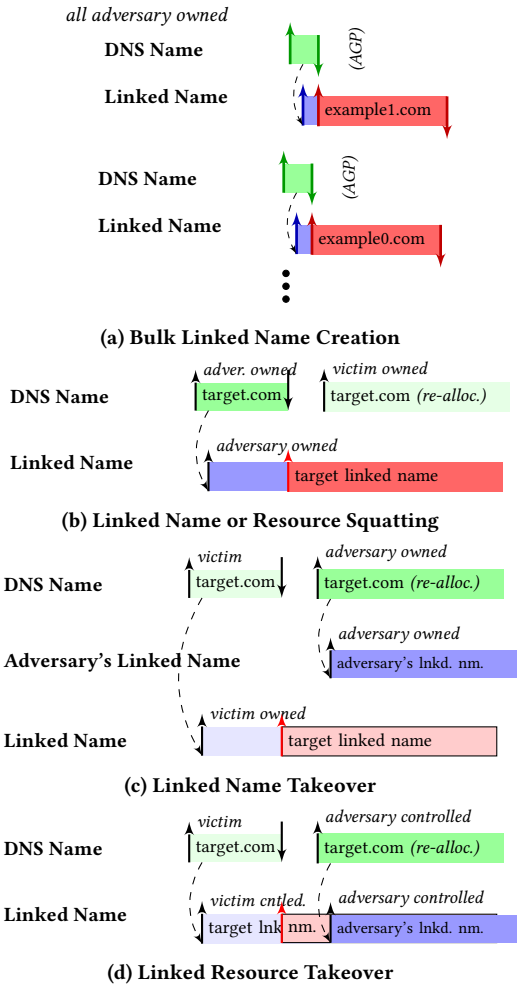


Figure 2: Different attacks on naming integration.

3.4 Attacks on DNS Integrations

Integration between DNS and other ecosystems raises the problem that they can fall out of synchronization, presenting new threats. We look at five types of attack, with three focusing on the DNS side: Bulk Linked Name Creation, Linked Name Squatting, and Linked Name Takeover. The two other attacks occur in the linked ecosystem: Linked Resource Squatting, and Linked Resource Takeover.

Squatting attacks are when an adversary acquires a *new* linked name or resource, anticipating that it will have value in the future. We generalize this term from domain squatting [5], to naming integration. *Takeover attacks* are when an adversary takes control of an existing linked name or resource with the hope of gaining value from it. This term follows from subdomain takeover attacks [36].

3.4.1 Bulk Linked Name Creation is the first type of squatting attack (Figure 2a). In this case the adversary acquires many DNS names at low cost. The adversary then immediately links them into another ecosystem, creating the linked name and resource, and then quickly deletes the DNS names. DNS names can be created at very low cost using *domain tasting* [9], where names discarded during Add Grace Period (AGP) [18] can be tried for as little as $0.10 USD per DNS name [8], plus any ecosystem-specific fees. Although ICANN limits the number of free AGP deletions per registrar per month [19], an adversary can distribute the attack across multiple registrars or over several months.

Once the adversary amasses linked names in bulk for cheap, they can be used in spamming campaigns, laundering data through a chain of names, or for other throw-away purposes.

The exact victim of this attack depends on how the linked names are used. For example, an adversary who retains unrevoked TLS certificates for DNS names that have since been suspended for abuse and who can direct victims to a server (such as through DNS cache poisoning [22, 28]) can serve content that browsers accept as authenticated. The adversary can then serve phishing or malware pages, or run trojan attacks on these victims.

3.4.2 Linked Name Squatting is when the adversary creates a DNS name they expect someone else will later desire, so the adversary can hold the linked name. If the ecosystem does not revoke the linkage entry when DNS name ownership ends, the result is a zombie linkage. There are two subcases of this attack depending on whether the DNS name is re-registered or not.

If the DNS name goes unclaimed (Figure 2a), as is typical in Bulk Linked Name Creation, the adversary exploits residual trust in the linked name.

In the second case (Figure 2b), the DNS name is re-registered by another party, but this new DNS name owner coexists with the zombie linked name held by the adversary. The adversary can then impersonate the new owner in the ecosystem without the new owner's knowledge. The victim is then either the new DNS name owner, who does not get customers, or customers that unwittingly use the wrong linked name.

3.4.3 Linked Resource Squatting occurs when the linked resource is not independent from the linked name, so a squatting attack extends to the linked resource. (It looks the same as linked name squatting, Figure 2b.) The adversary who first created the DNS name and linkage retains access to the linked resource even after losing control of the DNS name.

For example, Maven Central derives namespace access from domain ownership but does not revoke it when ownership ends. The original owner can continue to publish packages under the namespace after the domain expires. A subsequent owner who claims the same namespace inherits these packages and cannot remove them due to Maven's immutability guarantee [49], leaving the namespace polluted with packages from a previous ownership epoch.

3.4.4 Linked Name Takeover attacks occur when an adversary acquires a DNS name after its previous owner loses control (Figure 2c), such as by *dropcatching* it at the moment it is deleted from the registry after expiration [37]. The adversary then creates a new linkage pointing to a linked name under their control. The adversary captures value that has accumulated around the DNS name: existing web links, bookmarks, payment references, or social media recognition. The adversary can then impersonate the previous owner: victims who follow links using the DNS name, expecting to reach the original owner's Ethereum payment address, now reach the adversary instead.

The DNS ecosystem provides safeguards such as renewal grace periods and expiration notices, but these only help owners who intend to keep their domains. Once a domain lapses, dropcatching is common [37].

The victim in a linked name takeover is the original creator of the DNS and linked names, and potentially new users attempting to access that name.

3.4.5 Linked Resource Takeover attacks occur when the linked resource is not independent from the linked name, so controlling the linked name suffices to control the linked resource and everything accumulated under it (Figure 2d).

The consequences of a Linked Resource Takeover can be severe. For example, Maven Central derives namespace access entirely from domain ownership: whoever proves control of `example.com` gains publishing rights to `com.example`. An adversary who registers an expired domain gains publishing rights to a namespace that may contain years of trusted packages with many downstream dependents (the dark blue portion of linked name in Figure 2d). The adversary can then publish malicious new versions of those packages, indistinguishable from legitimate updates. An analogous attack recently succeeded in the WordPress ecosystem, where an adversary purchased a widely used plugin company with over 20k active installations, and published new versions that included a backdoor [14, 11]. Linked Resource Takeover enables the same outcome without requiring the original owner's cooperation. This attack is particularly dangerous with modern build systems that automatically download the latest version of packages from the Internet. Victims are any applications that build with compromised libraries, subject to remote code execution or exfiltration of secrets.

3.5 Automatic Defenses Against Attacks

Two factors help defend against some types of attacks: *automatic zombie revocation* and *linked resource independence*.

Some systems *automatically* revoke linkages. A simple method is to *time out* the linked name, requiring the linkage to be regularly renewed. TLS Certificates carry validity periods, and those have moved from years to months or even days [43, 34]. Timeouts provide a fail-safe window to safety (assuming users do not override timeout validation), and limit the impact of Bulk Creation. However, they still leave the linked name vulnerable for some time. Unfortunately, of the systems we consider, only Web PKI has timeouts.

Other systems may periodically confirm and invalidate zombie linkages. However, if confirmation is done periodically, it is not immediate, and the delay between the end of a DNS ownership epoch and actual revocation creates a window during which these attacks remain possible. None of the systems we consider periodically validates linkages.

Independence of Linked Resources from the linkage prevents Linked Resource Squatting and Linked Resource Takeover attacks, since any new linkage will create a new linked resource. Independent resources cannot directly accessed by the adversary, however, new users using the DNS and its linkage to find the linked name and resource may find the adversary's new linked resource instead of the victim's old resource. If the user accepts this resource as valid, the could inadvertently interact with the adversary, perhaps installing their malware or sending funds to their crypto wallet.

The goal of our study is to understand how these aspects affect different attacks across different ecosystems.

4 Methodology

Our methodology follows four steps to evaluate how well an ecosystem integrates with DNS: (1) identify all linkages in the ecosystem, (2) determine the birth and death of each, (3) map each linkage's birth to a DNS ownership epoch, and (4) determine when that epoch began and ended.

Because inferring DNS ownership epochs (steps 3 and 4) is common to all integrations, we describe our approach for these steps first, then detail each integration's methodology individually.

4.1 Inferring DNS Ownership Epochs

Determining whether a linkage is a zombie requires identifying the DNS ownership epoch during which the linkage was created, and whether that epoch has since ended. The ideal source of this information would be the full registration history of the DNS name, but registries do not make it publicly available.

4.1.1 Data Sources. We instead combine three complementary sources: current registration status from RDAP, historical TLD zone delegation data, and active DNS resolution scans. Each varies in authority and coverage, so combining them requires care.

Our approach builds on three principles.

First, registration is a prerequisite for delegation, which in turn is a prerequisite for resolution. Observing a domain as delegated or resolvable on a given day confirms it was registered on that day (ignoring transient states), though not which ownership epoch it belongs to.

Second, RDAP provides authoritative ownership epoch boundaries. A positive RDAP response returns the start of the current ownership epoch, though not whether prior epochs existed. A negative RDAP response confirms that no registration is currently active.

Third, sustained gaps in delegation data typically indicate domain expiry and deletion. For gTLDs, a domain typically passes through auto-renew grace, redemption, and pending delete before its delegation is removed from the parent zone, a process taking about 80 days [1]. A gap of 80 days or more in delegation is therefore strong evidence that the domain was fully deleted, though not authoritative: ccTLDs follow different lifecycles, and delegation lapses can occur without a change in registration. When delegation data is unavailable, we fall back on gaps in active scan observations using the same threshold, though this is a weaker signal.

4.1.2 Algorithm. Our algorithm combines these three principles to infer DNS ownership epochs. We describe the algorithm briefly here; the full algorithm is in the appendix (Algorithm 1) for space.

The algorithm infers registration intervals for a given domain from three inputs: daily TLD zone delegation observations (Z), daily active scan observations (S), and RDAP observations (R). Two tunable parameters control sensitivity. A delegation gap threshold (t) filters transient gaps in zone delegation, and a grace window (g) tolerates small misalignments between RDAP-reported and delegation-derived interval boundaries.

The algorithm operates in four phases.

In the first phase, the algorithm builds a unified bitset from daily zone delegation and active scan data, assigning 1 to each day a domain was observed and 0 otherwise.

In the second phase, candidate registration intervals are extracted as runs of consecutive 1s in the bitset. These intervals are initially *open*, meaning either end may be extended or merged with adjacent intervals in subsequent phases.

In the third phase, RDAP data refines the candidate intervals. An authoritative registration date can *close* the start of an interval, or split it if the date falls mid-interval, indicating a re-registration. An authoritative negative response can prevent the merger of two adjacent intervals.

In the fourth phase, adjacent intervals are merged if the gap between them is less than t days and no RDAP evidence prevents it. We use $t = 80$ days to align with the duration of the typical gTLD domain expiration process.

The intervals that remain after all four phases are output as inferred registration intervals.

Once the DNS ownership intervals are inferred, classifying a linkage as a zombie is straightforward: find the ownership interval active when the linkage was created and check whether it has since ended. The same intervals yield additional metrics, such as zombie duration and domain age at the time of linkage creation.

4.1.3 Specific Datasets. Our active scan data consists of daily NS and SOA queries for each DNS name corresponding to a linkage in the three ecosystems we study.

We source historical zone delegation data from dns.coffee [12]. This data spans 1291 TLDs from 2021 onward, though coverage varies by TLD.

Our RDAP data consists of 149M successful queries to 113M unique domains since 2023. Most of these queries target newly registered domains observed in CT logs, but we query RDAP for every domain in our study at least once.

4.1.4 Algorithm Limitations. Strictly speaking, our algorithm infers *registration* epochs and not *ownership* epochs, so zombies caused by within-registration ownership changes go undetected. A domain that is sold or transferred without re-registration changes ownership without resetting its registration date, making such transitions invisible to our data sources. Detecting registrant changes through RDAP is impractical because registrant fields are heavily redacted post-GDPR, and frequently obscured by privacy proxy services.

Our 80-day gap threshold is deliberately conservative and may under-count zombies, particularly domains that are dropcatched immediately after expiration. For such cases we rely entirely on RDAP to detect the ownership change via a new registration date. We prefer under-counting to over-counting, since classifying a live linkage as zombie is more harmful to our analysis than a missed zombie.

Finally, our algorithm operates at day granularity, so events within the same day are indistinguishable. This is a deliberate choice matching the resolution of our data: zone files are published daily and our active scans run once per day.

These algorithmic limitations compound with the dataset limitations that we discuss next.

4.1.5 Dataset Limitations. Each of our three data sources has limitations in coverage and accuracy.

RDAP, our most authoritative source, has the most uneven coverage. Some TLDs lack RDAP servers entirely (such as `.ee`), others return responses that omit registration dates (such as `.de`), and rate limiting causes query failures even for TLDs that do support RDAP.

Zone delegation coverage is limited primarily to gTLDs participating in ICANN's CZDS program [20], leaving most ccTLDs uncovered. Even for covered TLDs, zone file update frequency varies, introducing timing uncertainty in when a delegation change becomes visible. Zone files may also miss short-lived domains that are registered and removed between successive snapshots [48].

Since our active scan coverage begins at the start of our data collection period, we have no visibility into earlier events. Additionally, scan failures may reflect network-layer issues such as timeouts rather than changes in registration or delegation status.

We are in the process of obtaining ground-truth registration history from a major TLD to evaluate the accuracy of our inferred ownership epochs in light of these limitations.

4.2 Measuring Linkages in Ecosystems

Next, we describe how we identify linkages in each ecosystem, and how we determine their birth and death.

Web PKI. The linkages we study in the Web PKI are domain-validated TLS certificates of *newly registered* domains. We focus on newly registered domains because they are more likely to be short-lived [48], and therefore more likely to produce zombies.

To collect certificates, we identify Fully Qualified Domain Names (FQDNs) under newly registered domains by monitoring Certificate Transparency (CT) logs via OpenINTEL's ZoneStream service [48], and contact each via TLS. We exclude invalid certificates from our dataset by validating the presented certificate chain against Mozilla's Root CA bundle.

We define the birth of a certificate as its `notBefore` timestamp, and its death as the earlier of its `notAfter` timestamp or its revocation timestamp, if revoked. We check for revocation first using CRLs to minimize traffic, and fall back to OCSP queries when CRLs are unavailable.

Over a period of six months (2025-10-01 to 2026-04-15), we collect 52.2 million certificates across 23.6 million newly registered domains. This subset represents 2.7% of 1.9 billion certificates issued during our study period [4].

ENS. Our ENS data collection differs between On-chain and Gasless linkages.

We identify On-chain linkages by inspecting Ethereum blockchain records. Since the On-chain linkage process involves calls to ENS's `DNSRegistrar` smart contract, we collect all transactions to this contract using Etherscan [10], and decode them using the contract's ABI. Successful linkages emit a `Claim` event, which we use to extract the list of linkages. We gather 1,882 linkages since On-chain's inception in 2018.

We define the birth of an On-chain linkage as the timestamp of the block containing its `Claim` event, and its death as the timestamp of the *next* `Claim` event (if any) for the same DNS name. On-chain linkages do not expire and cannot be revoked; the only way to invalidate one is to overwrite it by repeating the linkage process.

Unlike On-chain linkages, no master list of Gasless linkages exists by design, so we discover them through active DNS scans. We search OpenINTEL's forward DNS dataset [52] for TXT records

matching the format required by ENS Gasless [16]. Since these scans cover only DNS names found in zone files and CT logs, our count (see Figure 4) is a lower bound. Because Gasless linkages are validated using DNSSEC on each use, they cannot produce zombies by design.

Maven Central. The linkages we study in the Maven Central are software namespaces.

We identify namespaces and their published package versions by crawling a Maven Central repository mirror [32]. We restrict our analysis to DNS-based namespaces using reverse-DNS convention (now mandatory), identifying 19.1M package versions across 31,853 namespaces.

Because namespace creation dates are not publicly available, we use the date of the first published package version as the birth of the namespace. Due to Maven Central's immutability policy, namespaces do not expire, so we treat them as having no death.

5 Characterizing Zombies

We next characterize zombies using historic and new data. We look at how often linkages become zombies (§5.1), and causes of creation (§5.2). We consider zombie longevity and revocation (§5.3). Due to space limitations, we report on time-to-linkage as a risk indicator in an appendix §C.

5.1 What Fraction of Linkages are Zombies?

To establish the scale of zombies, we first identify the fraction of linkages that are active zombies at a given point in time.

Figure 3 shows the fraction of linkages that are zombies (dashed red, right axis) along with counts of all (solid blue) and zombie (dashed blue) linkages in each ecosystem. We also show our 80-day threshold (the gray dotted line, from §4.1); delegation and scan gaps after this point are too short to trigger zombie detection, though RDAP-based detection remains possible.

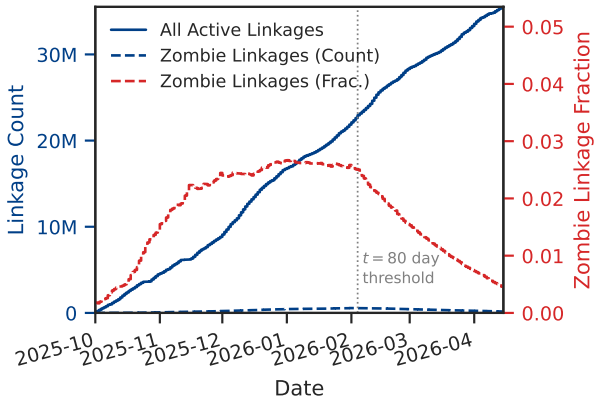
Most counts of active linkages grow over the measurement period, even though we track both births and deaths. The reason for this growth varies by ecosystem. For PKI, observed linkage births (52.2M total) outpace deaths (17.1M total); for ENS On-chain and Maven, linkages never die (see §5.3).

Web PKI. Figure 3a shows zombie fraction and count among 52.2M TLS certificates of newly created DNS names observed over six months (2025-10-01 to 2026-04-15), with a peak of 35.5M active certificates on a given day.

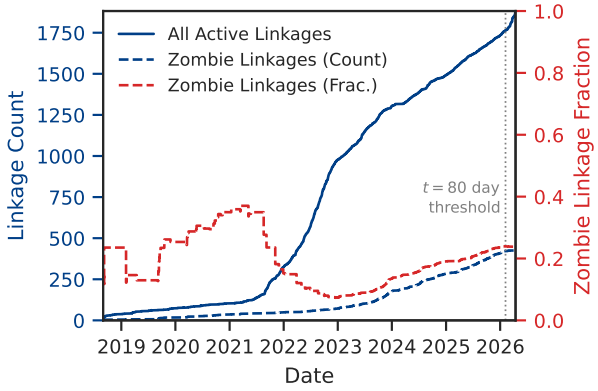
We see an increasing number of zombies over time, with a relatively small fraction of zombies (around 2.7%).

We expect the fraction of zombie TLS certificates to remain steady over time, assuming the creation rate is stable and old zombies expire. The apparent rise and fall of the zombie fraction is a measurement artifact, from newly registered DNS names having to die for zombies to form, and our 80-day classification threshold. We discuss zombie formation and cleanup in §5.2 and §5.3.

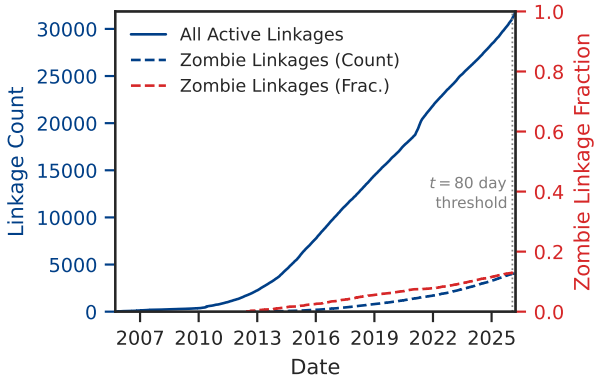
We see the zombie fraction is stable around 2.7% from about 2025-11 to 2026-02, away from measurement transients. Because newly created domains are disproportionately short-lived [48], this rate likely overstates the Web PKI-wide zombie fraction. At the same time, since we track only a few domains (25M, while `.com` alone has over 150 million [12]), the absolute number of zombies across the ecosystem is likely substantially larger. Evaluating the



(a) Web PKI



(b) ENS (On-chain)



(c) Maven Central

Figure 3: Zombie fraction over time across integrations. Each panel shows active linkage count (solid blue, left axis), zombie linkage count (dashed blue, left axis), and zombie linkage fraction (dashed red, right axis).

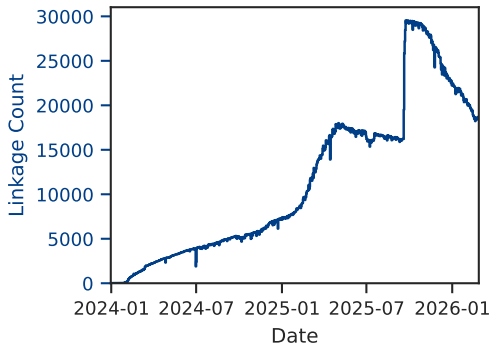


Figure 4: Active ENS Gasless linkages over time.

full Web PKI population would require pairing CT log data with registration histories across all TLDs.

ENS (On-chain). Since ENS On-chain linkages are valid indefinitely (until overwritten), we expect the number of zombies to grow. Figure 3b confirms this hypothesis. The number of zombies grow steadily over our 7 years of data. The fraction of zombies varies from 7 to 30%, dipping when there are many new registrations. Of the 1,882 active ENS On-chain linkages as of April 2026, 425 (23.8%) are zombies.

Since 2023, the fraction of zombie fraction has been tending up as older linkages age. Its nadir was 7% in late 2022, with a surge in new ENS On-chain linkages. Since ENS linkages have no expiration, we expect this growth to continue; in §5.3 we show that zombies have never been cleared.

For comparison, Figure 4 shows the growth of ENS Gasless linkages, which share the ENS ecosystem but are *zombie-proof by design* because they are validated on each use. Since late 2025, over 10k Gasless linkages have lost their backing TXT record and are no longer resolvable; had these been On-chain, every one would persist as a zombie indefinitely.

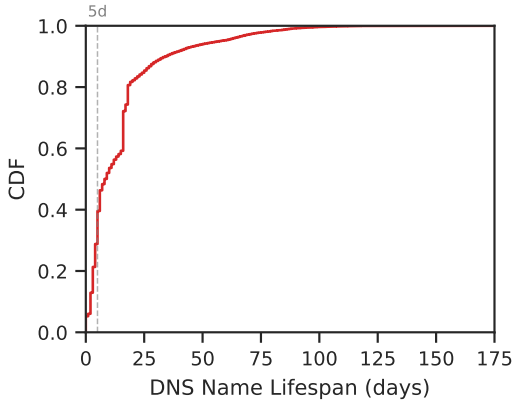
Maven Central. Like ENS On-chain, Maven Central verifies namespace claims only once, so we expect Maven Central zombies to accumulate over time.

Figure 3c confirms this hypothesis: of 31,853 namespace claims as of March 2026, 4,842 (15.2%) are zombies, and both the absolute count and fraction seem to grow over time.

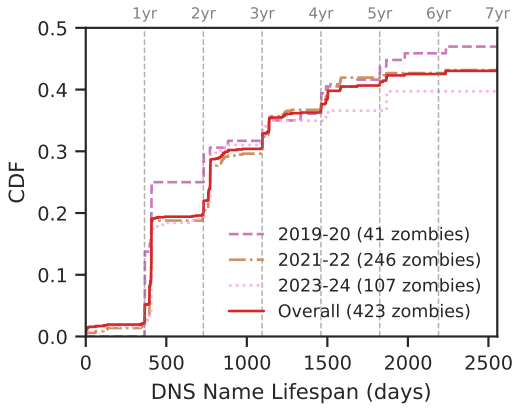
Maven's fraction of zombies is growing (15% after 20 years), but slower than ENS On-chain (24% after 7 years). Both systems validate-once, but we suggest that software publishers place more value in continuity than individuals experimenting with cryptocurrency. However, Maven Central and ENS On-chain show that validate-once systems show much higher zombie fractions than systems with expiration (Web PKI).

5.2 What Causes DNS Names to Go Away?

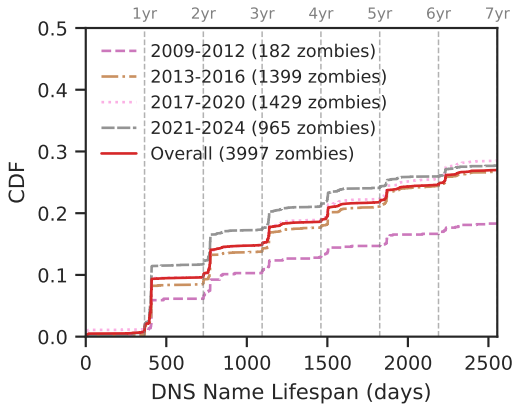
Zombies are created when the DNS name goes away, so what causes DNS names to go away? DNS names go away for four reasons. Rarely, they are taken away in response to abuse or trademark disputes. For domain tasting [9], DNS names are allocated conditionally and then removed by the registrant during a short Add Grace Period (typically 5 days). Users occasionally voluntarily close



(a) Web PKI



(b) ENS (On-chain)



(c) Maven Central

Figure 5: Distribution of DNS name lifespans across integrations. Colored curves show per-cohort trends; the solid red curve shows the overall population. Vertical gray dotted lines mark 5 days (AGP) for Web PKI, and years since registration for ENS On-chain and Maven Central.

a DNS name. Most often, DNS names automatically expire after their registration period ends and they are not renewed.

We next look at DNS lifetimes for two reasons: very short lifetimes suggest Bulk Linked Name Creation (Figure 2a), one of our attacks. Second, understanding why zombies are created helps us evaluate whether zombies are due to changes in user choices (people are creating more zombies over time), or due to natural DNS aging breaking the linkage.

Web PKI. Since we have only six months of DNS creation in our Web PKI observation, we cannot observe zombie formation due to annual DNS expiry. We can, however, assess early removals.

Figure 5a shows the distribution of DNS name lifetimes for 675k Web PKI zombies. The dashed gray line marks ICANN's five-day Add Grace Period (AGP).

We see two modes in this graph: 5 days and about 16 days, and a tail out to 100 days. We discuss each case next.

About two-fifths (39.6%) of zombies die within five days, inside the Add Grace Period (AGP). These zombies suggest domain tasting and Bulk Linked Name Registration. An alternative is that these names are removed involuntarily by registrars or registries in response to malicious activity [1], so our data establishes 40% as an upper bound on Bulk Linked Name Registration.

The second mode at 16 days falls outside the AGP, suggesting at least 20% are removed due to abuse.

The upper limits of observed DNS name lifespan follow from our methodology. All zombie-producing DNS names die within six months because a name must die to produce a zombie, and our observation period is six months.

Since nearly 4 in 5 zombie-producing names die within 18 days, their certificates remain valid for the bulk of their issued lifetime, leaving plenty of time to exploit zombie certificates. We examine certificate revocation in §5.3.

ENS (On-chain) and Maven Central. Since our ENS On-chain and Maven Central data spans years from each ecosystem's inception, we examine them together.

We show DNS name lifespans for linkages in ENS On-chain (Figure 5b) and Maven Central (Figure 5c), with each curve showing a cohort of linkages with about the same creation date (two-year groups for ENS On-chain, four-year groups for Maven Central). The solid red line shows the overall trend across all cohorts. To enable fair inter-cohort comparison, we include all linkages and use the Kaplan-Meier estimator [25] to account for right-censored observations. The CDFs therefore do not reach 1.0.

In both ENS On-chain and Maven Central, zombie formation is clustered just past a multiple of years since DNS name registration. The largest drop is always in the first year, with progressively smaller steps in subsequent years, consistent with a fixed annual non-renewal probability applied to a shrinking surviving population. Domain renewal patterns show first-year renewal rates are the lowest, and renewal rates increase in subsequent years [46]. Regardless, this pattern strongly suggests that most zombies result from natural DNS expiration, rather than conscious DNS termination.

ENS On-chain shows little cohort effect, with per-year curves overlapping closely. The curves start to diverge past year 5, but this spread reflects sparsity of data due to single-digit events post 5 years rather than a meaningful shift.

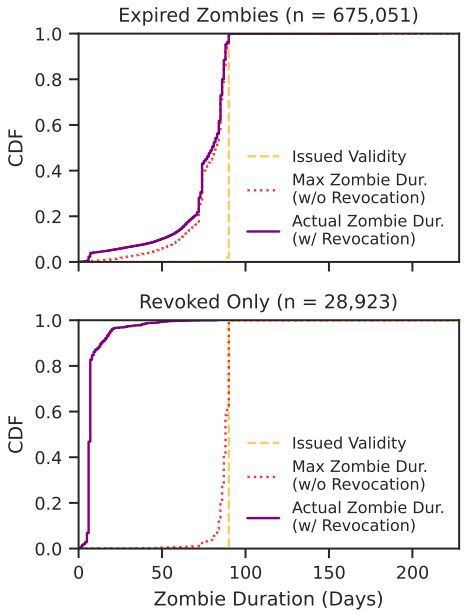


Figure 6: Distribution of zombie duration for expired Web PKI certificates. The top panel shows all zombies; the bottom panel shows only those whose certificates were revoked.

Maven Central, however, shows a clear cohort effect, with the earliest cohort (2009–2012, the lowest dashed line) showing much longer-lived DNS names than others. We suggest that early adopters of Maven Central were institutions or early Java users, both with long-held domains and important, long-lived software packages. Later cohorts are more similar with each other, with less stable DNS names.

While DNS name expiry is the dominant driver of zombie formation in both ecosystems, we observe a few early DNS removals. In ENS On-chain, 10 of 423 zombies (2.4%) were formed by early removal, five of them within the AGP. In Maven Central, 40 of 3,998 zombies (1.0%) died within a year of DNS name registration, eight within the AGP.

These trends suggest two takeaways: First, re-validation will be most efficient if it is done annually after DNS birth. Second, zombies occur relative to DNS name age, so increasing fractions of zombies seen in §5.1 reflect the age of each ecosystem and *not* a change in user behavior.

5.3 How Long do Zombie Linkages Persist?

Given zombies exist in all studied ecosystems (§5.1), we now ask: how long do zombies persist after their linkage is broken? The longer a zombie remains active, the longer an adversary has to carry out the attacks described in §3.4.

Web PKI. The top panel of Figure 6 shows the distribution of zombie duration for zombie TLS certificates that expired or were revoked during our observation period. We exclude certificates still valid at the observation cutoff because they may yet be revoked before their expiry.

We find 675k certificates that meet this criteria, and the lifetime is dominated by Let's Encrypt's 90-day certificates. This mode is in part because our six-month-long observation is too brief to capture both zombie birth and death for certificates with a 1-year expiration. We are in the process of getting multi-year historic data that will see these zombies.

Figure 6 shows the certificate validity period at issuance (dashed orange), which caps zombie lifetime and has almost all probability mass at 90 days. The two curves show remaining validity after DNS name expiration (dotted red), the maximum zombie duration assuming no revocation, and actual observed zombie duration (solid purple, left-most), which includes both expiration and revocation.

Most zombies persist for nearly the full validity period of 90 days: the median is 75 days, with 40% lasting 80 days or more, and only 10% under 45 days. Our measurement precision is ±24 h, which underestimates zombie lifetime, so this trend of "near 90 days" is very strong.

Shorter expiration times would shrink zombie duration. Let's Encrypt plans to reduce default certificate expiration to 45 days by 2027 [35], and the CA/Browser Forum mandates a maximum validity of 47 days by 2029 [43]. Both changes would cut median zombie lifetime, and therefore the number of active zombies, roughly in half.

Revocation, provides little benefit because it is almost never used in practice, as shown by the similarity of curves with and without revocation (solid and dotted).

Web PKI Revocation: Although rarely used, is revocation important when it is used? We consider the subset of zombie certificates that were revoked.

The bottom panel of Figure 6 examines the lifetime of the 28.9k (4.3% of all 675k) zombies that were revoked. DNS name expiration rarely results in certificate revocation.

The revoked cohort show a large reduction in zombie lifetime: the median reduction is 82 days (92%), eliminating nearly the entire zombie window. Thus when revocations occur, they occur very soon after certificate creation.

Nearly all zombies (96%), however, are never revoked, suggesting revocation today is not an effective defense against zombies. While encouraging revocation might reduce vulnerability, this data confirms the benefits of shorter expiration times to reduce the window of zombie vulnerability.

ENS (On-chain). ENS On-chain zombies persist until the new DNS owner actively reclaims the name with a new linkage. Our data shows that *none* of the 1,882 ENS On-chain zombies have been reclaimed, meaning *every* zombie that has ever formed remains active today. The median ENS On-chain zombie today is 1.9 years old.

Short of the new DNS name owner repeating the linkage process, there is currently no mechanism to revoke an ENS On-chain zombie. An NSEC3-based method for invalidating linkages via negative proofs once existed, but was removed in 2022 [3] without ever being used in practice.

This permanence of zombies makes ENS On-chain particularly susceptible to Linked Name Squatting. An adversary's zombie linked name remains resolvable indefinitely at no ongoing cost, even if the DNS name is later re-registered by a new owner. We discuss this attack further in §6.1.

	Web PKI	ENS (On-ch.)	Maven Central
Bulk Linked Name Creation	✓ §6.2	○	○
Linked Name Squatting	✓ §6.2	✓	↑
Linked Resource Squatting	×	×	✓ §6.3
Linked Name Takeover	?	○	↑
Linked Resource Takeover	×	×	✓ §6.3

Table 2: Attack indicators across ecosystems. Design properties: × prevented by design, ↑ escalates to resource attack. Data findings: ○ no evidence of attack, ? insufficient data to determine, ✓ actively available for exploitation.

Maven Central. Maven Central zombies are permanent.

Unlike ENS On-chain, where a new DNS name owner may overwrite a zombie by repeating the linkage process, Maven Central's immutability guarantee prevents any modification or removal of published package versions [49]. (In practice, however, Maven Central has removed packages containing malicious code in the past [50].) A zombie namespace and all packages published under it therefore persist indefinitely, regardless of DNS ownership changes.

Our data shows that, as of April 2026, Maven Central has 4,842 zombie namespaces, with a median zombie age of 3.5 years.

Because Maven Central namespaces are immutable and publishing rights rest with the linked name owner, zombie permanence enables both Linked Resource Squatting and Takeover. A former DNS name owner can continue publishing under a zombie namespace (Linked Resource Squatting) because Maven Central does not re-verify DNS ownership. A new DNS name owner can take over the namespace and publish compromised versions (Linked Resource Takeover), though namespace takeovers go through manual review. We discuss these attacks further in §6.3.

6 Identifying Attacks Involving Zombies

In §5 we show zombies *exist*, but are they exploited? We next use data for each ecosystem to evaluate if the attacks described in §3.4 are *actively available for exploitation*. Although fully confirming exploitation requires data we do not have, we can show indicators consistent with an attack, and for Maven Central (§6.3), resource modification.

6.1 Attack Evaluation

Table 2 summarizes our findings so far.

In the best case, *attacks are prevented by design* in some ecosystems (marked with × and green) in Table 2. Resource attacks (Linked Resource Squatting and Linked Resource Takeover) are prevented in Web PKI and ENS On-chain because their linked resources require a separate cryptographic key, independent of the linked name (see Table 1).

Our data shows *no evidence of several attacks that are otherwise viable* by design (labeled ○ and cyan). We find no evidence of Bulk Linked Name Creation in ENS On-chain or Maven Central, where very few zombies form within the Add Grace Period (5 of 423 for ENS On-chain, 8 of 3,998 for Maven Central; see Figure 5). No ENS On-chain zombie has ever been overwritten by a new DNS

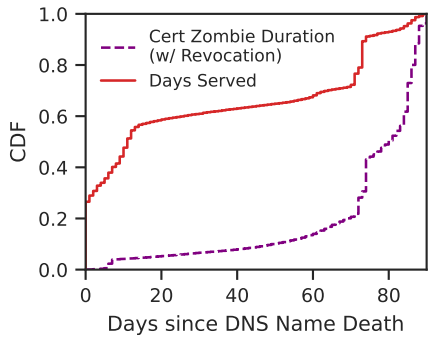


Figure 7: Duration Web PKI zombie certificates are served after DNS death (solid red line).

Category	Count	(%)	
Maven namespaces	31 853	100.0	
Live namespaces	27 011	84.8	
Zombie namesp. (unk. zombie start)	789	2.5	
Zombie namesp. (known zombie start)	4 053	12.7	*100.0*
No changes while zombie	3 506	11.0	*86.5*
New versions published while zombie	547	1.7	*13.5*
DNS Name not re-registered	257	0.8	*6.3*
DNS Name re-registered	290	0.9	*7.2*
No changes after re-registration	76	0.2	*1.9*
New versions after re-registration	214	0.7	*5.3*

Table 3: Zombies in Maven Central namespace.

name owner (§5.3), indicating that Linked Name Takeover has not occurred.

One attack–ecosystem pair lacks sufficient data to draw a conclusion (marked ?). Our Web PKI dataset contains only newly registered DNS names, so we cannot observe Linked Name Takeovers which require re-registration of an existing name. We will review this attack when we have more data.

The remaining attacks are both viable by design and actively available for exploitation based on our data (marked ✓). In Web PKI, nearly 300k newly created DNS names with TLS certificates die within the Add Grace Period (§5.2), leaving zombie certificates available for Bulk Linked Name Creation. Linked Name Squatting in Web PKI is supported by 580k active zombie certificates that persist for most of their validity period (§5.3). In ENS On-chain, 456 zombies persist indefinitely with none ever cleared (§5.3), each available for Linked Name Squatting. In Maven Central, 4,842 zombie namespace claims persist permanently due to immutability guarantees, leaving them available for both Linked Resource Squatting and Linked Resource Takeover given Maven Central's lack of resource independence.

Although our data shows evidence of zombie that could be exploited, that presence consistent with an attack does not imply *actual* exploitation. We now look for stronger evidence of possible attacks in Web PKI and Maven Central. We also examined ENS On-chain for evidence of exploitation, but the blockchain does not contain enough data for us to confirm. Linked Name Squatting requires that funds are sent to the zombie linked name, but Ethereum transactions do not record whether funds were sent via the linked name (potentially misdirected) or via the raw wallet address (legitimate).

6.2 Attacks on Web PKI

In §5.1 we show that there are many zombies in the PKI ecosystem. We next go two steps further in the attack chain to show that these zombies are active on servers, and that zombie certificates sometimes overlap with new domain owners. Evaluation of server contents for malware is future work.

6.2.1 Zombie PKI Certificates are Served: First we show that zombie certificates are actually served, posing a viable threat.

To determine how long zombie certificates are served, we contact each TLS server daily at its last known IP address, checking if it presents the now-zombie certificate. We study servers until the certificate expires or is revoked, since validating clients will then reject the certificate. Our measurements provide a lower-bound on this vulnerability since we contact only the last known IP, missing cases where the certificate is served at a new address.

Figure 7 shows the results for 675k zombie certificates during our six-month observation period. We compare how long certificates are actively served by a TLS server after DNS name death (the top, solid red line). We also show zombie duration, from DNS name death to certificate expiry or revocation (the lower dashed purple line), as in Figure 6, an upper-bound on days served.

While a quarter of zombie certificates are never served once they are zombies, about one-third (32%, or 215k) are served for 60 days or more. An adversary who can direct victims to such a server can impersonate the domain with a certificate that browsers accept as valid.

6.2.2 Served Zombie PKI Certificates Overlap with New DNS Owners: Next we show that some zombie certificates that are served overlap with new DNS owners. This overlap is a potential example of malicious Linked Name Squatting in the Web PKI ecosystem.

Of 675k zombie certificates, the validity period of 33k zombies (4.9% of all zombies) overlaps with a new DNS name registration, and 7.3k of these (1.1% of all zombies) are served past re-registration. Both figures account for revocation.

Figure 9 (in appendix) shows how long these 7.3k zombie certificates continue to be served after their DNS names are re-registered: median certificate is served for 49 days past re-registration, a window during which an adversary can impersonate the new DNS name owner.

Revocation is one defense against zombie certificate misuse. We therefore compare revocation rates for zombie certificates whose DNS names are re-registered (33k) with those that are not (642k). We see *much* higher revocation rates for zombies whose DNS names are re-registered (11.6%) than for those that are not (3.9%). This 3× higher fraction of revocations suggests that concern about zombie certificates results in action, yet the fact that 8 in 9 are never revoked and only go away on expiration suggests that considerable risk remains.

6.3 Attacks on Maven Central

To study Linked Resource Squatting and Takeover in Maven Central, we examine whether new package versions are published after a namespace becomes a zombie in Table 3. This table breaks down 31,853 Maven Central namespaces based on their zombie status and latest publishing activity.

Of 4,053 zombie namespaces with known zombie start dates, *547 (13.5%) publish new package versions after zombie birth.* Any application that depends on these packages may unknowingly pull code published by someone who no longer controls the DNS name.

Of those 547 namespaces, 290 have since had their DNS names re-registered, of which 214 continue publishing new versions after re-registration. These 214 namespaces pose the most risk: applications that pull new versions from them may unknowingly incorporate compromised code, exposing their users to attacks. Recall that our algorithm detects re-registrations via RDAP registration date changes or delegation gaps of 80 days or more, so same-owner registrar transfers are not counted (§4.1).

Confirming whether these zombie namespaces are active attack vectors will require comparing package signing keys and contents across versions. Nevertheless, our findings demonstrate that resource attacks on Maven Central are feasible. Following MavenGate [40], Sonatype stated in 2024 that accounts associated with expired domains were disabled [13], but our data shows zombie namespaces with publishing activity as recent as 2026, indicating this defense may be inadequate. It also remains to be seen whether manual validation as a defense against re-claiming existing namespaces [13] can withstand well-resourced adversaries.

7 Recommendations for DNS Integrations

Our findings suggest several design principles for current and future DNS integrations.

The most effective defense against zombies is to *validate linkage status on any use* of a linked name. ENS Gasless demonstrates this approach: it validates linkages on-demand using DNSSEC, bounding the zombie window to the DNS cache duration (typically minutes).

Limiting linkage lifetime with a time-to-live is a great second defense, because it guarantees a limit on the duration which a name can be exploited. The Web PKI ecosystem does this well, with every certificate including a time-to-live. Web PKI has experimented with many techniques (revocation lists [6], OCSP [42] are two), but shortening certificate validity has been very effective, prompting shortening certificate lifetime from years to a proposed 47 days [43], with Let's Encrypt recently trying 6 days [34].

We recommend Resource Independence, where the linked resource is protected by an ecosystem-specific mechanism. Resource Independence provides defense in depth, separating control of the linked resource with additional mechanisms (private key for PKI and ENS, for example). Of the three integrations we study, only Maven Central is vulnerable to resource tampering (injection of new versions) through linked name control alone.

Finally, our findings provide heuristics for prioritizing active re-validation to detect and mitigate zombies, for ecosystems that do not have other mitigations. Zombie formation concentrates around the first few registration anniversaries (§5.2), and linkages made soon after DNS registration are more likely to become zombies (§C).

8 Conclusion

We study synchronization of DNS integration across three ecosystems and find that zombie linkages exist in all of them, but at sharply different rates driven by ecosystem design choices. Our analysis shows that zombie linkages are actively available for exploitation in Web PKI and Maven Central. Based on these findings, we recommend that new integrations treat DNS ownership change as a first-class event by validating linkages on every use and bounding linkage lifetimes. As more ecosystems anchor trust in DNS names, the synchronization problem we characterize here will only grow in scope and consequence.

Acknowledgments

We sincerely thank Ian Foster of dns.coffee for providing access to historical TLD zone delegation data. We thank Wes Hardaker (Google) for starting this work when he was at USC/ISI, and for his technical comments on its evolution. We thank Swapneel Sheth and Andrew Kaizer (Verisign), and Rick Wilhelm and Suzanne Woolf (PIR), for their technical comments on early versions of this research. This research was partially supported by gifts from Verisign and Public Interest Registry (PIR). John Heidemann's work was support in part by NSF awards CNS-2319409, OAC-2530698, and 2453092.

References

[1] Antonia Affinito, Raffaele Sommese, Gautam Akiwate, Stefan Savage, K. C. Claffy, Geoffrey M. Voelker, Alessio Botta, and Mattijs Jonker. 2022. Domain Name Lifetimes: Baseline and Threats. In *6th Network Traffic Measurement and Analysis Conference, TMA 2022, Enschede, The Netherlands, June 27-30, 2022.* Roya Ensafi, Andra Lutu, Anna Sperotto, and Roland van Rijswijk-Deij, (Eds.) IFIP. https://dl.ifip.org/db/conf/tma/tma2022/tma2022-paper32.pdf.

[2] Apache Software Foundation. [n. d.] Introduction to repositories. Apache Maven Project. Retrieved Apr. 29, 2026 from https://maven.apache.org/guides/introduction/introduction-to-repositories.html.

[3] Arachnid. 2022. Commit 859622b: made DNSSEC oracle pure. GitHub. (May 29, 2022). Retrieved Apr. 30, 2026 from https://github.com/ensdomains/ens-contracts/commit/859622b3b9f7990bdc1d7bf8d005f64a53a0ffc3.

[4] Cloudflare. [n. d.] Certificate transparency. Cloudflare Radar. Retrieved Apr. 29, 2026 from https://radar.cloudflare.com/certificate-transparency?dateStart=2026-01-01&dateEnd=2026-04-15.

[5] Cloudflare. [n. d.] What is cybersquatting? | Domain squatting. Retrieved Mar. 31, 2026 from https://www.cloudflare.com/learning/dns/what-is-cybersquatting/.

[6] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. 2008. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280. Internet Request For Comments. doi:http://dx.doi.org/10.17487/RFC5280.

[7] T. Dierks and E. Rescorla. 2008. The Transport Layer Security (TLS) Protocol, Version 1.2. RFC 5246. (proposed standard). Internet Request For Comments, (Jan. 2008). ftp://ftp.rfc-editor.org/in-notes/rfc5246.txt.

[8] Dynadot. [n. d.] Grace Deletion - Domain Tasting and Returns. Retrieved Mar. 24, 2026 from https://www.dynadot.com/domain/grace-deletion.

[9] Dynadot. 2025. What is grace deletion aka domain tasting? Dynadot Help. (Feb. 10, 2025). Retrieved Mar. 19, 2026 from https://www.dynadot.com/help/question/what-is-grace-deletion.

[10] Etherscan. [n. d.] Etherscan API documentation. Retrieved Feb. 19, 2026 from https://docs.etherscan.io/introduction.

[11] Sead Fadilpašić. 2026. WordPress websites under attack — expert report says dozens of plugins hijacked to target thousands of sites. TechRadar. (Apr. 15, 2026). Retrieved Apr. 24, 2026 from https://www.techradar.com/pro/security/wordpress-websites-under-attack-expert-report-says-dozens-of-plugins-hijacked-to-target-thousands-of-sites.

[12] Ian Foster. 2026. dns.coffee: The DNS Historical Zone Database. dns.coffee, (2026). Retrieved Apr. 16, 2026 from https://dns.coffee.

[13] Brian Fox. 2024. Sonatype's ongoing commitment to Maven Central. Sonatype. (Jan. 19, 2024). Retrieved Apr. 28, 2026 from https://web.archive.org/web/20240122171220/https://www.sonatype.com/sonatypes-ongoing-commitment-to-maven-central.

[14] Austin Ginder. 2026. Someone bought 30 WordPress plugins and planted a backdoor in all of them. anchor.host. (Apr. 9, 2026). Retrieved Apr. 23, 2026

from https://anchor.host/someone-bought-30-wordpress-plugins-and-planted-a-backdoor-in-all-of-them/.

[15] Jay Graber. 2023. Domain Names as Handles in Bluesky. Bluesky. (Mar. 6, 2023). Retrieved Mar. 10, 2026 from https://bsky.social/about/blog/3-6-2023-domain-names-as-handles-in-bluesky.

[16] gregskril.eth. 2024. Gasless DNSSEC on Mainnet. ENS Blog. (Jan. 29, 2024). Retrieved Dec. 15, 2025 from https://ens.domains/blog/post/gasless-dnssec.

[17] Yacong Gu, Lingyun Ying, Yingyuan Pu, Xiao Hu, Huajun Chai, Ruimin Wang, Xing Gao, and Haixin Duan. 2023. Investigating package related security threats in software registries. In *2023 IEEE Symposium on Security and Privacy (SP)*. 2023 IEEE Symposium on Security and Privacy (SP). ISSN: 2375-1207. (May 2023), 1578–1595. doi:10.1109/SP46215.2023.10179332.

[18] ICANN. 2008. AGP (add grace period) limits policy. (Dec. 17, 2008). Retrieved Apr. 23, 2026 from https://www.icann.org/en/contracted-parties/consensus-policies/add-grace-period-limits-policy/agp-add-grace-period-limits-policy-17-12-2008-en.

[19] ICANN. 2009. The end of domain tasting | status report on AGP measures. (Dec. 8, 2009). Retrieved Apr. 23, 2026 from https://www.icann.org/en/contracted-parties/consensus-policies/add-grace-period-limits-policy/the-end-of-domain-tasting-status-report-on-agp-measures-12-08-2009-en.

[20] Internet Corporation for Assigned Names and Numbers. [n. d.] Centralized Zone Data Service. ICANN. Retrieved Apr. 17, 2026 from https://czds.icann.org/.

[21] Daiki Ito, Yuta Takata, Hiroshi Kumagai, and Masaki Kamizono. 2024. Investigations of Top-Level Domain Name Collisions in Blockchain Naming Services. In *Proceedings of the ACM Web Conference 2024* (WWW '24). Association for Computing Machinery, New York, NY, USA, (May 13, 2024), 2926–2935. ISBN: 979-8-4007-0171-9. doi:10.1145/3589334.3645459.

[22] Jian Jiang, Jinjin Liang, Kang Li, Jun Li, Haixin Duan, and Jianping Wu. 2012. Ghost Domain Names: Revoked Yet Still Resolvable. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS '12)*. (Feb. 8, 2012).

[23] Andrew Kaizer, Will Naciri, and Swapneel Sheth. 2024. Poster: Synchronization Concerns of DNS Integrations. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (CCS '24). Association for Computing Machinery, New York, NY, USA, (Dec. 9, 2024), 4982–4984. ISBN: 979-8-4007-0636-3. doi:10.1145/3658644.3691415.

[24] Andrew Kaizer, William Naciri, and Swapneel Sheth. 2025. Synchronization Concerns of DNS Integrations. (July 26, 2025). doi:10.36227/techrxiv.175355215.52122651/v1.

[25] E. L. Kaplan and Paul Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53, 282, (June 1, 1958), 457–481. doi:10.1080/01621459.1958.10501452.

[26] Tobias Lauinger, Abdelberi Chaabane, Ahmet Salih Buyukkayhan, Kaan Onarlioglu, and William Robertson. 2017. Game of Registrars: An Empirical Analysis of Post-Expiration Domain Name Takeovers. In 26th USENIX Security Symposium (USENIX Security 17), 865–880. ISBN: 978-1-931971-40-9. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/lauinger.

[27] Chaz Lever, Robert Walls, Yacin Nadji, David Dagon, Patrick McDaniel, and Manos Antonakakis. 2016. Domain-Z: 28 Registrations Later Measuring the Exploitation of Residual Trust in Domains. In *2016 IEEE Symposium on Security and Privacy (SP)*. 2016 IEEE Symposium on Security and Privacy (SP). ISSN: 2375-1207. (May 2016), 691–706. doi:10.1109/SP.2016.47.

[28] Xiang Li, Baojun Liu, Xuesong Bai, Mingming Zhang, Qifan Zhang, Zhou Li, Haixin Duan, and Qi Li. 2023. Ghost Domain Reloaded: Vulnerable Links in Domain Name Delegation and Revocation. In *Proceedings 2023 Network and Distributed System Security Symposium*. Network and Distributed System Security Symposium. Internet Society, San Diego, CA, USA. ISBN: 978-1-891562-83-9. doi:10.14722/ndss.2023.23005.

[29] Zane Ma, Aaron Faulkenberry, Thomas Papastergiou, Zakir Durumeric, Michael D. Bailey, Angelos D. Keromytis, Fabian Monrose, and Manos Antonakakis. 2023. Stale TLS Certificates: Investigating Precarious Third-Party Access to Valid TLS Keys. In *Proceedings of the 2023 ACM on Internet Measurement Conference* (IMC '23). Association for Computing Machinery, New York, NY, USA, (Oct. 24, 2023), 222–235. ISBN: 979-8-4007-0382-9. doi:10.1145/3618257.3624802.

[30] H. B. Mann and D. R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18, 1, (Mar. 1947), 50–60. doi:10.1214/aoms/1177730491.

[31] Maven Central. [n. d.] About. The Central Repository Documentation. Retrieved Apr. 29, 2026 from https://central.sonatype.org/pages/about/.

[32] Maven Central. [n. d.] Central repository. Retrieved Apr. 29, 2026 from https://repo1.maven.org/maven2/.

[33] mcdee. 2018. ENSIP-6: DNS-in-ENS. ENS Docs. (June 26, 2018). Retrieved Mar. 10, 2026 from https://docs.ens.domains/ensip/6.

[34] Matthew McPherrin. 2026. 6-day and IP address certificates are generally available. Let's Encrypt. (Jan. 15, 2026). Retrieved Apr. 26, 2026 from https://letsencrypt.org/2026/01/15/6day-and-ip-general-availability.html.

[35] Matthew McPherrin. 2025. Decreasing Certificate Lifetimes to 45 Days. Let's Encrypt. (Dec. 2, 2025). Retrieved Dec. 9, 2025 from https://letsencrypt.org/2025/12/02/from-90-to-45.html.

[36] MDN Contributors. 2025. Subdomain takeover - Security. MDN Web Docs. (Sept. 10, 2025). Retrieved Mar. 31, 2026 from https://developer.mozilla.org/en-US/docs/Web/Security/Attacks/Subdomain_takeover.

[37] Najmeh Miramirkhani, Timothy Barron, Michael Ferdman, and Nick Nikiforakis. 2018. Panning for gold.com: understanding the dynamics of domain dropcatching. In *Proceedings of the 2018 World Wide Web Conference* (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, (Apr. 23, 2018), 257–266. ISBN: 978-1-4503-5639-8. doi:10.1145/3178876.3186092.

[38] P. Mockapetris. 1983. Domain names—concepts and facilities. RFC 882. Internet Request For Comments, (Nov. 1983). https://www.rfc-editor.org/rfc/rfc882.txt.

[39] Muhammad Muzammil, Zhengyu Wu, Aruna Balasubramanian, and Nick Nikiforakis. 2024. Panning for gold.eth: Understanding and Analyzing ENS Domain Dropcatching. In *Proceedings of the 2024 ACM on Internet Measurement Conference* (IMC '24). Association for Computing Machinery, New York, NY, USA, (Nov. 4, 2024), 731–738. ISBN: 979-8-4007-0592-2. doi:10.1145/3646547.3689009.

[40] Oversecured. 2024. Introducing MavenGate: a supply chain attack method for Java and Android applications. Oversecured Blog. (Jan. 17, 2024). Retrieved Apr. 26, 2026 from https://oversecured.com/blog/introducing-mavengate-a-supply-chain-attack-method-for-java-and-android-applications.

[41] Audrey Randall, Wes Hardaker, Geoffrey M. Voelker, Stefan Savage, and Aaron Schulman. 2022. The Challenges of Blockchain-Based Naming Systems for Malware Defenders. In *2022 APWG Symposium on Electronic Crime Research (eCrime)*. (Nov. 2022), 1–14. doi:10.1109/eCrime57793.2022.10142131.

[42] S. Santesson, M. Meyers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. 2013. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol—OCSP. RFC 6960. Internet Request For Comments, (June 2013). doi:http://dx.doi.org/10.17487/RFC690.

[43] Server Cert WG. 2026. Latest baseline requirements. CA/Browser Forum. Version: 2.2.2. (Jan. 12, 2026). Retrieved Feb. 10, 2026 from https://cabforum.org/working-groups/server/baseline-requirements/requirements/.

[44] Swapneel Sheth and Andrew Kaizer. 2023. Call for Collaboration: DNS Integrations. In *Proceedings of the Applied Networking Research Workshop*. ANRW '23: Applied Networking Research Workshop. ACM, San Francisco CA USA, (July 24, 2023), 15–17. ISBN: 979-8-4007-0274-7. doi:10.1145/3606464.3606471.

[45] Swapneel Sheth, Andrew Kaizer, Bryan Newbold, and N. Johnson. 2025. Integration of DNS Domain Names into Application Environments: Motivations and Considerations. Internet Draft draft-ietf-dnsop-integration-01. Internet Engineering Task Force, (Oct. 7, 2025). Retrieved Jan. 27, 2026 from https://datatracker.ietf.org/doc/draft-ietf-dnsop-integration-01.

[46] SIDN. 2025. Why registrants don't renew their domain names? .nl domain name renewal patterns. SIDN News and Blogs. (Aug. 11, 2025). Retrieved Apr. 19, 2026 from https://www.sidn.nl/en/news-and-blogs/what-influences-customers-to-renew.

[47] Johnny So, Najmeh Miramirkhani, Michael Ferdman, and Nick Nikiforakis. 2022. Domains Do Change Their Spots: Quantifying Potential Abuse of Residual Trust. In *2022 IEEE Symposium on Security and Privacy (SP)*. 2022 IEEE Symposium on Security and Privacy (SP). (May 2022), 2130–2144. doi:10.1109/SP46214.2022.9833609.

[48] Raffaele Sommese, Gautam Akiwate, Antonia Affinito, Moritz Muller, Mattijs Jonker, and kc claffy. 2024. DarkDNS: Revisiting the Value of Rapid Zone Update. In *Proceedings of the 2024 ACM on Internet Measurement Conference*. IMC '24: ACM Internet Measurement Conference. ACM, Madrid Spain, (Nov. 4, 2024), 454–461. ISBN: 979-8-4007-0592-2. doi:10.1145/3646547.3689021.

[49] Sonatype. [n. d.] Immutability. Maven Central Repository Documentation. Retrieved Apr. 28, 2026 from https://central.sonatype.org/publish/requirements/immutability/.

[50] Sonatype Security Research Team. 2021. Sonatype Stops Software Supply Chain Attack Aimed at the Java Developer Community. (Jan. 13, 2021). Retrieved Apr. 28, 2026 from https://www.sonatype.com/blog/malware-removed-from-maven-central.

[51] Sulyab Thottingal Valapu and John Heidemann. 2026. Data about DNS integrations. website https://ant.isi.edu/datasets/dnsintegration/. (May 2026). https://ant.isi.edu/datasets/dnsintegration/.

[52] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. 2016. A high-performance, scalable infrastructure for large-scale active DNS measurements. *IEEE Journal on Selected Areas in Communications*, 34, 6, (June 2016), 1877–1888. doi:10.1109/JSAC.2016.2558918.

[53] Pengcheng Xia, Haoyu Wang, Zhou Yu, Xinyu Liu, Xiapu Luo, Guoai Xu, and Gareth Tyson. 2022. Challenges in decentralized name management: the case of ENS. In *Proceedings of the 22nd ACM Internet Measurement Conference* (IMC '22). Association for Computing Machinery, New York, NY, USA, (Oct. 25, 2022), 65–82. ISBN: 978-1-4503-9259-4. doi:10.1145/3517745.3561469.

A Ethics

In evaluating the ethics of our work, we considered data sources and personal data, and vulnerability assessment and the potential need for disclosure. Overall, we do not believe our work poses any ethical challenges given the steps we have taken.

This paper used a range of non-personal data including DNS zone files, RDAP queries, TLS certificates, Maven Central repository metadata [32], and Ethereum blockchain records. Much of this data is available publicly. Some data was aggregated by third-parties (dns.coffee [12] and OpenINTEL [52, 48]) and used with their permission. We will provide data that we collected for this paper available at no cost upon publication, and provide pointers to third-party aggregations that we used.

Our work uses no personally identifiable information that is not already public (for example, in the Ethereum blockchain). We therefore do not consider it to pose any privacy risks to individuals.

When querying public web endpoints (such as RDAP servers and Maven Central), we rate-limited our requests, respected throttling signals, and cached results wherever possible to avoid redundant queries.

Our work assesses security threats and identifies new classes of attacks on DNS integrations. We discuss the security implications of the design decisions of different ecosystems. However, we do not identify specific vulnerabilities that might require software patches or coordinated disclosure.

B Algorithm to Infer DNS Ownership Epochs

Algorithm 1 shows the algorithm we use to infer DNS ownership epochs. We discuss this algorithm in §4.1.

C Does Time from Registration to Linkage Predict Zombie Risk?

We study whether linkages that eventually become zombies are created sooner after DNS name registration than the general population. Users who link an established DNS name may do so at any point in its lifetime, but users who register a name specifically to create a linkage would do so soon after registration. If the two populations differ, registration-to-linkage timing may be used as a practical signal for identifying zombie-prone linkages early.

We limit this study to ENS On-chain and Maven Central. Our Web PKI dataset contains only newly created domains, so the registration-to-linkage gap is short by construction.

Figure 8 plots the years from DNS name registration to linkage creation for ENS On-chain (left) and Maven Central (right), for the full population (top) and zombies only (bottom). The figure visually supports our hypothesis: the fraction of linkages created immediately after registration is nearly double for zombies compared to the general population (55% vs. 30% for ENS On-chain, 30% vs. 15% for Maven Central).

To confirm this difference statistically, we apply a Mann-Whitney U test [30] comparing registration-to-linkage duration between zombie and non-zombie linkages. A significant result ($p < 0.05$) means that zombie linkages are systematically created sooner after registration than non-zombie linkages.

Both ecosystems confirm the hypothesis ($p < 10^{-63}$ for ENS On-chain, $p \approx 0$ for Maven Central). Zombie linkages are created far sooner after registration, with median gaps of 4 days (zombie) compared to 292 days (non-zombie) for ENS On-chain and 123 days (zombie) compared to 1,169 days (non-zombie) for Maven Central.

The gap between DNS name registration and linkage creation is therefore a practical predictor of zombie risk, and ecosystems could use it as a heuristic to periodically re-validate zombie-prone linkages.

D Days Web PKI Zombies are Served Past DNS Re-registration

Figure 9 shows the distribution of duration 7.3k Web PKI zombie certificates are served after their DNS names are re-registered. We discuss this figure in §6.2.

Input: TLD zone delegation observations Z
Active scan observations S
RDAP observations R
Delegation gap threshold t (default: 80 days)
Grace window g (default: 2 days)
Output: Inferred registration intervals $\mathcal{I}$

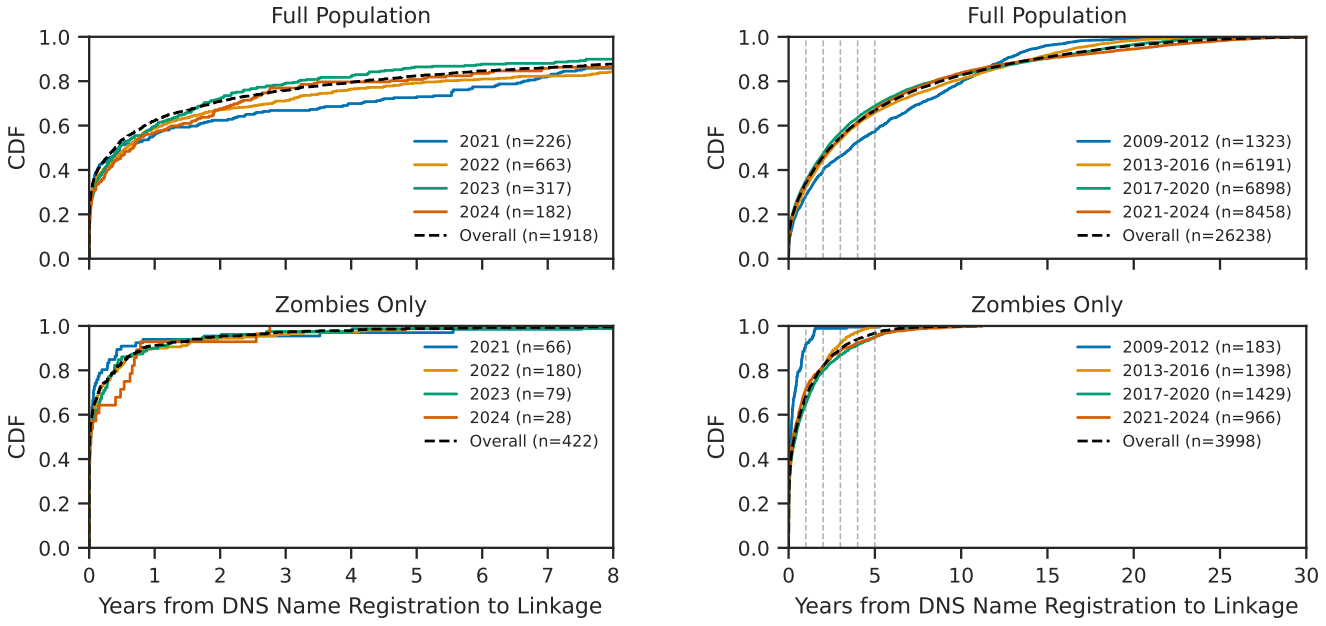
```
/* Phase 1: Bitset of delegation & scan */
```
$B_z \leftarrow \textsc{DayObsBitset}(Z)$; $B_s \leftarrow \textsc{DayObsBitset}(S)$;
$B \leftarrow B_z \vee B_s$;

```
/* Phase 2: Extract intervals */
```
$\mathcal{I} \leftarrow \textsc{ExtractRuns}(B)$; `// all ends open`

```
/* Phase 3: RDAP refinement */
```
$R^+ \leftarrow$ positive observations in R, keeping latest $o.time$ per $o.reg$;
$R^- \leftarrow$ negative observations in R;
foreach $o \in R^+$ *in order of* $o.reg$ **do**
  $I_{reg} \leftarrow$ interval containing $o.reg$, else create new;
  $I_{obs} \leftarrow$ interval containing $o.time$, else create new;
  if $o.reg - I_{reg}.start \leq g$ **then**
    make $I_{reg}.start$ closed;
  else
    split I_{reg} at $o.reg$ with new closed start;
  end
  foreach $I \in [I_{reg}, \text{prev}(I_{obs})]$ **do**
    $I.mergeNext \leftarrow$ True;
  end
end
foreach $o \in R^-$ *in order of* $o.time$ **do**
  if $o.time$ *is not within* g *days of any interval* **then**
    make start of next interval closed (if any);
  end
end

```
/* Phase 4: Merge */
```
foreach *adjacent* $(I_i, I_{i+1}) \in \mathcal{I}$ **do**
  if $I_{i+1}.start$ *is not closed* ***and*** $(I_{i+1}.start - I_i.end < t$ ***or*** $I_i.mergeNext)$ **then**
    merge I_i and I_{i+1};
  end
end

Algorithm 1: Inferring DNS name ownership epochs.



(a) ENS (On-chain) **(b) Maven Central**

Figure 8: Time from DNS name registration to linkage creation for ENS On-chain (left) and Maven Central (right). The top panels show the full population; the bottom panels show only zombies. Colored curves show per-cohort trends; the dashed black curve shows the overall population. Vertical gray dotted lines mark yearly boundaries.

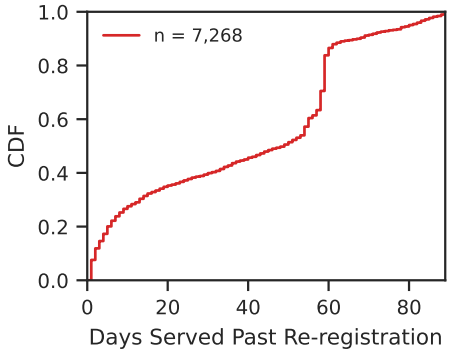


Figure 9: Duration Web PKI zombie certificates are served after their DNS names are re-registered.