Wolf in Sheep's Clothing: Evaluating Security Risks of the Undelegated Record on DNS Hosting Services

Fenglu Zhang*
Tsinghua University
Beijing, China
zfl23@mails.tsinghua.edu.cn

Yunyi Zhang*
National University of Defense Technology
Changsha, China
zhangyyzyy@nudt.edu.cn

Baojun Liu†
Tsinghua University
Beijing, China
lbj@tsinghua.edu.cn

Eihal Alowaisheq
King Saud University
Riyadh, Saudi Arabia
ealowaisheq@ksu.edu.sa

Lingyun Ying
QI-ANXIN Technology Research Institute
Beijing, China
yinglingyun@qianxin.com

Xiang Li
Tsinghua University
Beijing, China
x-l19@mails.tsinghua.edu.cn

Zaifeng Zhang
360 Security Technology Inc.
Beijing, China
zhangzaifeng@360.cn

Ying Liu
Tsinghua University
Beijing, China
liuying@cernet.edu.cn

Haixin Duan†
Tsinghua University; Quancheng Laboratory
Beijing, China
duanhx@tsinghua.edu.cn

Min Zhang
National University of Defense Technology
Changsha, China
zhangmindy@nudt.edu.cn

ABSTRACT

Leveraging DNS for covert communications is appealing since most networks allow DNS traffic, especially the ones directed toward renowned DNS hosting services. Unfortunately, most DNS hosting services overlook domain ownership verification, enabling miscreants to host undelegated DNS records of a domain they do not own. Consequently, miscreants can conduct covert communication through such undelegated records for whitelisted domains on reputable hosting providers. In this paper, we shed light on the emerging threat posed by undelegated records and demonstrate their exploitation in the wild. To the best of our knowledge, this security risk has not been studied before.

We conducted a comprehensive measurement to reveal the prevalence of the risk. In total, we observed 1,580,925 unique undelegated records that are potentially abused. We further observed that a considerable portion of these records are associated with malicious behaviors. By utilizing threat intelligence and malicious traffic collected by malware sandbox, we extracted malicious IP addresses from 25.41% of these records, spanning 1,369 Tranco top 2K domains and 248 DNS hosting providers, including Cloudflare and Amazon. Furthermore, we discovered that the majority of the identified malicious activities are Trojan-related. Moreover, we conducted case studies on two malware families (*Dark.IOT* and *Specter*) that exploit undelegated records to obtain C2 servers, in addition to the masquerading SPF records to conceal SMTP-based covert communication. Also, we provided mitigation options for different entities. As a result of our disclosure, several popular hosting providers have taken action to address this issue.

CCS CONCEPTS

• **Security and privacy** → **Network security**; • **Networks** → *Naming and addressing*.

KEYWORDS

Domain name system; DNS security; DNS hosting service

ACM Reference Format:
Fenglu Zhang, Yunyi Zhang, Baojun Liu, Eihal Alowaisheq, Lingyun Ying, Xiang Li, Zaifeng Zhang, Ying Liu, Haixin Duan, and Min Zhang. 2023. Wolf in Sheep's Clothing: Evaluating Security Risks of the Undelegated Record on DNS Hosting Services. In *Proceedings of the 2023 ACM Internet Measurement Conference (IMC '23), October 24–26, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3618257.3624839

1 INTRODUCTION

The domain name system (DNS) is one of the most fundamental infrastructures since numerous applications and security practices rely on DNS [20, 32, 39, 42]. Hence, DNS traffic is allowed, even on networks with stringent security requirements [15]. This fact attracts attackers to maintain malicious DNS infrastructure for covert communication.

Various studies have recently revealed techniques to maintain malicious DNS infrastructure. Some attackers set up servers and conceal malicious behaviors through DNS tunneling [34, 69]. However, the low reputation of these servers often exposes their activities. Alternatively, attackers can conduct domain shadowing [8, 47] or exploit dangling records [4, 46] to take over legitimate domains.

*Both authors contributed equally to the paper.
†Both authors are corresponding authors.

IMC '23, October 24–26, 2023, Montreal, QC, Canada

ACM ISBN 979-8-4007-0382-9/23/10.
https://doi.org/10.1145/3618257.3624839

However, these methods are insufficient to target well-managed domains.

Numerous mechanisms have also been developed to defend against emerging DNS-based attacks. Blacklisting is a well-known technique. However, it may inadvertently block legitimate services since attackers can share the same infrastructure with benign users [58, 61]. For accurate detection, reputation-based approaches, which block notorious domains and servers, have become popular [6, 22]. Also, some mechanisms that examine DNS traffic on the normal resolution, such as DNSSEC [31] and some advanced firewalls [14, 52], have been applied. Hence, adversaries have advanced toward more stealthy techniques.

In this paper, we uncover an emerging threat model of covert communication that *abuses the reputations of both popular domains and DNS hosting services*. Unlike some existing attacks, this threat model does not require establishing malicious servers or exploiting vulnerabilities in the targeted domain. Instead, it abuses the functionality of renowned DNS hosting providers. Specifically, DNS hosting providers (e.g., Cloudflare [17]) assist customers in managing their DNS records by assigning them nameservers. Typically, customers configure the assigned servers into the TLD zone to delegate the resolution. However, unlike CDNs and website hosting/development services that adopt ownership verification (e.g., in `TXT` records) commonly, DNS hosting providers tend not to verify the ownership of a domain, allowing attackers to host a domain with arbitrary DNS records on hosting services, even if they do not own the domain. We refer to such records as *Undelegated Record (UR)*. Hence, attackers can perform covert communications through records of an allowed domain provided by trusted nameservers. Other malicious campaigns (e.g., connecting to C2 servers) can be conducted using the information provided by URs. This attack can bypass existing reputation-based security mechanisms, as it exploits the reputation of both domains and providers. Furthermore, the detection based on examining normal resolution fails to identify the attack, as the malicious traffic associated with the attack does not rely on the default resolution.

We conducted a large-scale measurement and confirmed the URs are widely exploited in the wild. To this end, we designed a framework to collect URs and identify their malicious behaviors. Our measurement targeted 8,941 nameservers covering over 400 providers and tested whether they provided URs for the top 2K Tranco sites. After excluding cases such as past delegations, we discovered *1,580,925* unique URs that were potentially abused. By utilizing threat intelligence and malicious traffic collected by malware sandbox, we observed that malicious IP addresses could be found in *25.41%* of the URs. Such malicious URs accounted for *1,369* out of the top 2K Tranco sites and were hosted on *5,048* nameservers belonging to *248* providers, including popular ones such as Cloudflare and Amazon. We also found that the majority of the identified malicious activities were related to Trojan, which accounted for 41.67% of the malicious traffic and 89.01% of the malicious IP addresses. As case studies, we provided analysis on the variants of two malware families (*Dark.IOT* and *Specter*). Our analysis confirmed that both malware families utilized URs on ClouDNS to conceal their communications toward C2 servers. We also discovered masquerading `SPF` records that hid SMTP-based covert communication using Namecheap's nameservers. Our investigation revealed several renowned providers had adopted lenient hosting policies that even allow hosting domain suffixes belonging to government entities such as *gov.cn*.

Finally, we provided recommendations for network operators and hosting providers to mitigate the revealed threat. After our disclosure, Cloudflare, Tencent Cloud, and Alibaba Cloud have taken action to resolve this issue. We also make our artifacts publicly available [1].

2 BACKGROUND AND RELATED WORK

DNS namespace and delegation. DNS namespace is a tree structure with hierarchical delegation. Specifically, DNS root is at the top of the hierarchy and is partitioned into a series of Top-Level Domains (TLDs), such as *.com*. The next level is Second-Level Domains (SLDs), e.g., *example.com*. This domain is delegated to its authoritative servers by *.com*. Following the delegation, the authoritative server can respond with DNS records for the delegated domain.

DNS hosting service. DNS hosting is a third-party service that provides infrastructure for users to manage their domains. Two types of services are emerging. The first type only provides authoritative DNS servers to assist customers in managing DNS records (e.g., Godaddy [25]). The second type (e.g., Cloudflare [17] or WordPress [71]) provides both DNS servers and the infrastructure of upper applications, such as Web service hosting and Content Delivery Networks (CDN). These providers offer a comprehensive solution that helps customers simplify the maintenance of their domain services. DNS hosting services are becoming increasingly popular. Recent studies have shown that 89% of the top 100K websites utilize them [37, 50].

In a typical DNS hosting process, the service provider assigns some nameservers when a user requests to host a domain. Then, users set the assigned nameservers at the parent zone to *enable a delegation*. As a result, users can conveniently manage their domains' DNS records provided by the assigned nameservers through a user portal. The vendors that provide both DNS and upper applications infrastructures can even help users operate the upper application server directly without configuring DNS records.

Undelegated record (UR). Due to the hierarchical delegation of DNS, many DNS hosting providers have a relaxed policy when verifying the ownership of a hosted domain. Consequently, an adversary can host DNS records on a hosting service for any domain without owning the domain, including even some popular domains. We refer to such records as *undelegated records (URs)*, which can support various malicious activities. Note that the vendors offering both DNS and upper-layer infrastructure also provide URs since attackers can directly control upper-layer applications for malicious behaviors. In Section 5, we demonstrate the widespread abuse of URs and present evidence of related malware.

Related work. Extensive research has studied the related methodologies to maintain DNS infrastructure for malicious activities, including *establishing servers*, *domain takeover*, and *CDN exploitation*. A straightforward method for attackers is establishing their malicious servers. To avoid detection, attackers can also perform DNS tunneling by encoding their attack payload into domains [34] or

[1] https://github.com/zhangshanfen9/imc-ur

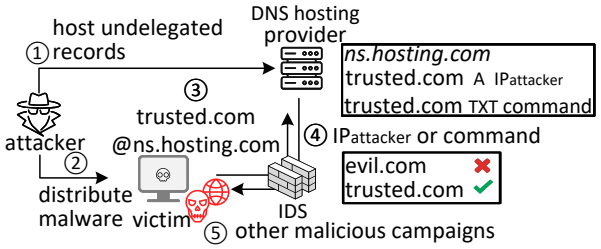


Figure 1: Threat model of URs.

DNS records [69]. Another method is dedicated to taking over existing domains. Some attackers perform domain shadowing to compromise legitimate domains and spawn subdomains under them [8, 47]. By exploiting dangling records (stale records), attackers can also take control of benign domains [4, 46], abuse email services [28], and even obtain SSL certificates [10]. The exploitation of CDN, such as domain fronting [23], hiding [12], and borrowing [67], is also an option to hide attackers' infrastructure. Such technologies leverage the characteristics of CDNs while forwarding HTTPS requests and further hide malicious communication behind benign traffic.

In this paper, we revealed an exploitation technique that can leverage well-known service providers and reputable domain names. The distinction between the uncovered attack and previous ones is outlined in Section 3.

3 THREAT MODEL

Covert communication plays a crucial role in criminal campaigns. This paper exposes an emerging method for covert communications by abusing renowned DNS hosting services.

Attack overview. The attack aims to achieve covert communication. Figure 1 illustrates the threat model. The victims are infected hosts that require further instructions from the attacker but are protected by a defense mechanism (e.g., IDS). To this end, the attacker first configures URs for a trusted domain (e.g., *trusted.com*) at a nameserver (e.g., *ns.hosting.com*) that belongs to a reputable hosting provider (①). The URs can provide information that the attacker expects the victim to receive (e.g., an `A` record with an IP address of a C2 server, or a `TXT` record with an encrypted command). Then, the attacker delivers the malware to the victim (②). According to the pre-issued instructions, the malware requests the resolution of the reputable domain from the provider's nameservers, which return the URs (③). Since both the abused domain and nameserver are reputable, the connection for URs can evade the protection (④). Utilizing the information in URs, the victim can further perform other malicious activities, such as connecting to C2 servers (⑤).

Assumptions. Attackers need two conditions to exploit URs: *(1) The hosting provider does not verify the ownership of hosted domains.* Through a large-scale measurement (Section 5) and a case study (Appendix C), we found that even leading providers (e.g., Cloudflare and Amazon) allow URs. *(2) Defense mechanisms do not block the DNS traffic from victims to the hosting provider.* Blocking such DNS traffic may inadvertently disrupt legitimate activities. This is because the traffic related to URs is hard to differentiate from benign DNS traffic, such as the traffic generated from configuring custom DNS resolvers (e.g., Google Public DNS [27]). Also, blocking such traffic toward hosting providers may cause outages since hosting services contribute to the centralization of internet traffic [29, 37, 50].

Advantages. The attack using URs offers several advantages over some existing attacks. Specifically, abusing URs does not require attackers to establish any infrastructure (e.g., domains or servers). Unlike taking over domains, exploiting URs does not require compromising the targeted domain (domain shadowing [8, 47]) or abusing dangling records [4, 46], making it a more feasible option even for well-managed domains.

The uncovered attack can also bypass several types of existing defense mechanisms. In particular, several defense mechanisms have been developed to filter out abnormal DNS traffic associated with notorious domains and directed toward malicious servers. Some solutions use reputation-based approaches (e.g., [6, 22]), while others rely on the characteristics of benign domain and nameserver (e.g., [9, 13, 41]) Unfortunately, URs capitalizing on the reputation of popular domains and service providers can bypass such protections. Another type of defense mechanism (e.g., DNSSEC [31] and some advanced firewall [14, 52]) focuses on examining the DNS traffic following the normal resolution. However, these mechanisms fail to detect the threat of URs since the traffic associated with URs does not rely on the default resolver.

4 METHODOLOGY

This section details our automated framework, named URHunter, to measure URs and related malicious campaigns.

Measuring malicious URs is not straightforward and requires addressing two challenges: ① *Determining potentially abused URs.* To this end, several cases should be removed. For example, misconfigured nameservers may provide URs through recursive resolution [57]. A past delegation can also result in URs remaining on providers no longer utilized. Besides, a hosting service may provide protective records (e.g., to a website with warning information) for undelegated domains. Therefore, URHunter should exclude cases not resulting from abuse. ② *Collecting and identifying malicious URs.* As a type of covert communication, the exact deployment and usage of URs remain unknown, and the metric of malicious URs should be carefully designed. To address the challenges, we designed URHunter with three components, which are described as follows.

4.1 Response collection

This component collects responses to obtain URs for further analysis (Challenge ②). In our measurement, we targeted the top domains and nameservers since attacks of UR abuse the reputation of popular domains and hosting providers. To this end, we selected nameservers that host more than 50 domains in the top 1M Tranco sites. Such nameservers often belong to renowned hosting providers and are preferred targets for attackers. Also, we selected the top 2K Tranco sites as our target domains. For efficient measurement, both ZDNS [35] and XMAP [44] satisfied our requirements, and URHunter collects the following responses with the support of ZDNS.

(1) Undelegated record. To collect URs, URHunter queries every selected nameserver for each targeted domain but excludes the domains exactly delegated to the nameserver. URHunter extracts URs from the `NOERROR` responses. Then, it collects *additional information* on each undelegated `A` record and stores it in a database for further

analysis: For each IP address in undelegated A records, URHunter obtains its autonomous system (AS) and geographic location by referring to the IP information database [49]. It also collects the HTTP responses and TLS certificate of each IP address.

(2) Correct record. We define the URs caused by recursive resolution and past delegation as *correct records*. Such records may be geo-distributed due to load balancing techniques (e.g., CDN). Also, the records of past delegations could differ from existing delegated records (e.g., switching service providers).

URHunter collects the correct records of each targeted domain for further analysis. Specifically, it collects geo-distributed correct records by leveraging open resolvers. Previous studies [53, 57, 64] utilized worldwide vantage points and observed that most of them did not return manipulated results. Inspired by this, URHunter selects 3K open resolvers worldwide and requests the A and TXT records of targeted domains[2]. Then, URHunter collects the additional information for every correct A record. It also utilizes historical DNS resolution to collect the correct records due to past delegations. We collaborated with one of the largest DNS providers in the world and collected all historical delegated records in the last six years from passive DNS data.

(3) Protective record. URHunter collects the URs that providers use as a protective measure (e.g., point to a website with alerting message). To this end, URHunter sends queries for our domain not hosted on any targeted nameservers and extracts the protective records in the response.

4.2 Determining suspicious record

In this stage, URHunter determines potentially abused URs, which can be mixed with cases that should be excluded (Challenge ①). To this end, URHunter excludes correct and protective records from previous measurement results. The protective records, which have been collected by URHunter, can be directly matched and excluded; however, no existing metric is designed to identify correct records.

We utilized the concept of *uniformity* to identify correct records. Studies that detected DNS hijacking have shown that the IP address information (e.g., AS, location, and TLS certificate) for a specific domain tends to be uniform, as it is typically managed by the same organizations [2, 57, 64, 68]. Also, different parts of HTTP responses can expose the purpose of a site [54, 72] and helps in excluding the URs pointing to parked or redirection pages. As a result, URHunter can label an undelegated A record as correct when it matches the conditions detailed in Appendix B. In addition, URHunter excludes correct TXT records that exactly match the correct records in the database. By matching regular expression, URHunter further classifies the undelegated TXT records according to the known categories [69].

We evaluated whether URHunter generated false-negative results. The evaluation follows the same steps of excluding correct and protective records but takes the *delegated records* of the top 2K Tranco sites as input instead of URs. The result showed that no input is labeled as a suspicious UR, indicating that URHunter achieved a zero false-negative rate.

[2]We believe this design is sufficient because even though ECS is enabled on most top domains, only a few of them return diverse IP addresses [40].

4.3 Malicious behavior analysis

The last component identifies the malicious URs and associated malicious behaviors. To ensure accuracy, we utilized off-the-shelf tools that have been widely used instead of developing our method. So, we utilized the threat intelligence from VirusTotal [70], QAX [59], and 360 Security [62], in which all maintain real-time updates on the IP address blacklists. Also, VirusTotal and QAX provide sandbox evaluation reports for millions of malware. All network traffic generated by each malware was collected and analyzed.

To reveal the exact usages of URs that have not been studied before (Challenge ②), we extracted the corresponding IP address of a UR and marked a UR as malicious for two reasons: (1) Threat intelligence explicitly labels an IP address as malicious. (2) IDS (Snort [16] or Suricata [55]) detects malicious traffic toward the IP address in a malware sandbox evaluation. Note that we only consider malicious traffic with a severity level of at least medium, excluding cases where malware only checks network connectivity. According to the two reasons, A records were classified based on their IP address. On the other hand, TXT records were labeled based on the IP addresses embedded in the resource data. We also checked whether an A and a TXT record are hosted on the same nameserver and serve the same domain. If satisfied, the IP address of the A record will be included as a corresponding IP address of the TXT record. We excluded the TXT records without corresponding IP addresses. Then, we labeled a UR malicious when its corresponding IP address was malicious. As a result, URHunter classifies URs into four categories: *malicious*, *correct*, *protective*, and *unknown*, with the latter representing the remaining records.

5 RESULT ANALYSIS

In this section, we first present an overview of URs from our measurement results. Then, we perform an in-depth analysis of the malicious activity on URs. Finally, we provide case studies to reveal their exploitation in the wild.

5.1 Overview of undelegated records

In Apr and Dec 2022, we conducted two large-scale measurements of undelegated A and TXT records, respectively. From Tranco Top 1M sites, URHunter obtained 8,941 IP addresses of nameservers hosting over 50 domains. More than 400 providers operated these servers. Then, URHunter collected URs for the top 2K Tranco domains from the 8,941 nameservers. We define a unique UR as a DNS record provided by a nameserver (IP address) for an undelegated domain since it gives a unique option for attackers to retrieve information. For example, even the same UR hosted in two nameservers provides two options for attackers since blocking one server is not enough to stop resolving the UR. After analyzing 23 million DNS responses, URHunter classified 5,011,483 unique URs. Figure 2 shows the categories and distribution of URs among the vendors with the most URs. While correct and protective records make up a significant portion, the presence of malicious and unknown URs should not be ignored, as they indicate potential exploitation. We categorized these records as suspicious and identified *1,580,925* suspicious URs.

Upon closer inspection of the suspicious records, we observed a significant portion of them labeled as malicious (Table 1). We confirmed that *401,718 (25.41%)* of the suspicious records are indeed

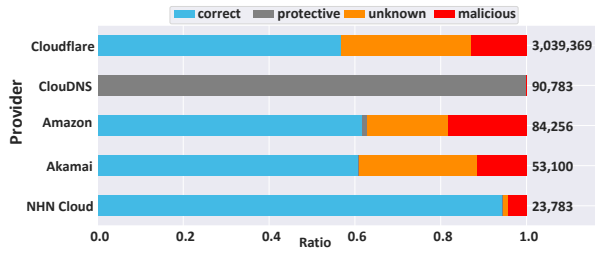


Figure 2: Categories and proportions of URs among the top five vendors with the most URs.

malicious. These malicious records were found across *1,369 (68.48%), 5,048 (79.48%), and 248 (71.47%)* of the top 2K sites, nameservers, and providers, respectively. The affected nameservers even belong to renowned hosting providers like Cloudflare [17], Amazon (Route 53 [5]), Akamai [1], and ClouDNS [18].

5.2 Analysis of malicious activities

We first extracted IP addresses from URs and marked IP addresses as malicious based on threat intelligence and IDS (Section 4.3). Among all malicious URs, 63.38% (34.20 + 29.18) of them were labeled since they involved malicious IP addresses that had been identified by security vendors directly. 65.80% (36.62 + 29.18) of them were marked since IDSes detected malicious traffic toward the UR-involved IP addresses. Besides, it's worth noting that 29.18% of the identified IP addresses met both conditions (Figure 3(a)). Zooming into the malicious IP addresses flagged by security vendors directly, we found some IP addresses are flagged by multiple security vendors, with instances flagged by up to 11 vendors (Figure 3(b)).

We then analyzed the malicious IP addresses extracted from URs to reveal the exact use of URs. When IDSes detect malicious traffic toward IP addresses in URs, they report various malicious activities, such as Trojan activity, that match predefined rules (Figure 3(c)). Additionally, security vendors provide supplementary information (tags), such as involving Botnet, while marking a malicious IP address (Figure 3(d)). This allowed us to identify the malicious activities related to URs. Notably, the most prevalent malicious activities (41.67% of alerts and 89.01% of tags) were associated with Trojans, which disguised themselves as legitimate software to compromise the victim's machines [65]. A significant number of IP addresses involved in scanning activities (41.01%) were also observed. Such a result is not surprising, given that reconnaissance is typically the initial stage of an attack. A malicious server could serve multiple functions such as scanning for vulnerabilities and conducting Trojan-related activities. In addition, we also found a notable proportion of IP addresses being used for C2 activities (10.82% of alerts and 16.25% of tags).

We also observed the utilization of malicious `TXT` records for covert communication. Remarkably, 90.95% of these records were acting as email-related DNS records (`SPF` and `DMARC`). Such a statistical result echoes our findings in a case study on a masquerading `SPF` record (Section 5.3).

5.3 Case Study

We provide case studies of several malicious behaviors, including malware families exploiting ClouDNS and masquerading `SPF` records. We also investigate the hosting strategy of several popular providers to reveal the variety of attacking options (e.g., allowed types of hosted domains).

Two malware families that exploit ClouDNS to obtain C2 servers. Due to inspecting sandbox traffic does not disturb any real-world servers, we included all FQDNs of the top Tranco 2K sites as tested domains to find the malware families that comprehensively exploit URs. Based on reports of our sandboxes, we manually analyzed two malware families (*Dark.IoT* and *Specter*) that carry covert communications for C2 servers through ClouDNS, which is a renowned vendor providing numerous protective records (Figure 1).

Dark.IoT [60], a malware family exploiting IoT devices, abuses URs on ClouDNS over a long period. We found two variants released on Dec 12, 2021, which query ClouDNS to resolve *api.gitlab.com* (whose SLD rank is 527 in Tranco). This UR pointed to their C2 server. The attackers also relied on EmerDNS[21] to host their OpenNIC domains. Interestingly, we observed a shift in their technique in the latest variant released on Mar 4, 2023, as they now rely on URs and have recently abandoned the use of EmerDNS. The attacker hosted their OpenNIC domains on ClouDNS and abused URs instead of utilizing EmerDNS resolution. Also, the malware switched to another popular domain (*raw.pastebin.com*, whose SLD rank is 2033 in Tranco). This behavior indicates that attackers perceive the reliability of URs.

We also found three variants of *Specter* [63], a type of Remote Access Trojan (RAT), exploited ClouDNS to maintain the connection with C2 servers. The malware leveraged URs for two popular domains (*ibm.com* and *api.github.com*, whose SLD ranks are 125 and 30 in Tranco, respectively). They have not been flagged yet as malicious by 74 mainstream security vendors (aggregated by VirusTotal).

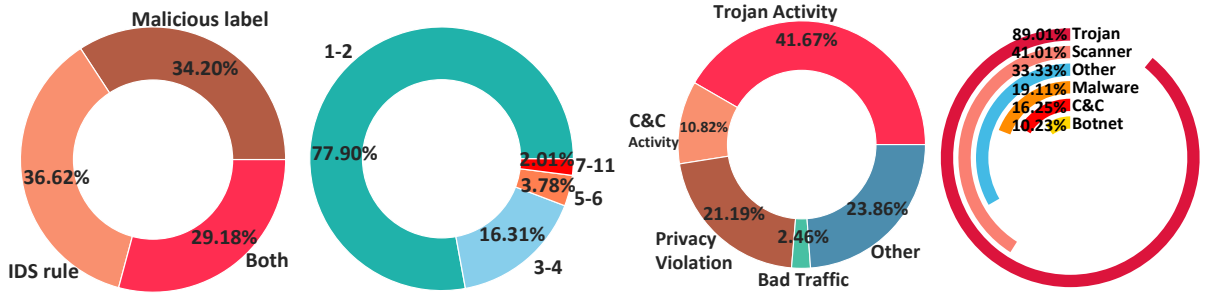
Masquerading `SPF` records hiding SMTP-based covert communication. We observed that masquerading `SPF` records for *speedtest.net* (rank 415 in Tranco) provided IP addresses of malicious servers, and malware further connects to the malicious servers through SMTP-based convert communication. The records were deployed on 11 nameservers belonging to two hosting providers (Namecheap [51] and CSC [19]). They contained three IP addresses in the same /24. All of the IP addresses are labeled as malicious by threat intelligence.

By inspecting sandbox traffic, we identified the malicious behaviors associated with the UR. We discovered six malware samples that triggered 16 alerts in the IDS. Five malware are labeled as Trojan by threat intelligence while the remaining one is classified as harmless by all 74 security vendors. Moreover, IDS flagged four of the related traffic as high-risk traffic. Micropsia Trojans generated two C2-related traffic, while Tesla Trojans generated others for SMTP-based covert communication. Such SMTP-based covert communication can be highly concealed with the abuse of masquerading `SPF` URs. Although not all of the URs related to the analyzed malware families can be resolved, the masquerading records can still be resolved at the time of writing this paper.

Various attacking options granted by the hosting strategy of providers. To determine the attacking options of UR (e.g., allowed domains), we investigated several renowned providers (detailed in Appendix C). Unfortunately, the leading hosting providers, including Cloudflare and Amazon, allow URs. Even worse, attackers can host some public domain suffixes for government entities and

Table 1: Overview of suspicious undelegated records (excluding correct and protective records).

Category	# Domain		# Nameserver		# Provider		# Undelegated record		# IP address	
	Total	Malicious	Total	Malicious	Total	Malicious	Total	Malicious	Total	Malicious
A	1,999	1,353 (67.68%)	6,262	4,981 (79.54%)	347	241 (69.45%)	1,366,164	395,095 (28.92%)	5,477	1,329 (24.27%)
TXT	448	221 (49.33%)	3,664	3,234 (88.26%)	102	67 (65.69%)	214,761	6,623 (3.08%)	1,147	273 (23.80%)
Total	1,999	1,369 (68.48%)	6,351	5,048 (79.48%)	347	248 (71.47%)	1,580,925	401,718 (25.41%)	6,346	1,494 (23.54%)



(a) Reasons why IP addresses being labeled.

(b) # Security vendors reported that the IP address is malicious

(c) Malicious activities detected in the traffic toward malicious IP addresses

(d) Tags from security vendors (an IP address may be with multiple tags simultaneously)

Figure 3: Analysis of malicious IP addresses corresponding to URs

educational institutions (e.g., *gov.cn* and *edu.cn*), which provides valuable options for their malicious activities.

6 DISCUSSION AND CONCLUSION

Mitigation. We provided mitigation suggestions for different entities and responsibly disclosed the issues to most hosting providers mentioned in this paper.

We believe network operators should take responsibility for mitigating the risk since one of the root causes is the lack of reviewing DNS traffic from clients to other DNS servers. To mitigate the risks, operators should give extra consideration to the DNS traffic that does not follow the recursive process and avoid overreliance on reputation-based detection.

We also recommend that hosting providers take steps to verify the ownership of hosted domains through the following options: (1) Verifying whether the NS records from TLD are pointing to the assigned nameservers. (2) Verifying the control of the hosted domain's zone (e.g., requiring configuring a randomly generated TXT record).

We have responsibly disclosed our findings to most of the mentioned providers in this paper. Accordingly, Tencent Cloud (DNSPod) fully adopted option (1). Cloudflare also expanded the blacklist of hosted popular domains. Alibaba has partially adopted option (2) and verified the control of the subdomain zone. However, by examining possible abuse options described in Appendix C, we found that Cloudflare and Alibaba are still exploitable, but available renowned domains become fewer. ClouDNS replied that they would make sure to take appropriate actions.

Limitations and future work. (1) Our study may miss some exploited URs since we only selected limited top SLDs and vendors. However, the selected targets are more valuable to be abused due to their popularity. We will measure more domains and nameservers in our further research. Specifically, we can recover legitimate subdomains from PDNS data and measure whether they appear in URs. Our methodology is also adaptive for measuring more nameservers and other types of records (e.g., MX records). (2) There may be under-reporting in our analysis since malicious exploitation of URs has not received widespread attention, resulting in the lack of information from security vendors. We also excluded the TXT URs lacking IP addresses since we cannot identify whether they were malicious (e.g., encrypted TXT URs) through existing data. However, as the first study for URs, we have identified 401,718 malicious records covering 1,369 top sites and 248 providers. We believe that matching the TXT URs without IP addresses with existing malware payloads is a valuable direction for future work.

Conclusion. This paper conducts the first large-scale measurement of URs and exposes the exploitations in the wild. Using URHunter, we discovered numerous URs and related malicious behaviors across renowned domains and DNS hosting services. After disclosure, several vendors have taken action to fix the security issue. We believe this work provides in-depth insights into the threat of URs and helps different entities defend against the attack.

ACKNOWLEDGMENTS

We thank the shepherd and all anonymous reviewers for their valuable and constructive feedback. This research is supported in part by the National Key R&D Program of China (2021YFB3100500), the National Natural Science Foundation of China (62102218), Alibaba Innovative Research Program (AIR), CCF-Tencent Rhino-Bird Young Faculty Open Research Fund (CCF-Tencent RAGR20230116), and Tsinghua University-China Telecom Corp., Ltd. Joint Research Center for Next Generation Internet Technology Research Fund. We also express our gratitude to Lu Liu from the QI-ANXIN Technology Research Institute for his irreplaceable contribution. Additionally, we extend our appreciation to VirusTotal for their support of our academic API.

REFERENCES

[1] Akamai. 2023. Edge DNS-Secure DNS Solution and Edge Platform. https://www.akamai.com/products/edge-dns.

[2] Gautam Akiwate, Raffaele Sommese, Mattijs Jonker, Zakir Durumeric, KC Claffy, Geoffrey M. Voelker, and Stefan Savage. 2022. Retroactive Identification of Targeted DNS Infrastructure Hijacking. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 14–32. https://doi.org/10.1145/3517745.3561425

[3] Alibaba. 2023. Domain Name Service. https://www.alibabacloud.com/domain.

[4] Eihal Alowaisheq, Siyuan Tang, Zhihao Wang, Fatemah Alharbi, Xiaojing Liao, and XiaoFeng Wang. 2020. Zombie Awakening: Stealthy Hijacking of Active Domains through DNS Hosting Referral. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) *(CCS '20)*. Association for Computing Machinery, New York, NY, USA, 1307–1322. https://doi.org/10.1145/3372297.3417864

[5] Amazon. 2023. Amazon Route 53|DNS Service. https://aws.amazon.com/route53/.

[6] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. 2010. Building a Dynamic Reputation System for DNS. In *Proceedings of the 19th USENIX Conference on Security* (Washington, DC) *(USENIX Security'10)*. USENIX Association, USA, 18.

[7] Baidu. 2023. Baidu AI Cloud. https://intl.cloud.baidu.com/.

[8] Nick Biasini and J Esler. 2015. Threat spotlight: Angler lurking in the domain shadows.

[9] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis.

[10] Kevin Borgolte, Tobias Fiebig, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. [n. d.]. Cloud Strife: Mitigating the Security Risks of Domain-Validated Certificates. In *Proceedings of the 25th Network and Distributed System Security Symposium (NDSS)* (25 ed.) (San Diego, CA, USA, 2018-02), Patrick Traynor and Alina Oprea (Eds.). Internet Society (ISOC). https://doi.org/10.14722/ndss.2018.23327

[11] Jonas Bushart and Christian Rossow. 2020. Padding Ain't Enough: Assessing the Privacy Guarantees of Encrypted DNS. In *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*. USENIX Association. https://www.usenix.org/conference/foci20/presentation/bushart

[12] Catalin Cimpanu. 2023. DEF CON: New tool brings back 'domain fronting' as 'domain hiding'. https://www.zdnet.com/article/def-con-new-tool-brings-back-domain-fronting-as-domain-hiding/.

[13] Daiki Chiba, Takeshi Yagi, Mitsuaki Akiyama, Toshiki Shibahara, Takeshi Yada, Tatsuya Mori, and Shigeki Goto. 2016. DomainProfiler: Discovering domain names abused in future. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 491–502.

[14] Cisco. 2023. Cisco Umbrella|Leader in Cloud Cybersecurity SASE Solutions. https://umbrella.cisco.com/.

[15] Cisco. 2023. DNS-Layer Security: The Ultimate Guide to What It Is and Why You Need It. https://umbrella.cisco.com/blog/what-is-dns-layer-security.

[16] Cisco. 2023. Snort-Network Intrusion Detection & Prevention System. https://www.snort.org/.

[17] Cloudflare. 2023. Cloudflare, Inc. - Investor Relations. https://cloudflare.net/home/default.aspx.

[18] ClouDNS. 2023. Free DNS hosting, Cloud DNS hosting and Domain names. https://www.cloudns.net/.

[19] CSC. 2023. DNS Services | Managed DNS Security - CSC. https://www.cscdbs.com/en/domain-security/dns-services/.

[20] Viktor Dukhovni and Wes Hardaker. 2015. The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance. RFC 7671. https://doi.org/10.17487/RFC7671

[21] Emercoin. 2022. EmerDNS - Blockchain Service. https://emercoin.com/en/emerdns/.

[22] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. 2010. On the Potential of Proactive Domain Blacklisting. In *Proceedings of the 3rd USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More* (San Jose, California) *(LEET'10)*. USENIX Association, USA, 6.

[23] David Fifield, Chang Lan, Rod Hynes, Percy Wegmann, and Vern Paxson. 2015. Blocking-resistant communication through domain fronting. *Proc. Priv. Enhancing Technol.* 2015, 2 (2015), 46–64.

[24] Mozilla Foundation. 2023. PUBLIC SUFFIX LIST. https://publicsuffix.org/.

[25] Godaddy. 2023. Domain Names, Websites, and Hosting. https://sg.godaddy.com/.

[26] Google. 2023. Google Scholar. https://scholar.google.com/scholar?start=0&as_sdt=2005&sciodt=0,5&cites=14996983484050759 76&scipsc=.

[27] Google. 2023. Public DNS-Google for Developers. https://developers.google.com/speed/public-dns.

[28] Daniel Gruss, Michael Schwarz, Matthias Wübbeling, Simon Guggi, Timo Malderle, Stefan More, and Moritz Lipp. 2018. Use-After-FreeMail: Generalizing the Use-After-Free Problem and Applying It to Email Services. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (Incheon, Republic of Korea) *(ASIACCS '18)*. Association for Computing Machinery, New York, NY, USA, 297–311. https://doi.org/10.1145/3196494.3196514

[29] Shuai Hao, Haining Wang, Angelos Stavrou, and Evgenia Smirni. 2015. On the DNS Deployment of Modern Web Services. In *2015 IEEE 23rd International Conference on Network Protocols (ICNP)*. IEEE, 100–110.

[30] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. 2021. How Great is the Great Firewall? Measuring China's DNS Censorship. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 3381–3398. https://www.usenix.org/conference/usenixsecurity21/presentation/hoang

[31] Paul E. Hoffman. 2023. DNS Security Extensions (DNSSEC). RFC 9364. https://doi.org/10.17487/RFC9364

[32] Paul E. Hoffman and Jakob Schlyter. 2012. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698. https://doi.org/10.17487/RFC6698

[33] Rebekah Houser, Shuai Hao, Chase Cotton, and Haining Wang. 2022. A Comprehensive, Longitudinal Study of Government DNS Deployment at Global Scale. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 193–204. https://doi.org/10.1109/DSN53405.2022.00030

[34] Naotake Ishikura, Daishi Kondo, Vassilis Vassiliades, Iordan Iordanov, and Hideki Tode. 2021. DNS tunneling detection by cache-property-aware features. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 1203–1217.

[35] Liz Izhikevich, Gautam Akiwate, Briana Berger, Spencer Drakontaidis, Anna Ascheman, Paul Pearce, David Adrian, and Zakir Durumeric. 2022. ZDNS: A Fast DNS Toolkit for Internet Measurement. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 33–43. https://doi.org/10.1145/3517745.3561434

[36] Bahruz Jabiyev, Steven Sprecher, Kaan Onarlioglu, and Engin Kirda. 2021. T-Reqs: HTTP Request Smuggling with Differential Fuzzing. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, Republic of Korea) *(CCS '21)*. Association for Computing Machinery, New York, NY, USA, 1805–1820. https://doi.org/10.1145/3460120.3485384

[37] Aqsa Kashaf, Vyas Sekar, and Yuvraj Agarwal. 2020. Analyzing Third Party Service Dependencies in Modern Web Services: Have We Learned from the Mirai-Dyn Incident?. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. Association for Computing Machinery, New York, NY, USA, 634–647. https://doi.org/10.1145/3419394.3423664

[38] Erin Kenneally and David Dittrich. 2012. The Menlo Report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102* (2012).

[39] Scott Kitterman. 2014. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. RFC 7208. https://doi.org/10.17487/RFC7208

[40] Athanasios Kountouras, Panagiotis Kintis, Athanasios Avgetidis, Thomas Papastergiou, Charles Lever, Michalis Polychronakis, and Manos Antonakakis. 2021. Understanding the Growth and Security Considerations of ECS.. In *NDSS*.

[41] Athanasios Kountouras, Panagiotis Kintis, Chaz Lever, Yizheng Chen, Yacin Nadji, David Dagon, Manos Antonakakis, and Rodney Joffe. 2016. Enabling Network Security Through Active DNS Datasets. In *Research in Attacks, Intrusions, and Defenses*, Fabian Monrose, Marc Dacier, Gregory Blanc, and Joaquin Garcia-Alfaro (Eds.). Springer International Publishing, Cham, 188–208.

[42] Murray Kucherawy, Dave Crocker, and Tony Hansen. 2011. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376. https://doi.org/10.17487/RFC6376

[43] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. 2015. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *Proceedings of the 2015 Internet Measurement Conference* (Tokyo, Japan) *(IMC '15)*. Association for Computing Machinery, New York, NY, USA, 355–368. https://doi.org/10.1145/2815675.2815683

[44] Xiang Li, Baojun Liu, Xiaofeng Zheng, Haixin Duan, Qi Li, and Youjun Huang. 2021. Fast IPv6 Network Periphery Discovery and Security Implications. In *Proceedings of the 2021 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '21)*. https://doi.org/10.1109/dsn48987.2021.00025

[45] Zhenhua Li, Cheng Jin, Tianyin Xu, Christo Wilson, Yao Liu, Linsong Cheng, Yunhao Liu, Yafei Dai, and Zhi-Li Zhang. 2014. Towards Network-Level Efficiency for Cloud Storage Services. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (Vancouver, BC, Canada) *(IMC '14)*. Association for Computing Machinery, New York, NY, USA, 115–128. https://doi.org/10.1145/2663716.2663747

[46] Daiping Liu, Shuai Hao, and Haining Wang. 2016. 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* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1414–1425. https://doi.org/10.1145/2976749.2978387

[47] Daiping Liu, Zhou Li, Kun Du, Haining Wang, Baojun Liu, and Haixin Duan. 2017. Don't let one rotten apple spoil the whole barrel: Towards automated detection of shadowed domains. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 537–552.

[48] Célestin Matte, Nataliia Bielova, and Cristiana Santos. 2020. Do Cookie Banners Respect my Choice? : Measuring Legal Compliance of Banners from IAB Europe's

Transparency and Consent Framework. In *2020 IEEE Symposium on Security and Privacy (SP)*. 791–809. https://doi.org/10.1109/SP40000.2020.00076
[49] MaxMind. 2023. IP Geolocation and Online Fraud Prevention|MaxMind. https://www.maxmind.com/en/home.
[50] Giovane C. M. Moura, Sebastian Castro, Wes Hardaker, Maarten Wullink, and Cristian Hesselman. 2020. Clouding up the Internet: How Centralized is DNS Traffic Becoming?. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. Association for Computing Machinery, New York, NY, USA, 42–49. https://doi.org/10.1145/3419394.3423625
[51] Namecheap. 2023. Namecheap: Buy a domain name - Register cheap domain. https://www.namecheap.com/.
[52] Palo Alto Networks. 2023. Next-Generation Firewalls - Palo Alto Networks. https://www.paloaltonetworks.com/network-security/next-generation-firewall.
[53] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. 2020. ICLab: A Global, Longitudinal Internet Censorship Measurement Platform. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 135–151.
[54] Adam Oest, Yeganeh Safaei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Kevin Tyers. 2019. Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1344–1361.
[55] OISF. 2023. Suricata-Home. https://suricata.io/.
[56] Craig Partridge and Mark Allman. 2016. Ethical Considerations in Network Measurement Papers. *Commun. ACM* 59, 10 (sep 2016), 58–64. https://doi.org/10.1145/2896816
[57] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. 2017. Global Measurement of DNS Manipulation. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 307–323. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/pearce
[58] Pieter Arntz. 2022. The pitfalls of blocking IP addresses. https://www.malwarebytes.com/blog/news/2022/12/the-pitfalls-of-blocking-ip-addresses.
[59] QAX. 2023. ALPHA Threat Intelligence Platform. https://ti.qianxin.com/.
[60] Radware. 2023. Dark.IoT Botnet. https://www.radware.com/security/threat-advisories-and-attack-reports/dark-iot-botnet/.
[61] Rob Shapland. 2022. How to defend against malicious IP addresses in the cloud. https://www.techtarget.com/searchsecurity/tip/How-to-defend-against-malicious-IP-addresses-in-the-cloud.
[62] 360 Security. 2023. 360 Threat Intelligence Platform. https://ti.360.net/.
[63] 360 Security. 2023. Ghost in action: the Specter botnet. https://blog.netlab.360.com/ghost-in-action-the-specter-botnet/.
[64] Ram Sundara Raman, Prerana Shenoy, Katharina Kohls, and Roya Ensafi. 2020. Censored Planet: An Internet-Wide, Longitudinal Censorship Observatory. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) *(CCS '20)*. Association for Computing Machinery, New York, NY, USA, 49–66. https://doi.org/10.1145/3372297.3417883
[65] TechTerms. 2023. Trojan Horse Definition. https://techterms.com/definition/trojanhorse.
[66] Tencent. 2023. Tencent Cloud. https://intl.cloud.tencent.com/?lang=en.
[67] Junyu Zhou Tianze Ding. 2021. Domain Borrowing: Catch My C2 Traffic if You Can. https://www.blackhat.com/asia-21/briefings/schedule/index.html#domain-borrowing-catch-my-c-traffic-if-you-can-22314
[68] Elisa Tsai, Deepak Kumar, Ram Sundara Raman, Gavin Li, Yael Eiger, and Roya Ensafi. 2023. CERTainty: Detecting DNS Manipulation at Scale using TLS Certificates. *arXiv preprint arXiv:2305.08189* (2023).
[69] Olivier Van Der Toorn, Roland van Rijswijk-Deij, Tobias Fiebig, Martina Lindorfer, and Anna Sperotto. 2020. TXTing 101: finding security issues in the long tail of DNS TXT records. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 544–549.
[70] VirusTotal. 2023. VirusTotal - How it work. https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works.
[71] WordPress. 2023. WordPress.com: Build a Site, Sell Your Stuff, Start a Blog & More. https://wordpress.com/.
[72] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, RC Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, et al. 2021. Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1109–1124.

A ETHICS.

The major ethical concern is to avoid disturbing real-world DNS servers during the measurement. To this end, we addressed each ethical aspect carefully with the guide of ethical principles [38, 56].

In the automated measurement stated in Section 4.1, we strictly limited our query rate and adopted a random order of queries to avoid burdening any DNS server. In particular, on average, we only queried a server once every 130 seconds while collecting responses from different DNS servers (authoritative servers and open resolvers). While collecting correct records, we only included stable open resolvers. A series of studies [43, 57] have shown that unstable DNS resolvers tend to be end-user devices with limited hardware capacities (e.g., home routers). Such devices are exposed to the Internet unintentionally due to misconfiguration, and we should not include them in our large-scale measurement. To obtain stable open resolvers for measurement, we selected the resolvers that were stably operating for two years. This can be archived by comparing the existing open resolver lists (i.e., find the open resolvers appearing in both the latest list and the list collected two years ago) rather than conducting multiple dedicated scans across the entire Internet space.

We also confirm that our measurement targeting top sites (e.g., Tranco sites) does not raise ethical concerns, despite the possibility that the list may include censored or illegal domains. First, previous studies have discussed the ethical considerations of DNS measurements involving top sites. For instance, Pearce et al. [57] utilized open resolvers to measure Tranco sites and demonstrated that their research adhered to ethical principles, which included respect for persons, beneficence, justice, respect for the law, and public interest. Another study [53], which provided the option to opt out of its DNS measurements for censored domains, found no complaints regarding legal concerns. Also, at least 494 studies (e.g., [11, 30, 36, 48, 53]) have involved the Tranco list in their experiments [26], indicating the widespread utilization of top sites in research studies. Second, a censorship system may block queries for censored or illegal domains directly. However, we believe such mechanisms avoid punishing the open resolver, as numerous innocent users may share the same open resolver. Third, we selected stable open resolvers as our targets to mitigate negative impacts. Resolving censored or illegal domains fall within the business scope of a stable open resolver, as normal internet users may attempt to visit such domains.

In the manual investigation described in Appenix C, we first hosted our domains to ensure that the remaining tests did not affect the normal resolution process before our testing. To avoid additional load on a targeted provider, we strictly limited the domain lists (30 domains) we attempted to host on a provider. We also ensured the URs we configured were harmless (`A` records pointed to *127.0.0.1*, and `TXT` records were with our intention and contact). After investigation, we removed all URs that we set.

Also, we have responsibly disclosed our findings and recommendations to most of the affected DNS hosting services mentioned in this paper. Several renowned providers have taken action to fix the security issue of UR.

B CONDITIONS OF EXCLUDING CORRECT RECORDS

URHunter labels r_i as a correct record if it meets any of the five conditions below.

$$\begin{cases} IP(r_i) \subseteq IP(database(d(r_i))) \\ AS(r_i) \subseteq AS(database(d(r_i))) \\ Geo(r_i) \subseteq Geo(database(d(r_i))) \\ Cert(r_i) \subseteq Cert(database(d(r_i))) \\ r_i \in PDNS(d(r_i)) \end{cases} \quad \text{(conditions)}$$

where $IP(r_i)$ returns the IP addresses for record r_i, $AS(r_i)$ returns the AS information for the IP addresses in the record r_i, $Geo(r_i)$ returns the geographic location for the IP addresses in the record r_i, $Cert(r_i)$ returns the HTTPS certificates for the IP addresses in the record r_i, $d(r_i)$ returns the domain name in the record r_i, $database(domain)$ returns the corresponding record for the domain from the database, and $PDNS(domain)$ returns all historical DNS records for the domain in the past six years.

URHunter also utilizes the HTTP responses collected for URs to exclude false-positive results. In particular, URHunter extracts keywords from the HTTP response (e.g., "parking", "parked", and "redirecting"). Then, URHunter performs a statistic analysis for these keywords to determine the actual use of the site and excludes the URs pointing to parked pages or redirection pages.

C HOSTING STRATEGY OF SERVICE PROVIDERS

We aim to investigate the option (e.g., available nameservers and domains) for attackers to exploit URs. To this end, we analyzed a set of renowned DNS hosting providers according to different aspects of test conditions. We selected the popular hosting providers that enjoy a high market share and have been widely studied in previous studies [4, 33, 45], but excluded the ones that we cannot apply for a free testing account. Finally, seven mainstream hosting providers were chosen in this study, including Cloudflare [17], Amazon (AWS Route 53, not GovCloud) [5], Godaddy [25], Tencent Cloud (DNSPod) [66], Alibaba Cloud [3], Baidu Cloud [7], and ClouDNS [18].

We investigated the hosting strategies that affect the attack of URs and proposed four test conditions, as detailed below: *(1) Domain ownership verification.* Once domain ownership is required to verify, attackers cannot set up URs. *(2) Nameserver allocation policy.* Following allocation policy, a hosting service provides a client with several nameservers to host a domain. For example, a service provider may assign random nameservers selected from a nameserver pool. An attacker can also abuse such nameservers. *(3) Supported domain.* The supported domains for hosting limit the choices of an attacker performing covert communication. To evaluate the threat from attackers, we checked whether various types of domains were supported, including unregistered domains, subdomains, second-level domains (SLDs), and *effective TLDs* (eTLDs) [47]. The eTLD represents the TLD operated by registries and includes public suffixes [24] such as *gov.cn*. *(4) Duplicate hosted domain.* Some providers allow hosting multiple zones for the same domain. As a result, attackers can set up URs for a domain, even while the domain owner is utilizing the same hosting provider.

Then, we introduce the test process for each aspect. First, we signed up using two separate accounts at each provider and examined each aspect mentioned above. Then, we checked if the providers set policies preventing attackers from claiming some types of domains. To verify this, we selected five domains from the top 100 Tranco sites, 20 eTLDs, and five unregistered domains to run our experiments. Moreover, we attempted to claim the domains and configured an `A` record pointing to the localhost and a `TXT` record indicating our identity, intentions, and contact information. We removed the URs immediately following the conclusion of the test.

The hosting strategies of popular DNS hosting service providers are summarized in Table 2. We discuss each aspect in detail.

Domain ownership verification. We observed that none of the selected vendors provided verification of domain ownership, and all of them allowed URs. Interestingly, Cloudflare, Tencent, Alibaba, and Baidu designed verification and notification that reminds customers to finish domain delegation. However, even if a user fails to verify the domain ownership, the nameservers assigned to the domain will still handle DNS requests for the hosted domain, which facilitates URs.

Nameserver allocation policy. In total, we found three types of allocation policies among the selected vendors. Some service providers (e.g., Godaddy and Alibaba) assigned *global-fixed* nameservers (i.e., all users shared the same nameservers). Except for nameservers (i.e., *ns[1-2].alidns.com*) that Alibaba assigned to all users, we found that the other nameservers belonging to Alibaba can also serve the hosted domain, despite no announcement to users (e.g., *dns[1-32].hichina.com*). For each hosted domain, Amazon assigned four *random* nameservers selected from a pool with 2,006 nameservers.

Cloudflare and Tencent assigned *account-fixed* nameservers. Typically, they assigned a different set of nameservers for each account, but the same nameservers when a user hosts multiple domains. While multiple users try hosting the same domain, the nameserver allocation policy can be more complex. Using Cloudflare as an example, it ensured the assigned nameservers to the same domain were different across multiple users, and it allocated a new set of nameservers if necessary. Paid users (or attackers) can even sync their URs to all of Cloudflare's nameservers.

Supported domain. We confirmed that various types of domains were supported for hosting URs. All tested providers supported hosting SLD and eTLDs with reserved lists, which limited hosting of extremely popular domains (e.g., *google.com*). However, we found that attackers can still utilize numerous renowned domains, even owned by government entities and educational institutions. For example, Godaddy allowed *google-analytics.com*, *windowsupdate.com*, *gov.kp*, and *edu.kp*, while ClouDNS supported *github.com*, *google.de*, and even *gov.cn*. Both *gov.gd* and *edu.fm* were allowed by Amazon and Alibaba, while Cloudflare supported *info.na* and *cci.fr*. Some providers, including Godaddy and Cloudflare (requiring an extra payment), supported hosting subdomains of SLDs. In addition, Amazon and ClouDNS even supported unregistered domains, indicating liberal hosting policies for attackers.

Duplicate hosted domain. We observed that Cloudflare, Amazon, and Tencent Cloud allowed multiple users to host duplicate domains. Amazon even allowed a single user to create multiple zones for the same domain name. Consequently, while elaborating URs for a popular domain, an attacker can even *share the same hosting provider with the domain owner.*

Table 2: Hosting strategy for common DNS hosting service providers.

Provider	NS allocation policy	Hosting without Verification	Supported domain				Duplicate hosted domain		
			Unregistered	Subdomain	SLD*	eTLD* [47]	Single user	Cross user	No retrieval
Alibaba Cloud	global-fixed	✔	✘	✔	✔	✔	✘	✘	✘
Amazon	random	✔	✔	✔	✔	✔	✔	✔	✔
Baidu Cloud	global-fixed	✔	✘	✘	✔	✔	✘	✘	✘
CloudDNS	global-fixed	✔	✔	✔	✔	✔	✘	✘	✔
Cloudflare	account-fixed	✔	✘	✔	✔	✔	✘	✔	✘
Godaddy	global-fixed	✔	✘	✔	✔	✔	✘	✘	✔
Tencent Cloud	account-fixed	✔	✘	✘	✔	✔	✘	✔	✘

*: Some domain names that we tested were prohibited from hosting.

A security issue of the vendors who prohibited duplicate hosted domains was also confirmed. Specifically, legitimate domain owners will fail to host their domains on a provider if an attacker hosts in advance. Through investigation, we found Tencent and Alibaba developed domain retrieval mechanisms, which allowed disabling the attacker's hosting by finishing ownership verification. However, Godaddy and ClouDNS did not support domain retrieval and were sensitive to this attack. Also found that Amazon was exploitable. Even though Amazon supported duplicate hosted domains across users, it prohibited creating new zones for a domain when its nameservers were exhausted. Through experiments, we found an attacker can achieve this by repeatedly hosting the same domain through API.

Summary. We found all of the selected reputable hosting providers allowed hosting without verification and facilitated URs. Most of them supported hosting well-known SLDs and eTLDs, which provided numerous choices for attackers to perform covert communication. Worse, some of them supported duplicate hosted domains across different users and allowed an attacker to share the infrastructure with the domain owner while setting up URs.