# Understanding the Implementation and Security Implications of Protective DNS Services

Mingxuan Liu[*†Φ], Yiming Zhang[†Φ], Xiang Li[†], Chaoyi Lu[†], Baojun Liu[†✉], Haixin Duan[†*✉], Xiaofeng Zheng[‡†]

[*]Zhongguancun Laboratory, China [†]Tsinghua University, China [‡]QI-ANXIN Technology Research Institute, China
liumx@mail.zgclab.edu.cn, {zhangyiming, luchaoyi, lbj, duanhx}@mail.tsinghua.edu.cn,
{x-l19, zxf19}@mails.tsinghua.edu.cn

***Abstract*—Domain names are often registered and abused for harmful and illegal Internet activities. To mitigate such threats, as an emerging security service, *Protective DNS* (*PDNS*) blocks access to harmful content by proactively offering rewritten DNS responses, which resolve malicious domains to controlled hosts. While it has become an effective tool against cybercrime, given their implementation divergence, little has been done from the security community in understanding the deployment, operational status and security policies of PDNS services.**

**In this paper, we present a large-scale measurement study of the deployment and security implications of *open* PDNS services. We first perform empirical analysis over 28 popular PDNS providers and summarize major formats of DNS rewriting policies. Then, powered by the derived rules, we design a methodology that identifies intentional DNS rewriting enforced by open PDNS servers in the wild. Our findings are multi-faceted. On the plus side, the deployment of PDNS is now starting to scale: we identify 17,601 DNS servers (9.1% of all probed) offering such service. For DNS clients, switching from regular DNS to PDNS induces negligible query latency, despite additional steps (e.g., checking against threat intelligence and rewriting DNS response) being required from the server side. However, we also find flaws and vulnerabilities within PDNS implementation, including evasion of blocking policies and denial of service. Through responsible vulnerability disclosure, we have received 12 audit assessment results of high-risk vulnerabilities. Our study calls for proper guidance and best practices for secure PDNS operation.**

## I. Introduction

Human-readable domain names make it easy for users to navigate on the Internet. Unfortunately, domain names are also frequently abused for malicious activities, such as botnet command and control (C&C) [143], [11], phishing [141], spam [131], and malware distribution [9]. According to Cisco, over 91% of Internet attacks are backed by the resolution of malicious domain names [100]. More recently, ICANN established the Domain Abuse Activity Reporting (DAAR) project and publishes monthly reports on threats associated with domain names. In March 2023, DAAR reported a total of over 622k domain names that are considered malicious [85].

To contain their associated cyber threats, over the past decade, the security community has relied on *domain take-down* efforts [84], which seize malicious domain names and prevent them from resolving on the entire Internet. For example, in 2020 Microsoft seized the C&C domain for the SolarWinds attack [115]. Though effective, domain take-down should undergo cumbersome procedures (e.g., submitting complaints to domain registrars and authorities) and often require support from law enforcement (e.g., court orders) [8]. As a result, the security community seeks new methods to block resolutions of malicious domain names.

Recently, there has been a growing industry of *Protective DNS* (PDNS). Compared to complex domain take-down operations, PDNS offers a simpler alternative: when a PDNS server is queried for malicious domain names hitting its blocklist (e.g., built-in threat intelligence data), it blocks them by *rewriting* DNS responses into providing "secure" answers (e.g., resolving to reserved IP addresses). PDNS requires no changes to the DNS protocol, nor does it require collaboration from other organizations (e.g., law enforcement and domain registration providers) to block domain names, and is able to offer real-time protection. Though the concept of PDNS has not been proposed for long, it has already gained support from dozens of large DNS services, such as Cloudflare [28] and Quad9 [57]. In addition, countries including the US, Canada, and the UK are also releasing initiatives for deploying national PDNS infrastructure [128], [144], [20].

**Research questions.** For propriety reasons such as protecting domain blocklists, security policies of PDNS servers are often kept private. At the same time, users and domain holders have been complaining about PDNS mistakenly blocking benign domains, even including YouTube and Gmail [1]. A recent study also shows that the implementation of flawed "smart" DNS services may expose information about end users [68]. Despite there has been enthusiasm for PDNS from both technical and policy perspectives, little has been done to understand its actual deployment and operational status. In this paper, we fill this research gap by presenting the first measurement study and aim at answering a set of research questions, including: *How many DNS servers in the wild are offering PDNS functionalities? What are the blocking policies? Are there any security risks within the PDNS infrastructure?* We believe that answers to the questions will provide guidance to a more robust ecosystem of PDNS in the future.

**Challenges and methodology.** Identifying PDNS servers at scale is non-trivial due to their highly diversified implementa-

[Φ] Both are first authors. ✉ Corresponding author.

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.24782
www.ndss-symposium.org

tion (i.e., operators adopt blocklists and DNS rewriting policies as they see fit), and we face two major challenges. *(i)* Without prior knowledge of their blocklists, it is difficult to trigger DNS rewriting operations of PDNS services. Blindly querying the global DNS infrastructure for a vast number of malicious domain names induces ethical concerns, especially for vantage points over home devices. *(ii)* Distinguishing DNS rewriting by PDNS and other DNS manipulation schemes (e.g., off-path response injection and censorship) is also a technical challenge, as they both produce forged DNS answers.

To bootstrap our study and address the challenges, we first perform an empirical analysis of 28 popular PDNS providers and collect key observations about their blocklists and DNS rewriting policies (Section II). All providers claim explicitly about offering PDNS functionalities on their official website, and are selected from the manual review of the PDNS market and reputation. Based on the observations, we design a pipeline methodology (Section III) that actively queries *stable open DNS resolvers* (i.e., answering queries for 6 scanning experiments). Particularly, we attempt to trigger their DNS rewriting behaviors by querying each server for 10,100 domain names carefully selected from multiple open-source blocklists and popular domain lists. Finally, by distinguishing forged DNS responses from other manipulation schemes, our system identifies PDNS services.

Our methodology explicitly considers *open DNS servers* as the scope of active probing and PDNS identification. We acknowledge that by this design, PDNS servers with limited service areas (e.g., deployed by ISPs for local networks only) are overlooked, but argue that probing such servers for the purpose of this study might be ethically inappropriate. More specifically, vantage points in the local networks are required for probing closed DNS servers, and prior studies often recruit them via residential proxy networks [87], [22] and advertising platforms [36]. However, as our methodology sends queries for malicious domains in order to trigger DNS rewriting, operators of vantage points (e.g., residential devices) shall bear risks. By contrast, for open DNS servers the ethical risk should be minimal, as we exclude home devices by keeping stable servers only, and evidence shows that most of them reside in cloud platforms rather than residential networks [118].

**Major findings.** In total, we identify 17,601 servers offering PDNS functionalities (9.1% of all probed), which are distributed in 117 countries and 1,473 ASes (Section IV). From sampled Netflow data provided by an educational network, we confirm their sufficient usage: an average of 4,279 client IP addresses query the identified PDNS servers every day. Compared to regular DNS, the additional overhead in query latency of PDNS is negligible, despite additional steps (e.g., checking against threat intelligence and rewriting DNS response) required from the server side. Additionally, by inspecting forged responses returned by PDNS, we find resolving malicious domains to secure hosts affiliated with the PDNS operator or special-use addresses (e.g., 127.0.0.1) are the most popular formats of DNS response rewriting.

However, we find several security flaws in PDNS operation in the wild, which enable evasion of blocking policies or denial of response (Section V). First, after receiving queries for malicious domains, 28 PDNS servers aggressively treat the source IP addresses as threats and refuse *all* subsequent

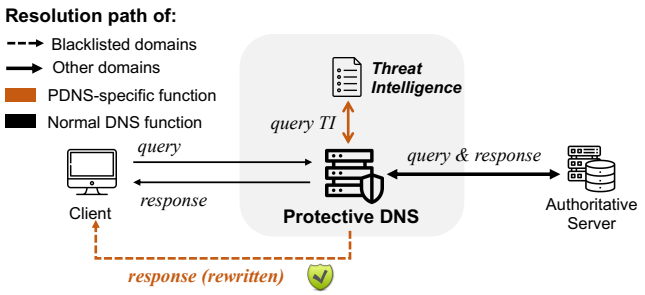


Fig. 1: The workflow of Protective DNS.

queries from them. As a result, an attacker can launch denial of response by spoofing IP addresses of legitimate users and initiating queries for malicious domains, such that the PDNS servers will reject the users. Second, when rewriting DNS responses, 26 PDNS servers return addresses pointing to dangling cloud infrastructure [16], [116], which enables domain takeover for attackers and re-engages illegal activities. Finally, when queried for harmful domains, 105 PDNS servers return *both* forged and genuine responses, and clients may still connect to malicious hosts. An additional 13 PDNS vendors only rewrite DNS messages of selected query type (e.g., `A`), which also leads to PDNS evasion if domains are resolved to malicious hosts using other resource records (e.g., `CNAME`). We have disclosed all security issues to affected operators, via their ownership reflected by ASN, PTR records and internet search results. Verisign, Neustar and ControlD DNS have responded to us and are actively engaged in discussions with us for resolution. For resolvers without vendor information, we have disclosed through national CERT agencies (the China National Vulnerability Database (CNVD) [108]). Currently, we received 12 audit verification results for *high-risk vulnerabilities* regarding Denial of Response (DoR).

**Contributions.** The contributions of this paper are as follows.

- *Understanding of PDNS ecosystem.* We present the first active measurement study on the emerging PDNS ecosystem. We design a methodology that finds 17,601 open PDNS servers, and comprehensively understand their operational status.

- *Security analysis of PDNS infrastructure.* We discover three types of security flaws within PDNS operation, which enable evasion of security protection and denial of service. We report them to affected vendors and get their positive responses, and give recommendations for PDNS implementation.

## II. Protective DNS Preliminaries

### A. Concept and Workflow

Protective DNS (PDNS) is a recursive DNS service that proactively prevents users from accessing malicious domain names at the DNS level. It achieves this through DNS response rewriting, where queries for malicious domain names are resolved to "secure" addresses (e.g., 127.0.0.1) to effectively block access. To detect malicious or suspicious domain names in DNS queries, PDNSes are often backed by (open-source or private) domain blocklists and threat intelligence [4], [18]. As

shown in the workflow of Figure 1, when PDNS receives a DNS request for a domain name, it queries threat intelligence and returns a rewritten DNS response if the queried domain should be blocked according to its policies, causing the client to visit secure servers; or queries and receives from the authoritative servers if the queried domain excludes from its blocklist and returns the normal responses.

Protective DNS is a thriving security business, and its features have been supported by numerous famous public recursive DNS services. In this study, we manually browse the official descriptions of 42 popular DNS providers (selected by their market share [12]) and find that up to 28 of them (66.67%) offer PDNS features. Overall, the 42 DNS services operate on 155 IP addresses, and 8 of them support PDNS and non-PDNS separately on different addresses. For example, Cloudflare's PDNS operates on `1.1.1.2` and `1.1.1.3`, while `1.1.1.1` provides the original DNS service.

Recently, country-level initiatives for PDNS infrastructure have emerged in the US [128], UK [20], [18], and Canada [144]. Government authorities have released official advice on PDNS deployment, including the US National Security Agency (NSA)[4], [113] and the UK National Cyber Security Centre (NCSC)[20]. This advice outlines the key features of PDNS and offers implementation guidance, e.g., the documentation for UK PDNS states that its blocklist uses various commercial and private blocklists. Besides, numerous official advice encourage the deployment of PDNS, highlighting its effectiveness in proactively mitigating cyber attacks.

### *B. Empirical Study on Popular PDNSes*

To explore the operation of Protective DNS, the two most important things to understand are which domains it will block (i.e., domain blocklists) and how to block them (i.e., DNS rewriting policies). To this end, we conducted an empirical survey of 28 PDNS providers, as summarized in Table I.[1] Below we elaborate on the details of domain blocklists and DNS rewriting policies.

**Domain blocklists.** Filtering DNS queries with domain blocklists is lightweight and flexible [136]. However, the variety and quality of protection offered by PDNS highly rely on the sources of blocklists. In our analysis of 28 renowned PDNS providers, we discovered that a single provider may utilize blocklists from multiple sources. These blocklists can be categorized into the following four groups:

• *Open-source domain blocklists.* 7 PDNS providers (25.0% of 28, e.g., AdGuard) use open-source blocklists. Blocklists in use include URLhaus [139], CyberCrime Tracker [33] and VX Vault [142]. However, reports show that open-sourced blocklists are often powered by user reports, leading to infrequent updates and abuse by malicious blocklist users [120], [93], [17], [69].

• *Private domain blocklists and threat intelligence.* Using private domain blocklists is a more popular option, adopted by 11 PDNS providers (39.29% of 28, e.g., Yandex DNS). Besides commercial threat intelligence (e.g., VirusTotal [77]), several PDNS providers also build their own domain blocklists, including OneDNS, 360 Secure DNS and 114 DNS.

[1]The survey results of non-PDNS providers is shown in Appendix A

TABLE I: Empirical study results of 28 PDNS services.

| DNS Service | CC | Type | Blocklist | Ad | Tr | Ma | Ph | Adu | DNS Rewriting |
|---|---|---|---|---|---|---|---|---|---|
| AdGuard DNS [43] | CY | Both | Open-source | ✓ | ✓ | | | ✓ | Secure IP |
| Yandex. DNS [145] | RU | Both | Open-source & Private | | | ✓ | ✓ | ✓ | Secure CNAMEs |
| Neustar UltraDNS [110] | US | Both | Unknown | | | ✓ | ✓ | ✓ | Secure IP |
| Cloudflare DNS [28] | US | Both | Unknown | | | ✓ | | ✓ | Special-use IP |
| CIRA DNS [129] | CA | Both | Open-source & Private | | | ✓ | ✓ | ✓ | Secure IP |
| OneDNS [114] | CN | Both | Private | ✓ | | ✓ | | ✓ | Secure IP |
| Quad9 DNS [57] | CH | Both | Open-source & Private | | | ✓ | | | Response Code |
| CleanBrowsing DNS [26] | US | PDNS-only | Open-source &Private | | | ✓ | ✓ | ✓ | Secure IP |
| Comodo DNS [30] | US | PDNS-only | Open-source & Private | ✓ | | ✓ | ✓ | | Secure IP |
| Open DNS [24] | AT | PDNS-only | Unknown | | | ✓ | | ✓ | Secure IP & No Data |
| SkyDNS [132] | RU | PDNS-only | Private | | | ✓ | | | Specil-use IP |
| Comss DNS [31] | RU | PDNS-only | Open-source & Private | ✓ | ✓ | ✓ | ✓ | | Special-use IP & No Data |
| SafeDNS [125] | EU | PDNS-only | Unknown | | | ✓ | | ✓ | Secure CNAMEs |
| DNS for Family [70] | DE | PDNS-only | Unknown | | | | | ✓ | Secure IP |
| CZ.NIC DNS [34] | CZ | PDNS-only | Unknown | | | | ✓ | | Secure IP |
| Ali DNS [7] | CN | PDNS-only | Private | | | ✓ | ✓ | | Secure IP |
| 360 Secure DNS [3]* | CN | PDNS-only | Private | | | ✓ | ✓ | | Secure IP |
| 114 DNS [42]* | CN | PDNS-only | Unknown | ✓ | ✓ | ✓ | ✓ | ✓ | Secure IP |
| DNSPod DNS+ [138] | CN | PDNS-only | Unknown | | | ✓ | | | No Data |
| ControID DNS [32] | CA | Both | Unknown | ✓ | | ✓ | ✓ | ✓ | Special-use IP |
| Norton DNS [54] | US | PDNS-only | Private | | | ✓ | ✓ | ✓ | Response Code |
| Privacy-First DNS [56] | US | PDNS-only | Unknown | ✓ | ✓ | ✓ | ✓ | | No Data |
| Strongarm DNS[61] | US | PDNS-only | Unknown | | | ✓ | | | Secure CNAMEs |
| SafeSurfer[59] | US | PDNS-only | Unknown | ✓ | ✓ | ✓ | ✓ | ✓ | Secure CNAMEs |
| DNS Forge[72] | US | PDNS-only | Unknown | ✓ | | ✓ | | | Secure IP |
| Aha DNS[44] | US | PDNS-only | Unknown | | | ✓ | | | No Data |
| Alternate DNS [45] | US | PDNS-only | Unknown | ✓ | | | | | Secure IP |
| SafeServe [58] | US | PDNS-only | Unknown | | | ✓ | | | No Data |

[1] Type: Both (offers PDNS and non-PDNS on different IP addresses) / PDNS-only.
[2] Domain categories of blocking: Ad (advertisements) / Tr (tracker) / Ma (malware), Ph (phishing) / Adu (adult contents)
* Supports user complaints and corrections.

• *Unknown sources of blocklists.* 16 PDNS providers (57.14% of 28, e.g., UltraDNS) refrain from disclosing any information about their blocklist source.

• *User complaints and corrections.* The use of private and unknown sources for domain blocklists creates uncertainty for domain holders regarding the inclusion and reasons for blocking their domains. Unfortunately, in cases of false positives where domains are mistakenly blocked, only two PDNS providers (360 Secure DNS and 114 DNS) explicitly provide channels for users to report and address such issues.

The variety of blocklist sources indicates that PDNS providers are focused on blocking different types of domains. Specifically, out of the providers analyzed, 24 block malware, 14 block phishing websites, 14 offer child mode protection to block adult or gambling content, and 2 (e.g., UltraDNS) allow users to customize their own filtering policies.

**DNS rewriting policies.** The DNS response rewriting policies of the 28 PDNS providers remain undisclosed. To uncover these policies, we constructed a small set of domain names using open-source threat intelligence. Following the methodology described in Section III-B, we randomly select 100 domain names for each of 5 categories, including Malware, Botnet, Phishing, Adult, Spam and Tracker. We then query all PDNS providers for the domain names and inspect the DNS responses for rewriting behaviors, and we find 5 possible policies.

• *Special-use IP addresses.* Malicious domain names are resolved to reserved IP addresses (e.g., `0.0.0.0`) or loopback addresses (e.g., `127.0.0.1`). Adopted by 4 PDNS providers (e.g., SkyDNS).

• *Secure IP addresses.* Malicious domain names are resolved to a limited number of secure IP addresses. Note that these secure IP addresses are not necessarily affiliated with PDNS providers (e.g., third-party sinkhole servers of security companies or cloud servers). Adopted by 14 PDNS providers (e.g., UltraDNS).

• *Secure CNAMEs.* Malicious domain names are pointed to a sinkhole domain name (e.g., `safe1.yandex.ru` and `block.safesurfer.io`) using a CNAME record. Adopted by 4 PDNS providers (e.g., Yandex DNS and Safe Surfure DNS).

• *Response code.* The DNS response code (RCODE) is changed to some error codes, adopted by 2 PDNS providers. Specifically, Norton DNS changes the DNS RCODE to `REFUSED`, refusing to resolve malicious domain names, and Quad9 DNS utilizes `NXDOMAIN` to block malicious domains.

• *No data.* The DNS response excludes any answer. Adopted by 6 PDNS providers (e.g., OpenDNS and Comss DNS).

From our results, a single PDNS provider may employ multiple rewriting policies. For example, OpenDNS may either resolve malicious domain names to secure IP addresses or provide "No data" for them. To validate the implementation of these blocking strategies by PDNS providers, we discussed with two collaborating PDNS providers, 114 DNS and 360 DNS. They confirmed the effectiveness of our blocklist and verified that the rewriting policies identified through our measurements align with their own implementations.

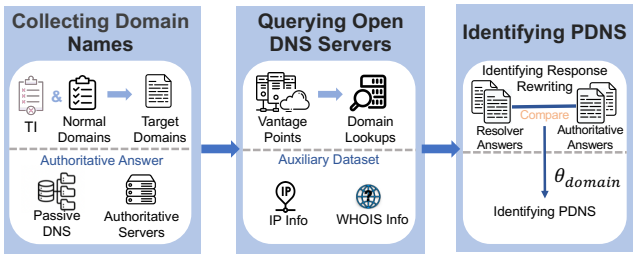


Fig. 2: PDNS identification system overview

Our empirical study reveals significant variations in PDNS policies. In fact, considering the diversified implementation and opaque filtering policies, this is an expected result as no consensus or technical standards around PDNS have ever been made. With the prominent roles the services could play in combatting malicious activities, limited efforts have been made to understand PDNS operation and security at scale, due to the challenges in identifying PDNS services. Meanwhile, some concerned voices about PDNS are emerging in the DNS community, e.g., privacy concerns [123]. One group of users has complained that PDNS providers may mistakenly block benign domains, even well-known ones, and the blocking function is not stable [1]. To address this research gap, our study aims to propose a methodology for identifying PDNS in the wild, as well as performing large-scale analysis to reveal their operational status and security implications.

## III. Methodology

Our goal is to develop an efficient system to identify PDNS operating in the wild. However, this task is non-trivial due to the opaque nature of PDNS security policies. To overcome this challenge, we base our methodology on key observations of popular PDNS services (mentioned in Section II). In this section, we elaborate on the methodological details. To aid the community's understanding of PDNS, we have made the identification scripts and selected results publicly available [2] (relevant ethical considerations are described in Section VI-A).

### A. Overview

From our survey on popular PDNSes, we find that all of them block malicious domain names through DNS response rewriting. As a result, the heuristics for identifying PDNS becomes straightforward: *(i)* collect a set of malicious domain names with their authoritative DNS answers, *(ii)* query open DNS servers for malicious domain names, *(iii)* compare DNS responses with authoritative answers, and *(iv)* find response rewriting and identify PDNS.

**Technical challenges.** The PDNS identification method faces three technical challenges. First, in step *(i)*, we need malicious domains likely to be blocked by PDNS servers, without prior knowledge of their domain blocklists. The blocked domain names in PDNSes, as shown in Table I, exhibit significant variation, and the blocklists are often too extensive to enumerate. As a solution, we compile a list of *"generally-malicious"* domains that are expected to be included in the blocklists of

[2]https://github.com/MingxuanLiu/ProtectiveDNS

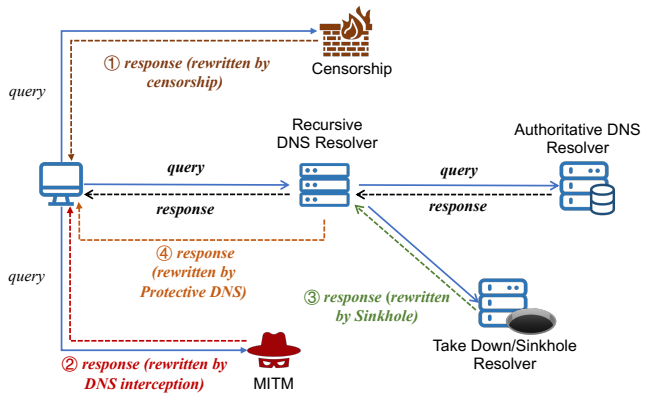


Fig. 3: Possible DNS Manipulation.

multiple PDNSes. Second, to enable response comparison in step *(iii)*, we need authoritative DNS answers of malicious domain names. Previous studies [118], [111], [87] collect authoritative answers by repeatedly querying the authoritative servers of a given domain. However, domains of our interest are barely benign and frequently change their authoritative answers to evade detection (e.g., malware domains switching IP addresses within a pool [127]), making it challenging to enumerate them by querying authoritative servers. Relying solely on data from authoritative servers may lead to errors during response comparison because the answers are not comprehensive. To address this challenge, we also extract DNS answers from an extensive passive DNS dataset[3], which includes aggregated DNS logs from distributed servers across networks and regions. Third, in step *(iv)*, we need to distinguish the modified responses from PDNS and from other DNS manipulations, specifically DNS hijacking and censorship [118], [111], [87]. To solve this challenge, we devise a series of identification methods.

**Identifying DNS response rewriting.** From our observations on popular PDNSes (Section II-B), we focus on the rewriting of the *address resource record set* (RRSet), i.e., type `A`. Following studies on detecting DNS response manipulation [118], [111], [87], we consider a DNS response is rewritten, if the enclosed IP addresses do not share any Autonomous System Numbers (ASN) with authoritative answers.

However, the above step only finds rewritten DNS responses, but does not tell whether the response is rewritten by PDNS or other DNS modification schemes. Figure 3 shows the 4 most common ways of modifying DNS resolution results. The core idea we use to distinguish between PDNS and other modifications is:

• Censorship. To minimize censorship impact, we initially selected vantage points (VPS devices for DNS queries) in the US, Japan, and the UK, with high network freedom rankings [135]. We further filtered out potentially censored domains, e.g., political websites, using the Citizenlab dataset [92]. Despite the vantage points being in high-freedom locations, censorship in the countries of the resolvers could still influence results. Therefore, we employed established DNS censorship detection methods [78], [112], sending test domains to a random IP in each resolver's AS not offering DNS resolution on port 53. Any returned result indicates censorship interference, leading to the exclusion of the corresponding censored domain from that PDNS resolver results.

• DNS hijacking. DNS hijacking is another potential source of modification in the resolution path. Existing research [40], [38] agrees that distributed domain queries can effectively mitigate the impact of hijacking. It is difficult for attackers to hijack requests originating from diverse geographical locations following different resolution paths simultaneously. In our study, we exclude inconsistent responses from 3 vantage points with PDNS modifications count. This may understate PDNS numbers but reduces false alarms effectively.

• Domain takedown (sinkhole). When domains are taken down, they often switch their NS to a secure or controlled NS that returns modified results. These modifications occur at the "authoritative level", not the "recursive level". Since our method identifies modifications by comparing them with authoritative resolution results, such changes on the authoritative side will not be mis-detected as PDNS modifications.

To mitigate other potential influences, we established a threshold for determining PDNS. Namely, a DNS resolver is deemed to deploy a PDNS strategy only if the number of rewritten domain resolution results surpasses a specific threshold. This involved evaluating 42 popular DNS providers (28 PDNS and 14 non-PDNS) to identify a clear threshold for blocked domains. Further details are provided in Section III-B.

### B. System Design

Following our heuristics for identifying PDNS, Figure 2 illustrates the workflow of our system.

**Step I: Collecting domain names.** To bootstrap our scan for PDNS, we compile a list of malicious domain names that are likely to be blocked by multiple PDNSes (i.e., "generally-malicious"). Since most PDNS blocklists are not publicly available, we gather such domain names from open-source blocklists. Through an extensive survey of previous studies utilizing domain blocklists [69], [136], [93], [17], we select seven sources (listed in Appendix B). The blocklists consist of 5,226,699 domain names, covering 6 malicious categories: malware, phishing, tracker, botnet, adult and spam.

To identify potentially malicious domain names, we initially select 40,753 domains flagged as malicious by at least two sources. We then cross-check these domains with VirusTotal [77] and retain 36,533 domains that are classified as highly malicious by affiliated security vendors. To reduce DNS query volume, we randomly sample and maintain the original ratios of each category. The final blocklist consists of 10,000 domain names, as shown in Table II. We also analyze their WHOIS status and find that approximately 22.52% are unresolvable due to being on `Hold` by domain registries and registrars or lacking delegation information (i.e., `inactive`) [79]. To test the DNS rewriting and blocklist updating policies of PDNS, we keep non-resolvable domains in our list. As the control group to test the DNS connectivity, i.e. whether DNS resolution is still available, we also collect a random sample of 100 popular domain names from Tranco's top list [119] and exclude

[3]This passive DNS dataset was obtained from our collaborating reputable security company.

TABLE II: Categories and WHOIS status of selected malicious domain names.

| Category | # Domains | WHOIS status | # Domains |
|---|---|---|---|
| Malware | 4,231 | Not resolvable | 2,252 |
| Botnet | 3,962 | `serverHold/clientHold` | 128 |
| Phishing | 867 | `inactive` | 2,124 |
| Adult | 667 | Resolvable | 7,748 |
| Spam | 259 | | |
| Tracker | 14 | | |
| **10,000 Malicious Domain Names** | | | |

possible censored domains using the Citizenlab dataset [92]. Our final domain list comprises **10,100** domains, with 10,000 malicious domains of 6 categories and 100 popular domains.

To compare resolution results, we gather authoritative answers for all domains from their authoritative servers and passive DNS. Before each scanning experiment, we query the authoritative servers of each domain from 3 vantage points (see Step II) and log all IP addresses in DNS responses as their authoritative answers. In passive DNS datasets, each record is a tuple, like $< time\_first, time\_last, count, rrname, rrtype, rdata >$, indicating that during the period from $time_first$ to $time_last$, the domain name $rrname$ was resolved to $rdata$ by associated DNS servers $count$ times. For each domain name $d$ in our domain list, we select records that satisfy: *(i)* $rrname = d$, *(ii)* $rtype$ = `A` *or* $rtype$ = `CNAME`, *(iii)* $count > 5$, *(iv)* $time\_last$ is later than 2022, and take all $rdata$ as its authoritative answers.

**Step II: Querying open DNS servers.** Studies have reported that open DNS servers perceive significant IP churn, because most are operating on network devices (e.g., home routers) [126]. Both for their unlikely role as PDNS and ethical concerns when querying them for malicious domain names, we focus on *"stable"* recursive DNS resolvers. To identify *"stable"* resolvers, we conducted an Internet-wide scan of port 53 from Feb to Mar 2022. We removed servers discovered in the initial scan that stopped responding during the two-month period. Additionally, we applied criteria from previous studies [118], [94] to select DNS servers: *(i)* those using the same IP address for receiving DNS queries from clients and sending queries to authoritative servers (indicating recursive mode), *(ii)* servers located in ASes associated with popular DNS services (e.g., AS13335 Cloudflare), or *(iii)* servers with DNS keywords in the `PTR` record of their IP address. Finally, we identified **193,888** "stable" recursive DNS resolvers.

We then query each selected DNS server for all 10,100 domain names in our list. We use XMap [95], a fast network scanner modified from ZMap, to issue type-`A` DNS queries and log responses. To mitigate network jitter, each DNS query is repeated 3 times. To avoid censorship, we utilize 3 cloud servers (country codes: US, JP, UK) on Alibaba Cloud [76] as vantage points, where we then initiate all DNS queries. Prior to the experiment, we obtained permission from Alibaba Cloud by reporting our scanning purposes and methods. For all cloud server addresses, we configure their `PTR` records as "research scanner" with a mail address. We also set the query rate at 2 queries per second per target DNS server to eliminate impact on their operation.

**Algorithm 1:** IDENTIFYING DNS REWRITING

**Input:** Answers from DNS resolver $R_{re}$; Answers from authoritative name server $R_{auth}$; Answers from passive DNS $R_{passive}$; Special-use and known secure IPs $K$; Censorship Groundtruth dataset $censor_{ground}$; Vantage point at $v_i$

**Output:** DNS response is rewritten (Boolean)

```
if R_re in censor_ground then
    return False                          ▷ Rewritten by censorship
end
if diff_with_vantages(v_i, v_others)==True then
    return False                          ▷ Rewritten by Hijacking
end
if R_re^RCODE ∩ R_auth^RCODE == ∅ then
    return True
end
if R_re^AS ∩ R_auth^AS ≠ ∅ or (R_re == ∅ and R_auth == ∅)
  then
    return False
end
else
    if R_re^AS ∩ R_passive^AS == ∅ then
        return True
    end
    else if (R_re ∩ K) ≠ ∅ then
        return True
    end
    return False
end
```

**Step III: Identifying PDNS.** From the scanning results of Step II, we interpret whether each DNS result has been rewritten, and subsequently identify PDNS based on a defined threshold for the number of rewrites for each DNS resolver.

First, we utilize Algorithm 1 to identify whether a DNS result is rewritten by "protective" mechanism of the DNS resolver. After querying a DNS server for a domain name, the algorithm takes the following as input:

- $R_{re}$: RRSets returned by the DNS server.
- $R_{auth}$: RRSets collected from authoritative servers (see Step I in Section III-B).
- $R_{passive}$: RRSets extracted from passive DNS datasets (see Step I in Section III-B).
- $K$: Special-use IP addresses, including private network addresses (e.g., `192.168.0.0/16`), loopback addresses (e.g., `127.0.0.1`) and reserved IP addresses (e.g., `240.0.0.0/4`). We also include several IPs that popular PDNSes use to block malicious domains (obtained via our survey in Section II). Because these addresses prevent domains from public access, attackers are unlikely to resolve malicious domain names to them. As a result, malicious domain names being resolved to special-use addresses becomes a compelling indicator of DNS response rewriting.
- $censor_{ground}$: Ground-truth of censorship interference, which is a list of the results that are returned as censorship

results for a domain by the AS in which a certain resolver is located (see "Censorship" in Section III-A).

• $v_i$: Vantage point from which $R_{re}$ is obtained, enabling the filtering of DNS hijacking by comparing it with results from other vantage points ($v_j (j \neq i)$) (see "DNS Hijacking" in Section III-A)).

Next, we query the GeoIP2 database [102] to obtain the ASNs for the IP addresses in $R_{re}$, $R_{auth}$, and $R_{passive}$, these ASNs are denoted as $R_{re}^{AS}$, $R_{auth}^{AS}$ and $R_{passive}^{AS}$. Using these data, we perform the following comparisons:

• *Filter out rewrite by censorship (Line 1~3).* If the ASN of a resolver and the result returned to a domain are in the censorship ground-truth list ($censor_{ground}$), then the rewrite is possibly caused by censorship.

• *Filter out rewrite by hijacking (Line 4~6).* If the DNS result obtained by vantage point $v_i$ is different compared to results from other vantage points, then the rewrite is possibly caused by DNS hijacking.

• *Comparison with Rcode from authoritative servers (Lines 7~9).* If the Rcode in $R_{re}$ matches the Rcode in $R_{auth}$, the response is not rewritten.

• *Comparison with answers from authoritative servers (Lines 10~12).* If there is any shared ASN between the addresses in $R_{re}$ and the addresses in $R_{auth}$, or if both $R_{re}$ and $R_{auth}$ are empty, it signifies that the response has not been rewritten.

• *Comparison with answers from Passive DNS (Lines 14~16).* If there are no shared ASNs between the addresses in $R_{re}$ and the addresses in $R_{passive}$, the response is rewritten.

• *Comparison with special-use and blocked IPs (Lines 17~19).* If any IPs in $R_{re}$ fall in $K$, the response is rewritten.

In addition to individual result comparisons, we determine a threshold $\theta_{domain}$ to identify PDNS server if it blocks more than $\theta_{domain}$ malicious domain names. To evaluate this, we assess the same set of 42 popular DNS vendors as studied in Section II. These DNS vendors are associated with 155 IP addresses, each classified as either PDNS or non-PDNS during our empirical study, which serves as our ground truth.

We conducted 30 rounds of scanning experiments from the same 3 vantage points (US, UK, and JP) as in Step II, with one round per day in May 2022. Each round involved querying 10,100 domain names towards each DNS server and logging the count of malicious domains with rewritten responses. We observed a significant difference in the distribution of blocked domain names: PDNSes blocked an average of 302 domain names, while non-PDNSes rewrote 33 domain names. Upon manual inspection, we speculate that the identified rewrites in non-PDNSes may be false alarms caused by network failures. When the network condition is poor, resolvers may directly return "no data" to clients. However, the proportion of false alarms can be filtered using a threshold. We compare the recognition results with the corresponding ground-truth data at different thresholds, and evaluate each threshold with its precision, recall, and f1-score (calculation details are listed in Appendix C). Table III presents the average values from 30-round measurement results at each vantage point. It can be seen that, $\theta_{domain}$ = 50 provides the highest performance, thus selected as the threshold.

TABLE III: Evaluation of threshold $\theta_{domain}$ selection

| Threshold | Precision | Recall | F1-score |
|---|---|---|---|
| 30 | 69.97% | 75.63% | 72.66% |
| 40 | 81.65% | 86.23% | 88.14% |
| **50** | 93.04% | 98.35% | 94.06% |
| 60 | 93.12% | 89.97% | 90.14% |
| 80 | 95.37% | 83.24% | 86.33% |
| 100 | 96.27% | 53.79% | 56.38% |
| 150 | 98.27% | 23.47% | 39.22% |

### C. Limitations

The opaque nature of the PDNS ecosystem imposes several limitations on our system. First, due to the invisibility of PDNS domain blocklists, we rely on selecting malicious domain names from multiple open-source blocklists. However, these open-source blocklists are often incomplete and infrequently updated. Consequently, we may miss PDNSes in the wild that employ different blocklists. Second, to collect authoritative answers for domain names, we merge answers from authoritative servers with data extracted from passive DNS. While answers from authoritative servers may not be comprehensive, the passive DNS data (spanning several years) can become stale. Although we remove data prior to 2022, there is still a possibility of including stale data in our authoritative answers, leading to errors in response rewriting identification. Third, we strived to eliminate other possible DNS manipulations including censorship, domain takedown and DNS hijacking. However, completely excluding other manipulations presents significant challenges. As for DNS hijacking, since it generally redirects to illegal or monetization-related content, we crawled and examined the webpages linked to the rewritten IP addresses to identify potential false alarms. To filter out suspicious pages, we employed a list from previous research [96] that contains keywords like "money/monetization" and "profit". Results indicate that among 155 Resolvers from 42 vendors mentioned in Section II, 96.13% of PDNS did not find false alarms of DNS hijacking in 30 scans. Only 6 resolvers (from 4 vendors) experienced DNS hijacking, averaging fewer than 7 instances. The number of false alarms is well below our threshold for identifying PDNS, ensuring minimal impact on the results. Despite our proactive efforts to identify and exclude censorship from the measurement results, accurately distinguishing them remains challenging due to their complex nature (a case study is given in Section IV-E). While fully understanding the precise impact of censorship remains a challenge, our study still provides valuable insights into the current state of PDNS. As a result, in later sections, we only pose our results as the lower bound of actual PDNS deployment.

## IV. PDNS Characteristics

In this section, we perform extensive studies at the macro-level (PDNS implementation) and micro-level analysis (including querying performance, domain blocklists, and rewriting policies) to understand the operational status of PDNS.

TABLE IV: Top 10 countries and ASNs with the most Protective DNS resolvers.

| CC | # IP | ASN | # IP |
|---|---|---|---|
| US | 6,296 (35.8%) | 20115 (CHARTER-20115) | 1,074 (6.1%) |
| IRN | 1,225 (7.0%) | 3303 (SWISSCOM) | 777 (4.4%) |
| CN | 1,205 (6.8%) | 209 (CenturyLink Communications) | 705 (4.0%) |
| JP | 1,056 (6.0%) | 5617 (TPNET) | 613 (3.5%) |
| CH | 804 (4.6%) | 17506 (UCOM) | 576 (3.3%) |
| PL | 745 (4.2%) | 10796 (TWC-10796-MIDWEST) | 570 (3.2%) |
| MD | 635 (3.6%) | 21342 (AKAMAI-ASN2) | 523 (3.0%) |
| ID | 540 (3.1%) | 8926 (MOLDTELECOM-AS) | 480 (2.7%) |
| OM | 380 (2.2%) | 2519 (VECTANT) | 420 (2.4%) |
| RO | 367 (2.1%) | 50010 (Nawras-AS) | 379 (2.2%) |
| 117 Countries | | 1,473 ASNs | |

### A. Large-scale Identification of PDNS

First, we identified 193,888 "stable" recursive DNS resolvers during a two-month scanning, covering 192 countries and 8,112 ASNs, with detailed information shown in Appendix D (Figure 9 and Table XII). Then we initiated the scanning experiment on 10,100 domains from 3 vantage points (US, UK and JP) towards these resolvers, identified the blocked domains, and utilized the threshold of 50 blocked domains to identify PDNS. To mitigate network jitter, we queried each domain name 3 times, and conducted a total of *6 separate scanning experiments* on these resolvers to discover PDNS resolvers. By intersecting the results from the 6 results, we identify **17,601** (9.08%) PDNS resolvers in the wild.

### B. Implementation of PDNS in the Wild

The identified 17k PDNSes from 193k open DNS resolvers, are widely dispersed throughout 117 countries and 1,473 ASNs. According to their geographic distribution shown in Appendix E, the deployment of PDNS differs significantly among different countries. From the top 10 countries shown in Table IV, it can be seen that the US dominates the largest number of PDNSes with 6,296 resolver IP addresses. As mentioned in Section II, official deployment policies for PDNS [23] have been published in the US, which has clearly corroborated the rollout process. Specifically, PDNSes in the US encompass 208 ASNs, demonstrating their wide coverage. Even university DNS resolvers have implemented protective schemes, with 13 PDNS identified across 8 universities, including Columbia University and the University of California, Los Angeles.

In addition to the absolute volume of PDNS resolvers, the deployment rate also varies significantly among countries. Our measurements show that the global average deployment rate of PDNS is currently 9.08%. For the US, which holds the largest number of PDNSes, the deployment rate is 21.60%. For China, despite ranking third in PDNS numbers, the deployment rate is only 4.53%, resulting in limited usage and traffic for protective services. To further analyze implementation rates in countries with mature and reliable DNS service ecosystems, we examined 33 countries with over 1,000 open resolvers. Among them, Russia has a low PDNS deployment rate of 1.05% (183 PDNSes), although it holds up to 17,431 open resolvers (3rd globally).

To assess the real-world adoption of PDNS, we leveraged a 1:1000 sampled 1-year Netflow data [109] provided by an education network. We focused on flow data associated with the 17k identified PDNSes with port 53, the official DNS service port. On average, we observed 33,467 unique (IP, port) tuples per day. In this dataset, we identified 23,847 clients utilizing PDNS resolvers, resulting in 9,470,810 DNS resolutions within the covered ISP. On average, there were 4,279 clients per day. Considering the sampling impact, the actual number of clients using PDNS is likely higher. Furthermore, we discovered certain network segments with a high volume of PDNS queries, concentrated on a few PDNS resolvers. For instance, the 222.192.186.0/24 segment, serving as an Internet exchange point (IXP) for a Chinese campus network, generated an average of 792 domain name query requests per day, all directed to two PDNS resolvers from DNSPai [29]. We speculate that this may be the result of a centralized configuration by the network administrator.

### C. Querying Performance of PDNS

To enhance resolution security with domain blocking, PDNS requires extra steps like blocklist-based domain matching beyond regular domain resolutions. Ideally, such modification should not compromise and even facilitate the DNS query processing performance. However, the actual impact remains unknown due to their non-transparent deployment. In this work, we use a representative metric, the round-trip-time (RTT), for querying performance evaluation. RTT records the time spent between sending DNS queries and receiving responses and is one popular benchmark for DNS performance assessment [96]. Given a large number of open resolvers (193k), we focus RTT evaluation on the 155 well-known DNS resolvers mentioned in Section II.

One important factor affecting DNS querying performance is cache utilization, i.e., does the resolver need to send recursive queries to authoritative resolvers (without cache) or return the cached responses directly. We conducted tests with 4 sets of parameters separately, measuring the RTT of PDNS with and without cache, for its blocked domains and other domains. Specifically, in the without-cache experiment, we only perform a single query for each domain in our domain dataset towards the target DNS resolvers. Conversely, in the with-cache experiment, we query each domain twice within a 5-second interval, confirming cache hits through the maximum TTL value [121] and bypassing potential load-balancing policies [94].

As shown in Figure 4, PDNS responds quicker to blocked domains (the blue solid line) than other domains (the orange solid line) in the absence of caching. Specifically, 70.47% of blocked domains would be replied to in less than 0.2 seconds, while that proportion for non-blocked domains is only 44.44%. However, the difference becomes less pronounced when caching is enabled (dashed lines). We speculate that the reason is the "deployment location of the protective mechanism", i.e., PDNS prefers to block domains before recursive resolution. Once the query hits the blocklist, it will return the modified (secure) response directly. We consulted experts from

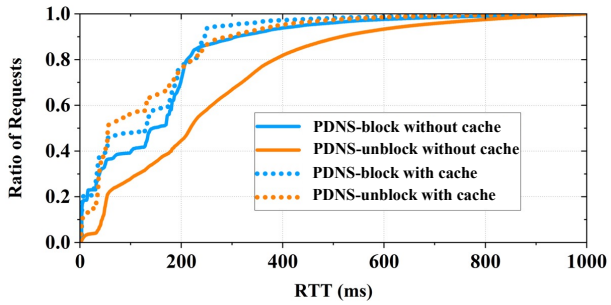


Fig. 4: ECDF of PDNS querying RTT.

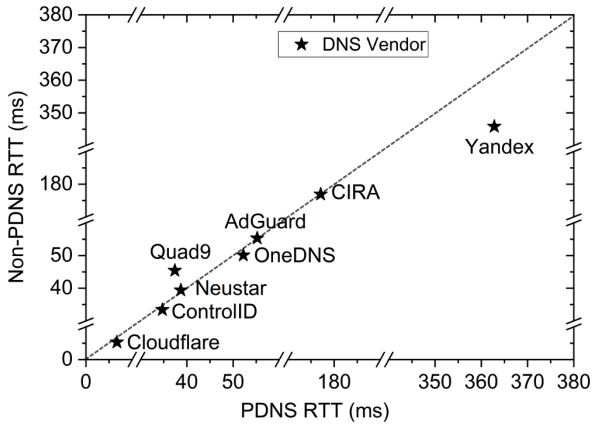


Fig. 5: Median of RTT results (ms) of PDNS and Non-PDNS within the same DNS vendor with caching.

well-known DNS vendors, who confirmed that their blocking mechanism aligns with our observations.

We conducted a performance comparison between PDNS and non-PDNS resolvers in terms of resolving RTT for non-blocked domains. Considering variations among DNS vendors, we selected 8 vendors that offer both protective and completely unprotected resolvers (see Section II), and made a comparison within the same vendor. Figure 5 shows that the difference in median RTT results (with cache) is minor for most (6) DNS vendors (within 2ms). However, Yandex's PDNS performs remarkably worse than non-PDNS resolvers, and its overall performance is also worse than other vendors (consistent with DNSperf [66]). We speculate the discrimination may be related to their specific protective policy implementation.

### D. Domain Blocklist of PDNS

The blocklist, pivotal to PDNS quality, often remains undisclosed as detailed in Section II-B. Consequently, we execute an initial evaluation of PDNS blocklists, informed by their actions on our collected 10k malicious domains.

Our findings reveal that 57% of PDNSes block over 500 malicious domains, as shown in Table V. Notably, prominent DNS vendors tend to limit their PDNS blocklists, with 43% blocking fewer than 100 domains. Analysis of domain categories indicates these vendors adopt a conservative blocklist approach, focusing on a narrow set of "high-risk" domains. Specifically, 74.84% of blocked domains are associated with

TABLE V: Blocked domain numbers.

| Category | 50~100 | 100~500 | 500~1000 | >1000 |
|---|---|---|---|---|
| Well-known | 36 (43%) | 19 (23%) | 9 (11%) | 20 (23%) |
| All identified | 2,733 (16%) | 4,813 (27%) | 5,373 (30%) | 4,682 (27%) |

TABLE VI: Category of domains blocked by PDNSes.

| Category | # Test domains | # Avg. blocked domains | PDNS Coverage |
|---|---|---|---|
| Malware | 4,231 | 961.9 | 17,596 (99.97%) |
| Botnet | 3,962 | 472.0 | 17,529 (99.59%) |
| Phishing | 867 | 160.9 | 17,213 (97.80%) |
| Adult | 667 | 119.8 | 12,680 (72.04%) |
| Spam | 259 | 96.6 | 16,628 (94.47%) |
| Tracker | 14 | 0.5 | 3,779 (21.47%) |

Malware and Botnet, while Adult (9.49%) and Spam (3.83%) domains are seldom blocked. We posit that well-known PDNSes employ "conservative" blocklists to avoid potential negative impact from aggressive blocking on their extensive, complex user bases, likely due to usability considerations.

We assess the blocking propensity of PDNSes by domain categories, as shown in Table VI, noting the most frequent pattern includes Malware, Botnet, Phishing, and Spam (yellow). Malware is the most blocked category with 99.97% of PDNSes blocking an average of 962 domains. Despite including only 259 spam domains, 94.47% of PDNSes blocked them.

While PDNS blocklist sources remain undisclosed, we approximate their similarities using the Jaccard Index [137] on their blocking domain sets. Figure 6 shows the similarity results for 28 well-known PDNS vendors and several exhibit significant correlations. The most analogous are SkyDNS and SafeDNS (similarity of 0.99), seemingly parallel services for Russia and other nations [2]. Quad9, a public threat intelligence provider [57], appears to be a blocklist hub for similar vendors like Ali DNS, DNSPod, and 114 DNS, with similarities exceeding 0.80. Alternate DNS, averaging a mere 0.21% blocklist similarity with others, presumably employs a distinct list or targets ad-related domains exclusively.

### E. DNS Rewriting Policies of PDNS

**Category of policies.** Table VII associates PDNSes with their rewriting policies divided into five categories identified in our empirical study (Section II-B). The most prevalent approach, adopted by 56.45% of PDNSes, is to respond with secure IP addresses, yielding 577 secure IPs. Of those, 28.0% (162 IPs) return a block notification we page when accessed via HTTP(S), e.g., block/interdict/intercept/obstruct and complaint/appeal/grievance. A subset (14 IPs) also provides avenues for user complaints. Notably, a PDNS of Interkam [86] itemizes all blocked domains on their secure IP address[4], derived from a public blocklist[5] of 26,882 domains. Our blocklists and theirs share 94 domains, affirming the representativeness of our selected blocklist.

[4]https://spyblock.interkam.pl/
[5]https://malwaredomains.com

TABLE VII: Rewriting policies and the average number of blocked domains for each category.

| # Rewriting Policy | # PDNS | # Policy | # Blocked Domains | # Malware | # Botnet | # Phishing | # Adult | # Spam | # Tracker |
|---|---|---|---|---|---|---|---|---|---|
| **Secure IP** | 9,935 (56.45%) | 577 | 483 | 332 | 58 | 45 | 27 | 20 | 1 |
| **Special-use IP** | 7,209 (40.96%) | 351 | 424 | 371 | 12 | 12 | 8 | 20 | 1 |
| **No Data** | 822 (4.67%) | - | 222 | 142 | 44 | 16 | 9 | 11 | 0 |
| **Secure CNAME** | 449 (2.55%) | 70 | 544 | 375 | 58 | 46 | 24 | 40 | 1 |
| **Error Response Code** | 408 (2.32%) | 3 | 362 | 267 | 28 | 33 | 13 | 20 | 1 |

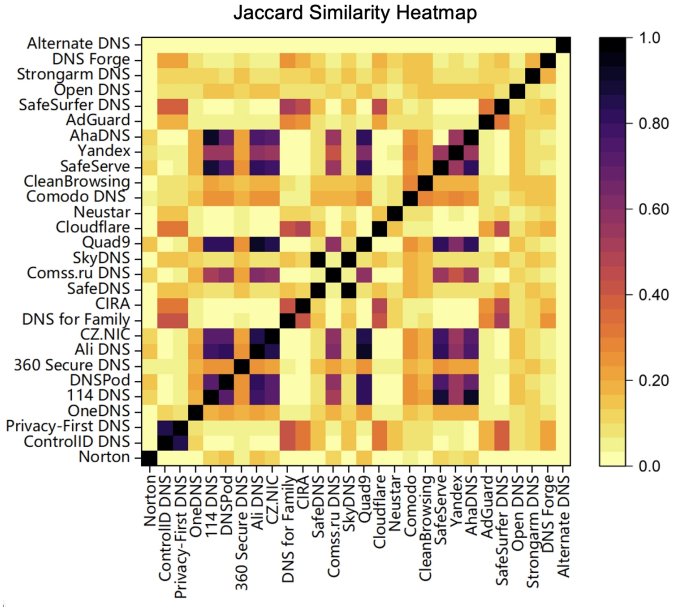


Fig. 6: Blocklist similarities between PDNS services.

TABLE VIII: Top 5 groups of PDNSes.

| Group | # PDNS | Country | AS |
|---|---|---|---|
| **Group 1** | 379 (2.2%) | Oman | 50010 (Omani Qatari Tele. Company SAOC) |
| **Group 2** | 378 (2.1%) | United States | 7029 (Windstream Communications LLC) |
| **Group 3** | 143 (0.8%) | United States | 4181 (TDS TELECOM) |
| **Group 4** | 119 (0.7%) | United States | 7018 (AT&T Services, Inc.) |
| **Group 5** | 63 (0.4%) | Romania | 9050 (ORANGE ROMANIA COMMUNICATION S.A) |

Special-use IP addresses [81] are prevalent in PDNS, comprising 40.9%. PDNSes utilize 3 types of special IPs, i.e., private-use (`0.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`), shared (`10.64.0.0/10`), and loopback addresses(`127.0.0.0/8`). Intriguingly, 35 PDNSes return category-specific IPs. For instance, PDNSes from Hosting24 [80] return `127.42.0.15` for Malware and `127.42.0.155` for Phishing domains.

Despite only 2.55% of PDNSes using secure CNAME, they block most domains on average, encompassing 545. Aside from employing their sinkhole CNAME domains like Yandex DNS, PDNSes also use third-party sinkhole blocking services. For instance, 68 PDNSes utilize DNSFilter's sinkhole-blocking service [65], such as `malware.demo.spsredir.dnsfilters.com`.

Within 17k PDNSes, 3 error response codes, namely NXDomain, ServFail, and Refused, are employed to block domain resolution. NXDomain, misleading the client into presuming domain nonexistence, dominates with 72.34%. Refused and ServFail, indicating server refusal and inability to fulfill the request, account for 17.76% and 9.9% respectively.

We note that 1,222 PDNSes apply diverse rewriting policies per domain category. For instance, 15 PDNSes (within ASN 50673) return a private-use IP (`0.0.0.0`) for malware and botnet domains, but redirect other malicious categories to secure IPs. This suggests these PDNSes may rank domains by maliciousness and assign fitting rewriting policies.

**Additional information in DNS responses.** Our analysis of 17k identified PDNS resolvers reveals that 3 embed blocking explanations in various DNS records. For instance, PDNS resolvers within ASN 4766 (Korea Telecom) return a `TXT` record of "`NX Service`" and a secure IP in a DNS `A` record.

To assess the impact of query types on PDNS, we conducted an additional scan experiment involving 4 common DNS requests (`AAAA`, `CNAME`, `TXT`, `NS`) towards 155 reputable DNS resolvers. We observe that 8 PDNSes return specific secure responses for corresponding query types. For `AAAA` and `CNAME` queries, 2 DNS vendors, including CIRA Canadian Shield DNS and CleanBrowsing, mirror the `A` record results. For `CNAME` records, 2 vendors, Yandex and SafeDNS, respond with the corresponding safe *CNAME*, while others do not. Five vendors apply similar rewriting policies to `AAAA` as to `A` records, e.g., Cloudflare returns `::` for `AAAA` and `0.0.0.0` for `A`, and SafeDNS returns an IPv6 with an embedded secure IPv4, like `::ffff:148.x.x.121`. However, the remaining 13 PDNS vendors respond with "insecure" results to non-configured query types, i.e., returning authoritative results, discussed further in Section V-C.

**PDNS groups.** Our empirical study in Section II reveals a high likelihood that PDNS providers using the same secure IP addresses are from the same DNS vendor, with an average of 2 secure IPs. Thus, we cluster the 17k PDNSes based on their shared secure IP addresses. Specifically, PDNSes with at least 2 identical secure IPs may belong to the same group. In total, we identify 12 groups containing over 50 PDNSes, with the top 5 presented in Table VIII, which were manually identified. We find that Group 1 and Group 5 use two controlled secure IPs within the same ASN as the PDNS' IPs. In contrast, US-based Groups 2~4 use secure IP addresses provided by Akamai [5], a cloud service company. It is worth noting that,

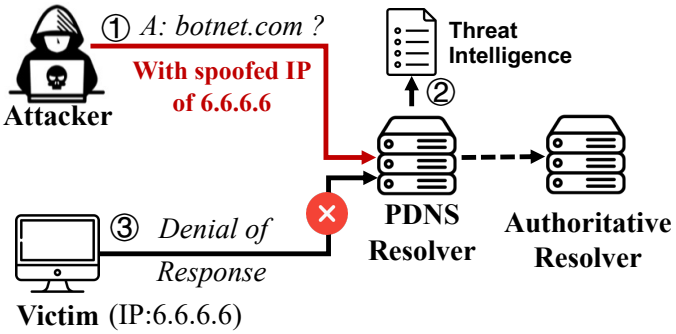


Fig. 7: Threat model of denial of response attack.

the largest group of PDNS is located in Oman, a result that may seem unusual, but could be tied to the intricate relationship between PDNS and country-level censorship. Oman is known for its extensive DNS filtering censorship [140]. Examination of Group 1's blocked domains reveals primary allocation to Malware (98 domains), Phishing (11 domains), and Adult content (8 domains). Adult content is explicitly subjected to Oman's national censorship. The challenge of distinguishing between PDNS and censorship remains a key limitation of this study, as illustrated by Group 1 in Oman, where censorship rewriting may still be conflated with current results.

## V. Security Issues

Our empirical study on 17k PDNSes uncovers 3 security risks arising from flawed blocking strategy implementations: (*i*) denial of responses (*DoR*) due to aggressive non-responsive policies, (*ii*) dangling cloud IPs susceptible to takeover and misuse by attackers, (*iii*) and multiple flawed blocking strategies implementations subversing PDNS protective features.

### A. Denial of Response

As stated in Section IV-E, 822 PDNSes employ *No Data* to block malicious domains. However, we find that this aggressive *No Data* approach can inadvertently impede normal domain resolution. Specifically, after querying multiple malicious domains, we found that *28 PDNSes* temporarily block all domain resolutions for the client, even those legitimate. We term this security risk as *denial of response* (*DoR*) induced by PDNS.

**Threat Model.** Attackers can exploit this security issue of PDNS to deny DNS resolution services for arbitrary victims by spoofing the source IP address. As depicted in Figure 7, we assume an off-path attacker incapable of eavesdropping on traffic between the client and resolver. To launch this attack, an attacker only needs a bulletproof hosting service that allows IP spoofing, currently permitted by over 30.5% of IPv4 ASes [99], [101]. Consistent with other works [124], we assume the victim can query the target PDNS, which accepts queries from any source IP. Specifically, the attacker *(1)* spoofs the victim's IP, sends malicious domain queries to the PDNS resolver, *(2)* triggers the PDNS resolver's blocking mechanism if these domains are on the blocklist, and *(3)* subsequently causes the PDNS to deny DNS lookup service to the victim for a certain period.

**Evaluation of DoR.** To validate the DoR of PDNS, we devise a series of experiments for verification and evaluation. In *test* experiments, we query each of the 28 potential PDNSes from 3 vantage points (UK, US, JP) for 10,000 malicious and 100 popular domains. We conduct 7 such experiments at varying time intervals (15m, 30m, 1h, 2h, 4h, 12h). Concurrently, we run *control* experiments from 2 other vantage points (AE, CH), querying 28 PDNS resolvers for 100 popular domains to assess their service availability to other users. We then tally the responses of popular and blocked malicious domains for each experiment. All 28 PDNS resolvers exhibit denial of response, evidenced by no responses for popular domains in the test experiment, contrasted by received results in the control. Even, 7 PDNS resolvers from renowned DNS vendors demonstrate DoR issues, with results in Table IX. Notably, after blocking 1,123 malicious domains, one PDNS resolver of ControlD DNS is unable to resolve any queries for up to 12 hours, even for common domains like `google.com`.

Ethical considerations precluded extensive actual attack testing. However, our evaluations attest to this attack's feasibility, evidenced by the absence of DNS resolution results from PDNS at test vantage points, contrasted with results at control points. Crucially, this denial-of-response attack illuminates best practices for PDNS implementation, detailed in Section VI.

### B. Dangling PDNS Infrastructure

As covered in Section IV-E, secure IPs are the prevalent rewriting policy for PDNS, employed by 56% of PDNSes. However, not-in-use "secure" IPs may be vulnerable due to the risks of dangling records (*Dare*)[97]. If a third-party attacker manipulates these Dare resources, they become insecure and exploitable. Three attack vectors to hijack these resources have been identified in existing works, involving expired domains[97], [134], [130], obsolete cloud IP addresses [16], [116], and third-party hosting services [97], [134].

**Threat Model.** The potential takeover and abuse of a PDNS's security-orientated policy by a third-party adversary could pose serious security implications. As depicted in Figure 8, we propose the attack workflow leveraging PDNS's dangling resources. Initially, we assume an attacker identifies a PDNS security policy through pre-testing, with controllable resources such as obsolete cloud IPs or expired sinkhole domains. Upon gaining control of a resource (e.g., IP `6.6.6.6`), the attacker can trigger DNS queries of malicious domains (`botnet.com`) to the PDNS resolver through phishing, etc. (steps *(1)* and *(2)*). Subsequently, PDNS returns modified results to the victim (step *(3)*), enabling victim-attacker server connections. Finally, the attacker can relay any malicious content to the victim (steps *(4)* and *(5)*).

**Evaluation of dangling DNS resources.** Following Liu's work [97], we assess this security risk in identified PDNSes. We select 8 renowned cloud platforms, including Amazon EC2 [10], Google Cloud [75], Cloudflare [27], Ali Cloud [76], Linode [98], Digitocean [41], and Microsoft Azure [103], to identify cloud-owned IP addresses. A comparison of ASNs and PTR records with these services reveals 61 cloud IPs employed by 693 PDNSes, blocking an average of 579 malicious domains. To identify seizable IPs, we use a two-step process,

TABLE IX: Example of well-known DNS vendors of Denial of Response

| Resolver | DNS Vendor | # Blocked Time | # Blocked Domain | # Malware | # Botnet | # Phishing | # Adult | # Spam | # Tracker |
|---|---|---|---|---|---|---|---|---|---|
| 76.76.2.1 | ControlD DNS | 12h | 1,123 | 1,073 | 24 | 17 | 5 | 4 | 0 |
| 156.154.71.3 | Neustar DNS | 15m | 538 | 390 | 58 | 63 | 22 | 4 | 1 |
| 156.154.71.2 | Neustar DNS | 15m | 76 | 50 | 3 | 15 | 3 | 4 | 1 |
| 64.6.65.6 | Verisign DNS | 15m | 440 | 395 | 20 | 11 | 9 | 5 | 0 |
| 199.85.126.10 | Norton DNS | 15m | 75 | 48 | 6 | 14 | 3 | 4 | 0 |
| 199.85.126.20 | Norton DNS | 15m | 82 | 44 | 7 | 16 | 9 | 6 | 0 |
| 199.85.126.30 | Norton DNS | 15m | 80 | 44 | 6 | 15 | 10 | 4 | 1 |

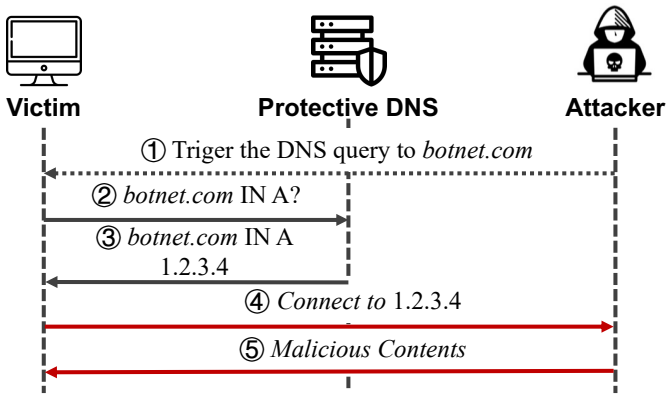


Fig. 8: Workflow of attacking dangling IP addresses.

we first assess host reachability using ICMP's echo mechanism (ping), and confirm port openness using the Censys [19] database, e.g., port 22 (SSH service). Unreachable IPs with closed common ports are deemed available for seizure. Our findings reveal 7 obsolete cloud IPs employed by *21 PDNSes*, averaging 279 malicious domain references.

Furthermore, we send DNS queries for 364 secure CNAME domains, identifying 5 returning `NXDomain`. Cross-checking with the registrar, Godaddy [74], reveals a seizable CNAME domain (`denied.gold`) impacting *5 PDNS resolvers* in Indonesia (ASN 138843, Goldnet).

Our experiments confirm the feasibility of attacks employing dangling DNS resources, particularly insecure IPs and CNAMEs, demonstrating the potential security implications of existing implementations of PDNSes.

### C. Subversion of PDNS

Beyond the exploitable security vulnerabilities, we expose 2 subversions of PDNS due to inadequate implementations, potentially nullifying its protective efficacy.

**Flawed implementations of PDNS.** We observe 105 PDNSes returning both forged (e.g., 127.42.0.148) and authoritative answers for malicious domain queries. Therefore, it is feasible for stub resolvers to select malicious resource records, leaving end users vulnerable to malicious activities. While the exact cause remains unclear, we hypothesize that PDNS operators adopt this approach to mitigate collateral damage from complete disablement of (erroneously) blocked domains.

**Non-configured query types of PDNS.** As mentioned in Section IV-E, we find 13 PDNS vendors returning original resolution results for types that are not configured with blocking measures. For example, 7 PDNSes return the original `TXT` records, e.g., Yandex and CIRA Canadian Shield DNS. This security issue could be exploited to bypass PDNS protection, particularly by malicious domains concealing harmful resolutions in uncommon request types.

### D. Vulnerability Disclosure

We responsibly disclosed security issues of PDNS service to DNS vendors. Among the identified vendors, we reported vulnerabilities to 14 vendors, with Verisign DNS, ControlD DNS, and Neustar DNS responding and actively discussing potential defenses. Despite thorough investigations with their ASN, PTR, and other details, we could not identify all affected resolver vendors. For vendorless resolvers, we collaborate with national CERT agencies, like the China National Vulnerability Database (CNVD) [108], for disclosure assistance. We have submitted 21 vulnerability reports for review. Up to now, 12 vulnerabilities regarding Denial of Response (DoR) have received audit verification results for *high-risk vulnerabilities*. For the other vulnerabilities, we are presently cooperating with them to connect with the service provider for problem resolution. To respect privacy, we avoid disclosing the information (IP addresses) of vendorless resolvers in this paper.

## VI. Discussion

### A. Ethics

We refer to previous research on DNS server probing and authoritative guidelines, e.g., the Belmont Report [71] and Menlo Report [88], and meticulously design each experiment step to mitigate the ethical risks of this work. Although our institution lacks an Institutional Review Board (IRB), our study has obtained authorization and supervision from our network management department. Below we describe how we designed the experiment according to authoritative principles [88] in detail.

The principle of Beneficence requires balancing potential benefits and harms. In our work, this principle is applied in 5 steps: 1) Select the range of test DNS servers. We need to test a set of DNS servers, initiate malicious domain queries, and identify PDNS based on their responses. In theory, we could test all DNS resolvers to find as many PDNS as possible and gain a comprehensive understanding. However,

initiating bulk malicious requests could pose risks to servers. Therefore, we use three criteria to limit the testing range to public, stable recursive DNS resolvers (see Step II), and avoid querying home devices. 2) Select measurement nodes (devices to initiate DNS queries). Although using query nodes with diverse geographical distribution can yield richer results, to avoid ethical risks, we have deployed measurement nodes only in countries with higher internet freedom [135]. 3) Select the test domain list. To find PDNSes, we need to trigger their blocking behaviors. However, to avoid ethical risks, we also need to control the total number of malicious domain queries. Therefore, we carefully diversify the categories and sources of malicious domains to trigger various PDNS protective strategies, while controlling total request numbers. 4) Instead of providing raw Netflow data, our provider performed aggregation by providing statistics at the network segment level, e.g., the count of unique (IP, port) tuples within each network segment (/24) and their total query numbers towards a certain DNS provider on a given day. While losing the original information of raw Netflow data and preventing us from knowing details of PDNS adoptions, such anonymization could better protect the privacy of users. 5) As for the open-sourcing of our code and results, we have not made all results public, but randomly sampled and disclosed the results of 5,000 resolvers to avoid potential security impacts. Specifically, we ensured not to disclose any resolvers related to discovered vulnerabilities. For sensitive results prone to takeover risks, e.g., secure IP addresses and CNAMEs, we anonymized them by applying MD5 hashing with a salt string of 10 random characters in length to compute their corresponding MD5 values (e.g., "$SELF\_USE\_MD5$").

The guidelines concerning law and public interest are applicable to all entities involved in our experiments. As our vantage points are cloud servers, initiating DNS queries for malicious domains presents potential risks in terms of security and censorship policies. Adhering to the guidelines of "law and public interest", we informed Alibaba Cloud of our study's goals, methods, potential risks, and benefits before starting the experiment. We obtained permission to utilize their servers, which were exclusively created for this study and comply with Alibaba Cloud's terms of use. To ensure minimal disruption to the servers' normal operations, we set a conservative query rate of 2 queries per second.

One important ethical principle of "respect for persons" needs extra attention in our measurement experiments, which emphasizes the need to respect the rights of potentially affected individuals as autonomous agents. In our study, we consider the tested DNS servers and their operators to be potentially affected by "humans". To uphold this principle, we set `PTR` records for our vantage points, clearly stating our research objectives and providing our contacts for opt-out purposes. Still now, we have not received any complaints about our experiment.

Finally, the principle of justice emphasizes equal benefits for all entities involved, particularly those undertaking associated risks. Our study is the first to depict the thriving PDNS ecosystem. In addition to shedding light on PDNS deployment and policies in the wild (see Section IV), we uncover security vulnerabilities in PDNS services, (see Section V). Taking responsibility, we promptly disclose all identified vulnerabilities to resolvers through our national CERT agency. Moreover, we believe that this disclosure will assist vulnerable resolvers in making timely fixes and improvements to enhance the security of their PDNS service. Based on our findings, we present recommendations for corrections and future deployments (see Section VI). Our study provides valuable insights into all entities involved in PDNS, including PDNS vendors (e.g., secure implementations of PDNS functionality), PDNS users (e.g., appropriate selection of PDNS services), and the DNS community (e.g., development of PDNS implementation guidance).

### B. Recommendations

In this section, we propose best practice recommendations for PDNS implementation. Our research indicates that PDNS is widely deployed in over 100 countries. Yet, due to the absence of standard guidelines, implementation flaws and security risks have left end users vulnerable. Most gravely, insecure blocking measures could be leveraged by attackers for illicit activities. Thus, we offer recommendations for PDNS implementation and deployment based on our measurement insights.

• *Transparent blocking activity.* Our findings suggest that using special IP addresses (e.g., `0.0.0.0`) is the safest policy, given the absence of identified security risks. However, this may impair user experience due to the perceived unresolvable domain names for no apparent reason. To mitigate PDNS false positives, we recommend setting up a web page to inform users of block reasons (e.g., Malware domain) and providing channels (e.g., email) for user complaints.

• *Utilizing safe rewriting infrastructures.* Utilizing third-party resources for rewriting infrastructure is convenient, and several PDNS operators adopt this approach. Nevertheless, dangling resources may introduce serious security issues exploitable by attackers for further malicious activities. Hence, we urge PDNS vendors to exercise increased caution when utilizing third-party resources like cloud IPs and sinkhole domains.

• *Defense of denial of response.* Although we do not advocate for aggressive non-responsive policies, they effectively combat genuine malware transmissions like C&C. We thus recommend PDNS operators implement defenses against DoR attacks. For instance, in response to clients issuing numerous DNS queries for malicious domains, PDNS operators can reply with a large DNS answer, forcing the client to use DNS over TCP, which also serves as a robust IP spoofing defense.

## VII. Related Work

The majority of DNS requests remain sent unencrypted. Due to the lack of encryption and authentication, adversaries can manipulate unprotected DNS traffic arbitrarily. Prior research has devoted significant effort to measuring the prevalence of DNS manipulation and understanding the underlying causes. Typically, their primary motivations include Internet censorship, malware distribution, and performance improvement.

To prohibit Internet users from accessing particular domains, DNS manipulation has become a widely adopted mechanism to conduct Internet censorship, including China [25], [117], Egypt [39], Iran [13], Pakistan [106], [89], and

Syria [21]. As an example, in 2015, a large-scale measurement study demonstrated that over 3 million DNS resolvers manipulated the response of certain domain names to a set of IP addresses for Internet censorship of 34 countries [90]. Leveraging the rogue DNS resolvers, adversaries are able to corrupt the DNS resolution path. By injecting advertisements [96] or distributing malware [35], the traffic interceptor gains illegal profits from DNS manipulation. In addition, transparent network middle-boxes and on-path devices can impersonate the IP address of popular DNS resolvers and intercept DNS queries to improve the performance of DNS resolution [96]. For instance, to improve the DNS root latency, several networks were found to deploy unauthorized DNS root server instances [146]. Also, the DNS queries targeted at well-known popular resolvers in hundreds of ASes were also exhibited being intercepted. Recent studies show that competent adversaries can tamper with DNS infrastructure and hijack Internet traffic to harvest credentials for target organizations [6], [37].

As stated previously, domain take-down operations are an effective weapon for combating cybercrime. Prior research has primarily examined the efficacy of take-down methods or how to detect sinkhole servers and IPs from domain blacklists [91], [107], [105], [104], [82], [14], [122]. In 2019, Alowaisheq *et al.* [8] conducted a systematic study to view the lifecycle of domain take-down operations. A number of weaknesses were also uncovered, such as misconfiguration of DNS records and expired sinkholes. The above studies show the ecosystem of domain take-down remains obscure.

As an emerging security service, PDNS is far less well-studied. Rodríguez *et al.* [123] explore the client-side adoption of PDNS, including whether users/ISPs want to use PDNS and the influencing factors. In contrast, we first study the server-side deployment of PDNS, including the extent of PDNS deployment among DNS providers, their blocking policies, and potential security issues. Compared to earlier studies, our study is the first to investigate the deployment and implementation of PDNS in the industry, serving as a supplement to the DNS ecosystem.

## VIII. Conclusion

In this paper, we investigate the implementation and security implications of worldwide PDNS services, which shield users from malicious domains, through a large-scale measurement study. After analyzing and summarizing characteristics of manually collected well-known PDNS vendors like Cloudflare and AdGuard DNS, we design and implement a system to identify potential open PDNS resolvers and open-source it for the community. We find 17,601 (9.08%) stable resolvers from numerous countries providing Protective DNS resolution services with varied domain filtering policies, thus safeguarding clients and maintaining minimal query latency overhead. However, we also uncover insecure PDNS implementations susceptible for attackers to subvert the PDNS infrastructure by hijacking queried domains and their clients or denying the normal resolution. Our work underscores the urgent need to review the implementation of PDNS services.

## Acknowledgment

We thank all the anonymous reviewers for their valuable comments to improve this paper. This research is supported by the National Key Research and Development Program of China (No. 2023YFB3105600), the National Natural Science Foundation of China (62102218, U1836213, U19B2034, 62302258), the Alibaba Innovative Research Program (AIR), CCF-Tencent Rhino-Bird Young Faculty Open Research Fund (CCF-Tencent RAGR20230116). Haixin Duan is supported by the Taishan Scholars Program. Yiming Zhang is partially supported by the Shuimu Tsinghua Scholar Program.

## References

[1] Cloudflared service not working with new family/malware protection dns 1.1.1.3 1.0.0.3? https://www.reddit.com/r/pihole/comments/fu64av/cloudflared_service_not_working_with_new/, 2019.

[2] Safedns safe@home free for 12 months. https://www.boards.ie/discussion/2057938161/safedns-safe-home-free-for-12-months, Accessed in Oct, 2022.

[3] 360 Secure Company. 360 Secure DNS. https://sdns.360.net/, 2023.

[4] Cybersecurity & Infrastructure Security Agency. Joint nsa and cisa guidance on strengthening cyber defense through protective dns. https://www.cisa.gov/uscert/ncas/current-activity/2021/03/04/joint-nsa-and-cisa-guidance-strengthening-cyber-defense-through, March, 2021.

[5] Akamai Technologies, Inc. Akamai. https://www.akamai.com/, 2022.

[6] Gautam Akiwate, Raffaele Sommese, Mattijs Jonker, Zakir Durumeric, KC Claffy, Geoffrey M. Voelker, and Stefan Savage. Retroactive Identification of Targeted DNS Infrastructure Hijacking. In *Proceedings of the ACM Internet Measurement Conference*, 2022.

[7] AliDNS. Alidns. https://alidns.com/, 2023.

[8] Eihal Alowaisheq, Peng Wang, Sumayah Alrwais, Xiaojing Liao, XiaoFeng Wang, Tasneem Alowaisheq, Xianghang Mi, Siyuan Tang, and Baojun Liu. Cracking the Wall of Confinement: Understanding and Analyzing Malicious Domain Take-downs. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, 2019.

[9] Omar Alrawi, Charles Lever, Kevin Valakuzhy, Ryan Court, Kevin Snow, Fabian Monrose, and Manos Antonakakis. The Circle Of Life: A Large-Scale Study of The IoT Malware Lifecycle. In *Proceedings of the 30th USENIX Security Symposium*, 2021.

[10] Amazon Web Services, Inc. Amazon EC2. https://aws.amazon.com/, 2022.

[11] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. From throw-away traffic to bots: Detecting the rise of dga-based malware. In Tadayoshi Kohno, editor, *Proceedings of the 21th USENIX Security Symposium, Bellevue, WA, USA, August 8-10, 2012*. USENIX Association, 2012.

[12] APNIC. Use of dns resolvers for world. https://stats.labs.apnic.net/rvrs, 2022.

[13] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. Internet Censorship in Iran: A First Look. In *Proceedings of the 3rd USENIX Workshop on Free and Open Communications on the Internet*, 2013.

[14] Hadi Asghari, Michael Ciere, and Michel J.G. van Eeten. Post-Mortem of a Zombie: Conficker Cleanup After Six Years. In *Proceedings of the 24th USENIX Security Symposium*, 2015.

[15] BlackWeb. Blackweb. https://github.com/maravento/blackweb/, 2022.

[16] Kevin Borgolte, Tobias Fiebig, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. Cloud Strife: Mitigating the Security Risks of Domain-Validated Certificates. In *Proceedings of the 25th Network and Distributed System Security Symposium (NDSS)*, 2018.

[17] Xander Bouwman, Victor Le Pochat, Pawel Foremski, Tom Van Goethem, Carlos H Gañán, Giovane Moura, Samaneh Tajalizadehkhoob, Wouter Joosen, and Michel van Eeten. Helping hands: Measuring the impact of a large threat intelligence sharing community. In *Proceedings of the 31st USENIX Security Symposium*. USENIX Association, 2022.

[18] Andrew Carr, Abu Alam, and Jordan Allison. Monitoring malicious dns queries: An experimental case study of utilising the national cyber security centre's protective dns within a uk public sector organisation. 2023.

[19] CENSYS. Censys. https://censys.io/, 2022.

[20] National Cyber Security Centre. Protective dns for the private sector. https://www.ncsc.gov.uk/guidance/protective-dns-for-private-sector, July, 2022.

[21] Abdelberi Chaabane, Terence Chen, Mathieu Cunche, Emiliano De Cristofaro, Arik Friedman, and Mohamed Ali Kaafar. Censorship in the wild: Analyzing internet filtering in syria. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, 2014.

[22] Taejoong Chung, David R. Choffnes, and Alan Mislove. Tunneling for transparency: A large-scale analysis of end-to-end violations in the internet. In Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan, editors, *Proceedings of the 2016 ACM on Internet Measurement Conference, IMC 2016, Santa Monica, CA, USA, November 14-16, 2016*, pages 199–213. ACM, 2016.

[23] Cybersecurity & Infrastructure Security Agency (CISA). Cisa launches its protective dns resolver with general availability for federal agencies. https://www.cisa.gov/blog/2022/09/27/cisa-launches-its-protective-dns-resolver-general-availability-federal-agencies, Sep 27th, 2022.

[24] Cisco. Opendns. https://www.opendns.com/, 2023.

[25] Richard Clayton, Steven J. Murdoch, and Robert N. M. Watson. Ignoring the Great Firewall of China. In *Proceedings of the 6th International Workshop on Privacy Enhancing Technologies*, 2006.

[26] CleanBrowsing. Cleanbrowsing dns. https://cleanbrowsing.org/, 2023.

[27] Cloudflare. Cloudflare. https://www.cloudflare.com/, 2022.

[28] CloudFlare. Cloudflare dns. https://1.1.1.1/dns/, 2023.

[29] Shanghai Stream Software Technology Co. Dns pai. https://www.dnspai.com/, 2023.

[30] Comodo. Comodo secure dns. https://www.comodo.com/securedns/, 2023.

[31] Comss.one. Comss.one dns. https://www.comss.ru/, 2023.

[32] ControlD. Controld dns. https://controld.com/free-dns/, 2023.

[33] Cybercrime. Cybercrime. http://cybercrime-tracker.net/, 2022.

[34] CZ.NIC. CZ.NIC ODVR DNS. https://www.nic.cz/odvr/, 2023.

[35] David Dagon, Niels Provos, Christopher P. Lee, and Wenke Lee. Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority. In *Proceedings of the Network and Distributed System Security Symposium*, 2008.

[36] Tianxiang Dai, Philipp Jeitner, Haya Shulman, and Michael Waidner. From IP to transport and beyond: cross-layer attacks against applications. In Fernando A. Kuipers and Matthew C. Caesar, editors, *ACM SIGCOMM 2021 Conference, Virtual Event, USA, August 23-27, 2021*, pages 836–849. ACM, 2021.

[37] Tianxiang Dai, Philipp Jeitner, Haya Shulman, and Michael Waidner. The Hijackers Guide To The Galaxy: Off-Path Taking Over Internet Resources. In *Proceedings of the 30th USENIX Security Symposium*, 2021.

[38] Tianxiang Dai, Haya Shulman, and Michael Waidner. Let's downgrade let's encrypt. In Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi, editors, *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pages 1421–1440. ACM, 2021.

[39] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. Analysis of Country-wide Internet Outages Caused by Censorship. In *Proceedings of the ACM Internet Measurement Conference*, 2011.

[40] Xavier de Carné de Carnavalet. *Last-Mile TLS Interception: Analysis and Observation of the Non-Public HTTPS Ecosystem*. PhD thesis, Concordia University, 2019.

[41] LLC. DigitalOcean. Digitocean. https://www.digitalocean.com, 2022.

[42] 114 DNS. 114 dns. https://www.114dns.com/, 2023.

[43] AdGuard DNS. Adguard dns. https://adguard-dns.io, 2023.

[44] Aha DNS. Aha dns. https://ahadns.com/, 2023.

[45] Alternate DNS. Alternate dns. https://alternate-dns.com/, 2023.

[46] CenturyLin DNS. Centurylin. https://www.centurylink.com/home/help/internet/dns.html, 2023.

[47] CyberGhost DNS. Cyberghost vpn. https://support.cyberghostvpn.com/hc/en-us/articles/360015373219-How-does-Smart-DNS-work-, 2023.

[48] Dyn DNS. Dyn dns. http://security-research.dyndns.org/pub/malware-feeds/, 2022.

[49] Dyn Public DNS. Oracle. https://help.dyn.com/internet-guide-setup/, 2023.

[50] FDN DNS. Fdn. https://www.fdn.fr/actions/dns/, 2023.

[51] Free DNS. Free dns. https://freedns.zone/en/, 2023.

[52] Freenom World DNS. Freenom. https://freenom.world/en/index.html, 2023.

[53] Google Public DNS. Google. https://dns.google/, 2023.

[54] Norton DNS. Norton dns. https://nortondns.com/, 2023.

[55] OpenNIC DNS. Opennic. https://www.opennic.org/, 2023.

[56] Privacy-First DNS. Privacy-first dns. https://tiarap.org/, 2023.

[57] Quad9 DNS. Quad9 dns. https://www.quad9.net/, 2023.

[58] SafeServe DNS. Namecheap. https://www.namecheap.com/dns/free-public-dns/, 2023.

[59] Safesurfer DNS. Safesurfer dns. https://safesurfer.io/, 2023.

[60] Snopyta DNS. Snopyta. https://snopyta.org/, 2023.

[61] Strongarm DNS. Strongarm dns. https://strongarm.io/, 2023.

[62] Switch Public DNS. Switch. https://www.switch.ch/security/info/public-dns/, 2023.

[63] Verisign Public DNS. Verisign. https://www.verisign.com/en_US/domain-names/internet-resolution/index.xhtml, 2023.

[64] DNSCrypt. Captnemo. https://captnemo.in/dnscrypt/, 2023.

[65] DNSFilter. Dnsfilter. https://www.dnsfilter.com/.

[66] DNSPerf. Dns performance analytics and comparison.

[67] DNS.Watch. Dns.watch. https://dns.watch/index, 2023.

[68] Rahel A. Fainchtein, Adam J. Aviv, Micah Sherr, Stephen Ribaudo, and Armaan Khullar. Holes in the geofence: Privacy vulnerabilities in "smart" DNS services. *Proc. Priv. Enhancing Technol.*, 2021(2), 2021.

[69] Márk Félegyházi, Christian Kreibich, and Vern Paxson. On the potential of proactive domain blacklisting. In Michael Bailey, editor, *3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET '10, San Jose, CA, USA, April 27, 2010*. USENIX Association, 2010.

[70] DNS for Family. Dns for family. https://dnsforfamily.com/, 2023.

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

[72] DNS Forge. Dns forge. https://dnsforge.de/, 2023.

[73] Fourth Estate DNS. Fourth Estate DNS. https://www.fourthestate.org/services/dns/, 2023.

[74] Godaddy. Search and buy domains in bulk. https://sg.godaddy.com/domains/bulk-domain-search, Accessed Janurary, 2022.

[75] Google. Google cloud. https://cloud.google.com/, 2022.

[76] Alibaba Group. Alibaba cloud services. https://www.alibabacloud.com/.

[77] Hispasec Sistemas Company. Virus Total. https://www.virustotal.com/gui/home/search. (Access in October, 2021).

[78] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. How great is the great firewall? measuring china's DNS censorship. In Michael Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 3381–3398. USENIX Association, 2021.

[79] Scott Hollenbeck. Extensible provisioning protocol (EPP). *RFC 5730*, 2009.

[80] Hosting24. Hosting24. https://www.hosting24.com/.

[81] G Huston, M Cotton, and L Vegoda. Iana ipv4 special-purpose address registry. https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml, 2010.

[82] Alice Hutchings, Richard Clayton, and Ross Anderson. Taking Down websites to Prevent Crime. In *Proceedings of 2016 APWG Symposium on Electronic Crime Research*, 2016.

[83] I-Blocklist. I-blocklist. https://www.iblocklist.com/, 2022.

[84] ICANN. Guidance for preparing domain name orders, seizures & takedowns, 2012.

[85] ICANN. Domain Abuse Activity Reporting (DAAR) System. https://www.icann.org/en/system/files/files/daar-monthly-report-31may23-en.pdf, 2023.

[86] Interkam. Interkam. https://www.interkam.pl/.

[87] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. Understanding the practices of global censorship through accurate, end-to-end measurements. In D. Manjunath, Jayakrishnan Nair, Niklas Carlsson, Edith Cohen, and Philippe Robert, editors, *SIGMETRICS/PERFORMANCE '22: ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, Mumbai, India, June 6 - 10, 2022*, pages 17–18. ACM, 2022.

[88] Erin Kenneally and David Dittrich. The menlo report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102*, 2012.

[89] Sheharbano Khattak, Mobin Javed, Philip D. Anderson, and Vern Paxson. Towards Illuminating a Censorship Monitor's Model to Facilitate Evasion. In *Proceedings of the 3rd USENIX Workshop on Free and Open Communications on the Internet*, 2013.

[90] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. Going wild: Large-scale classification of open dns resolvers. In *Proceedings of the 2015 Internet Measurement Conference*, 2015.

[91] Marc Kührer, Christian Rossow, and Thorsten Holz. Paint it Black: Evaluating the Effectiveness of Malware Blacklists. In *Proceedings of the 17th International Symposium on Recent Advances in Intrusion Detection*, 2014.

[92] Citizen Lab. Citizen lab. https://github.com/citizenlab/test-lists, 2022.

[93] Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. Reading the tea leaves: A comparative analysis of threat intelligence. In Nadia Heninger and Patrick Traynor, editors, *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*. USENIX Association, 2019.

[94] Xiang Li, Baojun Liu, Xuesong Bai, Mingming Zhang, Qifan Zhang, Zhou Li, Haixin Duan, and Qi Li. Ghost domain reloaded: Vulnerable links in domain name delegation and revocation. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society, 2023.

[95] Xiang Li, Baojun Liu, Xiaofeng Zheng, Haixin Duan, Qi Li, and Youjun Huang. Fast ipv6 network periphery discovery and security implications. In *51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2021, Taipei, Taiwan, June 21-24, 2021*. IEEE, 2021.

[96] Baojun Liu, Chaoyi Lu, Hai-Xin Duan, Ying Liu, Zhou Li, Shuang Hao, and Min Yang. Who is answering my queries: Understanding and characterizing interception of the DNS resolution path. In *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*. USENIX Association, 2018.

[97] Daiping Liu, Shuai Hao, and Haining Wang. All your dns records point to us: Understanding the security threats of dangling dns records. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[98] Linode LLC. Linode. https://www.linode.com, 2022.

[99] Matthew Luckie, Robert Beverly, Ryan Koga, Ken Keys, Joshua A Kroll, and K Claffy. Network hygiene, incentives, and regulation: deployment of source address validation in the internet. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[100] Owen Lystrup. Cisco security report: Majority of orgs do not monitor dns. https://umbrella.cisco.com/blog/cisco-security-report-more-orgs-should-be-monitoring-dns, 2020.

[101] Keyu Man, Zhiyun Qian, Zhongjie Wang, Xiaofeng Zheng, Youjun Huang, and Haixin Duan. Dns cache poisoning attack reloaded: Revolutions with side channels. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[102] MaxMind. Geoip2 databases. https://www.maxmind.com/en/geoip2-databases, Accessed May, 2022.

[103] Microsoft. Azure. https://azure.microsoft.com/, 2022.

[104] Tyler Moore and Richard Clayton. Examining the Impact of Website Take-down on Phishing. In *Proceedings of 2007 APWG Symposium on Electronic Crime Research*, 2007.

[105] Tyler Moore and Richard Clayton. The Impact of Incentives on Notice and Take-down. In *Proceedings of the 7th Workshop on the Economics of Information Security*, 2008.

[106] Zubair Nabi. The Anatomy of Web Censorship in Pakistan. In *Proceedings of the 3rd USENIX Workshop on Free and Open Communications on the Internet*, 2013.

[107] Yacin Nadji, Manos Antonakakis, Roberto Perdisci, David Dagon, and Wenke Lee. Beheading hydras: Performing Effective Botnet Takedowns. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, 2013.

[108] National Computer Network Emergency Response Technology Processing Coordination Centre (CNCERT). China national vulnerability database (cnvd). https://www.cnvd.org.cn, Accessed in September, 2023.

[109] Netflow. Isp netflow. https://www.cisco.com/c/en/us/products/collateral/ios-nx-ossoftware/ios-netflow/prod_white_paper0900aecd80406232.html., 2012.

[110] Neustar. Neustar ultradns public. https://www.publicdns.neustar/, 2023.

[111] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. Iclab: A global, longitudinal internet censorship measurement platform. In *2020 IEEE Symposium on Security and Privacy, SP 2020*. IEEE, 2020.

[112] Sadia Nourin, Van Tran, Xi Jiang, Kevin Bock, Nick Feamster, Nguyen Phong Hoang, and Dave Levin. Measuring and evading turkmenistan's internet censorship: A case study in large-scale measurements of a low-penetration country. In Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben, editors, *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 1969–1979. ACM, 2023.

[113] Cybersecurity Maturity Model Certification (CMMC) office. Cybersecurity maturity model certification (cmmc) standard (sc.3.192). https://www.cubcyber.com/cmmc/practices/sc-3-192, Accessed in July, 2022.

[114] OneDNS. Onedns. https://www.onedns.net/, 2023.

[115] Pierluigi Paganini. Microsoft partnered with security firms to sinkhole the c2 used in solarwinds hack, 2020.

[116] Eric Pauley, Ryan Sheatsley, Blaine Hoak, Quinn Burke, Yohan Beugin, and Patrick McDaniel. Measuring and mitigating the risk of ip reuse on public clouds. *arXiv preprint arXiv:2204.05122*, 2022.

[117] Abhishek Bhaskar Paul Pearce. Many roads lead to rome: How packet headers influence dns censorship measurement. In *Proceedings of the 31st USENIX Security Symposium*. USENIX Association, 2022.

[118] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nicholas Weaver, and Vern Paxson. Global measurement of DNS manipulation. In Engin Kirda and Thomas Ristenpart, editors, *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*. USENIX Association, 2017.

[119] Victor Le Pochat, Tom van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019.

[120] Sivaramakrishnan Ramanathan, Jelena Mirkovic, and Minlan Yu. BLAG: improving the accuracy of blacklists. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020.

[121] Audrey Randall, Enze Liu, Gautam Akiwate, Ramakrishna Padmanabhan, Geoffrey M. Voelker, Stefan Savage, and Aaron Schulman.

Trufflehunter: Cache snooping rare domains at large public DNS resolvers. In *IMC '20: ACM Internet Measurement Conference, Virtual Event, USA, October 27-29, 2020*. ACM, 2020.

[122] Mohammad Rezaeirad, Brown Farinholt, Hitesh Dharmdasani, Paul Pearce, Kirill Levchenko, and Damon McCoy. Schrödinger's RAT: Profiling the Stakeholders in the Remote Access Trojan Ecosystem. In *Proceedings of the 27th USENIX Security Symposium*, 2018.

[123] Elsa Rodríguez, Radu Anghel, Simon Parkin, Michel van Eeten, and Carlos Gañán. Two sides of the shield: Understanding protective DNS adoption factors. In Joseph A. Calandrino and Carmela Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*. USENIX Association, 2023.

[124] Christian Rossow. Amplification hell: Revisiting network protocols for ddos abuse. In *NDSS*, 2014.

[125] SafeDNS. Safedns. https://www.safedns.com/, 2023.

[126] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. On measuring the client-side dns infrastructure. In *Proceedings of the 2013 conference on Internet measurement conference*, 2013.

[127] Imperva Security. Ip blacklist. https://www.imperva.com/learn/application-security/ip-blacklist/, Accessed in September, 2022.

[128] National Security Agency/Central Security Service. Protective domain name system services. https://www.nsa.gov/About/Cybersecurity-Collaboration-Center/PDNS/, 2022.

[129] CIRA Canadian Shield. Cira canadian shield dns. https://www.cira.ca/, 2023.

[130] Wiz.io Shir Tamari, Ami Luttwak. New class of dns vulnerabilities affecting many dnsaas platforms. 2021.

[131] Ravindu De Silva, Mohamed Nabeel, Charith Elvitigala, Issa Khalil, Ting Yu, and Chamath Keppitiyagama. Compromised or Attacker-Owned: A Large Scale Classification and Study of Hosting Domains of Malicious URLs. In *Proceedings of the 30th USENIX Security Symposium*, 2021.

[132] SkyDNS. Skydns. https://www.skydns.ru/, 2023.

[133] Stop Forum Spam. Stop forum spam. https://www.stopforumspam.com/, 2022.

[134] Marco Squarcina, Mauro Tempesta, Lorenzo Veronese, Stefano Calzavara, and Matteo Maffei. Can i take your subdomain? exploring {Same-Site} attacks in the modern web. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[135] Statista. Degree of internet freedom in selected countries according to the freedom house index. https://www.statista.com/statistics/272533/degree-of-internet-freedom-in-selected-countries/, 2021.

[136] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Márk Félegyházi, and Chris Kanich. The long "taile" of typosquatting domain names. In Kevin Fu and Jaeyeon Jung, editors, *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014*. USENIX Association, 2014.

[137] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.

[138] Tencent. DNSPod Public DNS+. https://www.dnspod.com/, 2023.

[139] URLHaus. Urlhaus. https://urlhaus.abuse.ch/, 2022.

[140] US Department of State. 2022 country reports on human rights practices: Oman. https://www.state.gov/reports/2022-country-reports-on-human-rights-practices/oman/, 2022.

[141] Amber van der Heijden and Luca Allodi. Cognitive triaging of phishing attacks. In Nadia Heninger and Patrick Traynor, editors, *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*. USENIX Association, 2019.

[142] VX Vault. Vx vault. http://vxvault.net/, 2022.

[143] Ryan Vogt, John Aycock, and Michael J. Jacobson Jr. Army of botnets. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2007, San Diego, California, USA, 28th February - 2nd March 2007*. The Internet Society, 2007.

[144] Emma Woollacott. Canadian shield offers dns-based protection against malware and phishing attacks, July, 2021.

[145] Yandex. Yandex.dns. https://dns.yandex.com/, 2023.

[146] Fenglu Zhang, Chaoyi Lu, Baojun Liu, Haixin Duan, and Ying Liu. Measuring the Practical Effect of DNS Root Server Instances: A China-Wide Case Study. In *Proceedings of the 23rd International Conference on Passive and Active Measurement*, 2022.

[147] ZoneFiles. Zonefiles. https://zonefiles.io/detailed-domain-lists/, 2022.

## APPENDIX

### A. Survey of Non-Protective DNS Vendors

To gain more insights into Protective DNS, we select 42 famous DNS vendors based on market share [12] for empirical study. Table I shows detailed information on protective services for 28 DNS vendors, and Table X shows 14 DNS vendors without protective services with their countries.

TABLE X: Empirical study results of 14 public DNS vendors with no protective services.

| DNS Service | CC | DNS Service | CC |
|---|---|---|---|
| Google Public DNS [53] | US | Verisign Public DNS [63] | US |
| Dyn Public DNS [49] | US | Switch Public DNS [62] | US |
| DNS.WATCH [67] | US | OpenNIC DNS [55] | US |
| Snopyta DNS [60] | US | FDN DNS [50] | FR |
| Free DNS [51] | US | Fourth Estate DNS [73] | US |
| Freenom World DNS [52] | US | DNSCrypt [64] | US |
| CenturyLink DNS [46] | US | CyberGhost DNS [47] | US |

### B. Public Source of Malicious Domain Names

From an extensive survey of previous studies that use domain blocklists, we choose 7 blocklist sources. Table XI shows the source of malicious domain names. The blocklists comprise 5,226,699 domain names and cover 6 malicious categories: malware, phishing, tracker, botnet, adult and spam.

TABLE XI: Blacklists utilized to collect malicious domains.

| Blacklist | Categories | # of Unique Domains | Time Range |
|---|---|---|---|
| URLhaus [139] | Maware | 10,579 | 03/07/2022 - 06/07/2023 |
| CyberCrime [33] | Malware | 13,020 | 07/19/2012 - 06/07/2023 |
| ZoneFiles [147] | Malware, Phishing and Tracker | 113,978 | ~ - 06/07/2023 |
| BlackWeb [15] | Malware, Tracker, Botnet and Adult contents | 5,162,674 | 10/21/2016 - 06/07/2023 |
| I-Blocklist [83] | Malware, Ads and Spam | 82,610 | ~ - 06/07/2023 |
| Dyndns [48] | Malware, Phishing and Spam | 16,491 | 02/07/2013 - 06/07/2023 |
| Stop Forum Spam [133] | Abused Domain | 38,311 | ~ - 06/07/2023 |

### C. Evaluation Details

Below, we detail the 3 evaluation indicators used in Section III-B, where "positive" denotes PDNS and "negative" refers to non-PDNS.

• Precision signifies the proportion of samples predicted as positive (PDNS) that are indeed positive. It quantifies the ratio of "correctly identified PDNS servers" to all "identified PDNS servers", including potential false positives.

• Recall represents the proportion of accurately identified positive samples within all actual positive samples. It computes

the ratio of "correctly identified PDNS servers" to all "labeled PDNS servers".

- F1 score is a weighted average of precision and recall.

### *D. Geographical Distribution of Recursive DNS Resolvers*

In total, we identified 193,888 "stable" recursive DNS resolvers (Step II in Sec III-B) during a two-month scanning, covering 192 countries and 8,112 ASNs. Figure 9 presents the geographical distribution of recursive DNS resolvers, and Table XII lists the top 10 countries and ASes with the most recursive DNS resolvers.

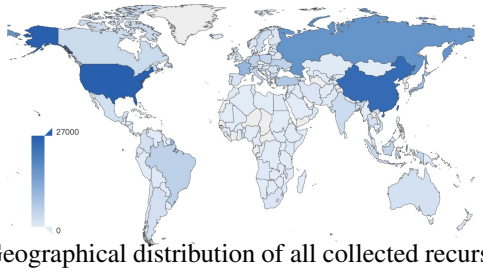


Fig. 9: Geographical distribution of all collected recursive DNS resolvers.

TABLE XII: Top 10 countries and ASes with the most recursive DNS resolvers.

| CC | # IP | ASN | # IP |
|---|---|---|---|
| US | 26,842 (13.8%) | 4837 (CHINA169-Backbone) | 7,750 (4.0%) |
| CN | 24,376 (12.5%) | 3462 (HINET) | 4,918 (2.5%) |
| RU | 16,753 (8.6%) | 4134 (CHINANET-BACKBONE) | 3,916 (2.0%) |
| FR | 10,197 (5.2%) | 9121 (TTNet) | 3,898 (2.0%) |
| JP | 8,570 (4.4%) | 16276 (OVH) | 3,654 (1.8%) |
| TR | 6,517 (3.3%) | 8866 (BTC-AS) | 3,077 (1.6%) |
| DE | 5,064 (2.6%) | 209 (CENTURYLINK-US) | 3,027 (1.6%) |
| BR | 4,890 (2.5%) | 3215 (Orange S.A.) | 2,934 (1.5%) |
| UA | 4,623 (2.4%) | 4713 (OCN) | 2,931 (1.5%) |
| IT | 4,360 (2.3%) | 12389 (ROSTELECOM-AS) | 2,537 (1.3%) |
| 192 Country | | 8,112 ASes | |

### *E. Geographical Distribution of PDNS Resolvers*

In total, we identify **17,601** (9.08%) PDNS resolvers in the wild. Figure 10 presents the geographical distribution of identified PDNS resolvers.

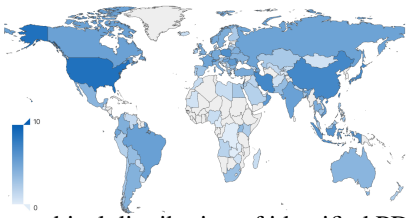


Fig. 10: Geographical distribution of identified PDNS resolvers (with the value after $log_e$).