HADES Attack: Understanding and Evaluating Manipulation Risks of Email Blocklists

Ruixuan Li[*], Chaoyi Lu[*], Baojun Liu[*†], Yunyi Zhang[*], Geng Hong[‡], Haixin Duan[*†],
Yanzhong Lin[§], Qingfeng Pan[§], Min Yang[‡] and Jun Shao[¶‖]
[*]Tsinghua University, [†]Zhongguancun Laboratory, [‡]Fudan University, [§]Coremail Technology Co. Ltd,
[¶]Zhejiang Gongshang University [‖]Zhejiang Key Laboratory of Big Data and Future E-Commerce Technology
[*]liruixuan@mail.tsinghua.edu.cn, [*]{luchaoyi, lbj, duanhx}@tsinghua.edu.cn, [*]zhangyyzyy@nudt.edu.cn,
[‡]{ghong, m_yang}@fudan.edu.cn, [§]{tim, pqf}@coremail.cn, [¶‖]chn.junshao@gmail.com

***Abstract*—DNS-Based Blocklist (DNSBL) has been a longstanding, effective mitigation against malicious emails. While works have focused on evaluating the quality of such blocklists, much less is known about their adoption, end-to-end operation, and security problems. Powered by industrial datasets of non-delivery reports within 15 months, this paper first performs large-scale measurements on the adoption of DNSBLs, reporting their prevalent usage by busy email servers. From an empirical study on the end-to-end operation of 29 DNSBL providers, we find they heavily rely on capture servers, concealed infrastructure to lure blind senders of spam, in generating blocklists. However, we find such capture servers can be exploited and report the `HADES` attack, where non-abusive email servers are deliberately injected into popular DNSBLs. Legitimate emails from victims will then be broadly rejected by their peers. Through field tests, we demonstrate the attack is effective at low costs: we successfully inject our experimental email servers into 14 DNSBLs, within a time frame ranging from as fast as three minutes to no longer than 24 hours. Practical assessment also uncovers significant attack potential targeting high-profile victims, e.g., large email service providers and popular websites. Upon responsible disclosure, five DNSBL providers have acknowledged the issue, and we also propose possible mitigation. Findings of this paper highlight the need for revisiting DNSBL security and guidelines in its operation.**

I. Introduction

For over two decades, DNS-Based Blocklist (DNSBL) has been effective in filtering malicious emails [37]. Usually comprising IP addresses and domains of email servers considered abusive, the blocklists are maintained by security organizations (e.g., *Spamhaus* [63]) and can be queried using regular DNS messages. When email servers receive incoming emails, they are thus empowered to mitigate spam by querying the blocklists and rejecting emails originating from abusive IP addresses or domains. Reportedly, many popular email service providers (e.g., *Yahoo* [11]), spam filtering software (e.g., *SpamAssassin* [62]), and domain registries (e.g., *Radix* [51]) are integrated with DNSBL filtering functionalities.

Together with multiple reputation systems, the quality of DNSBLs has been examined by some early studies, reporting that sources of spam can be overlooked [33], [52] or misclassified [60], [61] by the blocklists. Remedies, such as analyzing spammer delivery behavior [53], leveraging auxiliary domain datasets [26], and aggregating multiple blocklists [54] are then proposed to mitigate false negatives or positives. Much less is known about end-to-end operation and security of DNSBLs: *how broadly are they leveraged by email servers; how do abusive servers enter and exit; more importantly, are current DNSBLs prone to manipulation, similar to multiple other DNS reputation systems [27]?* We believe seeking answers to these important questions should help examine comprehensively and bring the email community closer to an enhanced version of this longstanding filtering mechanism.

Studying adoption and end-to-end operation of DNSBLs. To begin, we perform a large-scale measurement of how broadly DNSBLs are leveraged by email servers (§III). Traditionally, given an email server `mx.domain.com`, one may presume its DNSBL adoption by sending emails from bulks of IP addresses or domains to it, and will observe sizable rejections when some of the senders hit blocklists. However, such active approaches face significant ethical risks and may not scale, as bulk emails are unsolicited to recipients. To address such challenges, we turn to passive datasets and leverage a unique observation that email servers report DNSBL usage in *non-delivery reports* (NDRs, or bounce messages) back to senders after rejecting emails [39]. NDRs are privacy-insensitive, as the contents of rejected emails are not enclosed. By collaborating with *Coremail* [22], a large industrial email service provider (ESP) serving over 20,000 enterprises, we inspect 190 million NDRs received by them in 15 months and identify DNSBLs leveraged by their peers (i.e., email servers rejecting emails sent from *Coremail*'s customers). In the end, we find email servers under 307,244 domains report DNSBL usage in NDRs, over 90% of which rely on *Spamhaus*. Among the top 100 and 1,000 recipient domains receiving the highest number of emails, 53% and 45% respectively are leveraging DNSBLs for spam filtering, suggesting the prevalent adoption of this mechanism by busy email servers.

From an empirical analysis of the end-to-end operation of 29 DNSBL providers logged within NDRs, we find they heavily rely on *capture servers* in identifying abusive servers. For example, *spamtraps* are regular but concealed email addresses established by DNSBL providers. Because they are not disclosed, spamtraps will not receive emails, unless contacted by spammers whose major purpose is scanning for email

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.242156
www.ndss-symposium.org

servers blindly and sending messages in bulk. Eventually, such senders are added to DNSBLs with high confidence.

Manipulating DNSBLs: the HADES attack. However, we find capture servers can be exploited and report the HADES attack model, where non-abusive email servers can be injected into DNSBLs by arbitrary attackers (§IV). The key exploit is that despite being concealed, capture servers of several DNSBL providers can be discovered from outside. With capture servers at hand, attackers simply instruct victim servers to send emails, injecting them into DNSBLs due to the design that senders contacting capture servers are highly suspicious, eventually causing subsequent email deliveries from victims to fail.

Depending on the attacker's capabilities, numerous email servers can become victims. For ESPs or enterprises under which attackers possess accounts, their outgoing mail servers become victims when attackers send emails directly to capture servers (Internal attack). For others, if their websites offer subscription or password reset via emails, attackers trigger outgoing emails from these victims by submitting subscription or recovery requests pointing to capture servers (External attack). Additionally, when capture servers do not perform sender identity checks (e.g., via SPF), attackers send them emails from arbitrary spoofed domains (Forgery attack).

With field tests, we demonstrate that HADES is effective at low costs (§V). By summarizing characteristics and building heuristics, we first shortlist capture server candidates from multiple email domain datasets. To test whether attackers can manipulate DNSBLs by sending emails to capture servers, we create dedicated cloud servers and deploy email services under controlled domains, send emails from them at low rates (e.g., one email per minute per machine), and query DNSBLs to check if our IPs/domains appear. Eventually, we successfully injected our servers into blocklists of 14 DNSBL providers; a total of 140,449 domains are confirmed pointing to spamtraps of *Spamhaus* alone. Our servers are injected within a time frame ranging from as fast as *three minutes* (targeting *Spamhaus*) to no longer than 24 hours. In contrast to the short injection time, our servers remain in blocklists until automatically removed seven to 30 days later, depending on the policies of DNSBLs. Finally, we create bogus SPF/DKIM records on our domains and find injection into three DNSBL providers still succeed, e.g., *Spamhaus*, suggesting their capture servers do not check sender identities, rendering Forgery attacks possible against arbitrary domains.

Practical considerations of high-profile victims. Though the HADES model offers the possibility of targeting high-profile victims (e.g., large public ESPs and popular websites), practical considerations remain (§VI). First, email servers that have historically been blocklisted can be injected in DNSBLs by attackers as abuse entries. To evaluate how many popular email servers can become victims, we extract outgoing mail servers under Adobe/Tranco top 1K domains from email logs provided by *Coremail* in 12 months and monitor their existence in DNSBLs for two months. We find up to 77% of them have ever been blocklisted in history, which can become victims. Second, from our field test, attackers should be able to send emails regularly at a given rate, which is infeasible if victim servers exhibit strict rate limits. From a survey of public ESPs and subscriptions of popular websites, we find they are not sufficient to defend against HADES. Third, if the victim is equipped with many outgoing servers, the cost of HADES will rise, as attackers should strive to inject most of them into DNSBLs to create a significant impact on their service. However, we find that about half of the domains in the top lists rely on less than 20 outgoing servers, making them promising victims. Also surprisingly, four domain registries managing 51 Top-Level Domains (TLDs) delete domains when they are injected into DNSBLs, thus escalating the damage of HADES.

Disclosure and end-to-end mitigation. We have responsibly reported the HADES attack to all 14 DNSBL providers prone to manipulation and received acknowledgments from five of them. One provider has expressed willingness to fix this issue, while others are currently reserving due to cost considerations. From our investigation of DNSBL end-to-end operation, we propose mitigation for each stage of the DNSBL workflow, calling for specific community guidelines.

Contributions. Contributions of this paper include:

- *End-to-end measurement of DNSBLs*. Leveraging large-scale passive datasets, we depict the current adoption and operation characteristics of 29 DNSBL providers.

- *A novel attack*. We propose the HADES attack model and demonstrate its efficacy through field tests. We also evaluate its impact on targeting real-world high-profile victims.

- *Mitigation and disclosure*. We reported risks to all DNSBL providers prone to manipulation and received confirmation from five of them. We also propose end-to-end mitigation.

II. Workflow and Usage of DNSBL

The DNSBL mechanism, originating in the 20th century, enables anyone to know the reputation of the host through a single DNS query [37]. Currently, DNSBLs are widely adopted by numerous prominent ESPs to combat spam, such as *Yahoo* [11]. More than 20 open-source mail server software solutions integrate the DNSBL as a built-in anti-spam feature [7]. Furthermore, domain registries also utilize DNSBLs to delete malicious domain names [51], [58].

Figure 1 shows the construction process and usage of the DNSBL. Initially, DNSBL providers capture spammers in the wild through active detection and passive data (①). Then, they determine the maliciousness of the host according to their inclusion rules and blocklist abusive ones (②). The blocklists are then published via DNS zones (③) and regularly updated by DNSBL providers. We categorize DNSBLs that list IP addresses as DNS-IBLs (e.g., zen.spamhaus.org) and those listing domains as DNS-DBLs (e.g., dbl.spamhaus.org). Typically, DNSBL providers remove hosts from blocklists based on specific delisting rules once they are no longer involved in malicious activities (④). Additionally, many DNSBL providers allow users to apply for early removal of blocklisted hosts.

In the following, we introduce the email delivery process and explain how email servers utilize DNSBLs to block spam sources. Initially, the sender delivers the email to the outgoing mail server via the SMTP protocol (❶). Outgoing mail servers can be deployed by individuals or provided by ESPs and email hosting providers. Subsequently, the outgoing mail server tries to establish SMTP communication with the incoming mail server to transmit the email (❷). Before receiving the email

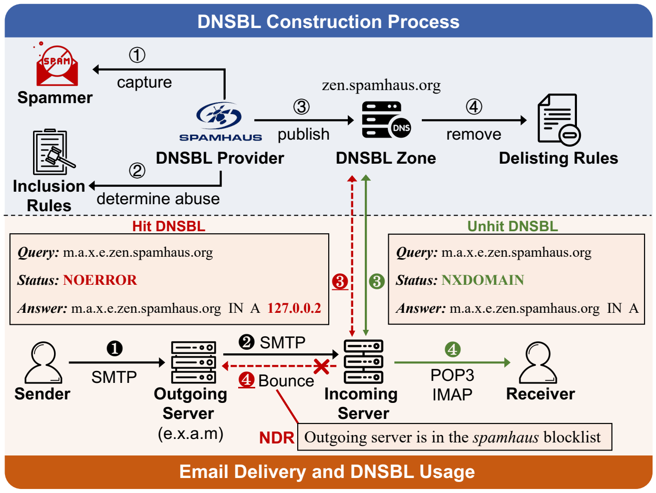
DNSBL Construction Process
Spammer
① capture
SPAMHAUS
DNSBL Provider
② determine abuse
Inclusion Rules
zen.spamhaus.org
③ publish
DNSBL Zone
④ remove
Delisting Rules
Hit DNSBL
Query: m.a.x.e.zen.spamhaus.org
Status: NOERROR
Answer: m.a.x.e.zen.spamhaus.org IN A 127.0.0.2
❸
❸
Unhit DNSBL
Query: m.a.x.e.zen.spamhaus.org
Status: NXDOMAIN
Answer: m.a.x.e.zen.spamhaus.org IN A
Sender
❶ SMTP
Outgoing Server (e.x.a.m)
❷ SMTP
❹ Bounce
Incoming Server
❹ POP3 IMAP
Receiver
NDR
Outgoing server is in the spamhaus blocklist
Email Delivery and DNSBL Usage

Figure 1. The construction process and usage of the DNSBL.

content, the incoming mail server queries DNSBL zones to check if the outgoing IP address (e.g., `e.x.a.m`) is blocklisted. Specifically, the incoming mail server reverses the order of the octets of the outgoing IP address and appends it to the DNS-IBL zone (e.g., `m.a.x.e.zen.spamhaus.org`), then queries its `A` record. If the DNSBL zone returns IP addresses (❸), indicating that the outgoing IP address is blocklisted; or an `NXDOMAIN` code, indicating that is not (❸). The incoming mail server rejects the email from the blocklisted host and returns a non-delivery report (NDR) (❹), or conversely receives the email and forwards it to the receiver (❹). The incoming mail server can also directly append the sender domain (e.g., `domain.com`) to the DNS-DBL zone (e.g., `domain.com.dbl.spamhaus.org`), and query its `A` record to determine the domain reputation. Moreover, DNSBL zones can signify various types of blocklists by returning different IP addresses, commonly within the 127.0.0.0/8 loop network. For example, `zen.spamhaus.org` assigns 127.0.0.2 to spam sources and 127.0.0.4 to malware-infected hosts.

III. Studying Adoption and Operation of DNSBLs

DNSBL has evolved over more than two decades, but the community remains unclear about its deployment and end-to-end operation. Filling these knowledge gaps is crucial for DNSBL improvement. In this section, we introduce our approach to measuring DNSBL deployments based on NDR messages. Furthermore, we conduct a systematic empirical analysis to scrutinize the operation and potential defects of DNSBL providers.

A. *Methodology for Measuring DNSBL Adoption*

Our goal is to identify domains that deploy DNSBLs to block spam sources. Some researchers have proposed methods for actively measuring DNSBL deployment [15], [64]. They use blocklisted and non-blocklisted IP addresses to send many emails to the target domain. A low success rate of emails sent from the blocklisted IP addresses suggests that the target domain deploys DNSBLs. However, this approach is not feasible to measure numerous domains. The main reason is the need to obtain legitimate accounts for each target domain, which is impractical on a large scale, especially considering that many domains restrict email services to internal users.

We investigated previous works [39] and reports [11], [46] to collect information about email servers utilizing DNSBL feeds. The key observation is that some email servers report DNSBL usage in NDRs to help senders resolve email delivery failures. For example, *Outlook* returns "*550 5.7.1 Service unavailable, Client host [x.x.x.x] blocked using Spamhaus*" or "*550 5.7.1 Service unavailable, MailFrom domain is listed in Spamhaus*", indicating that it deploys *Spamhaus* blocklists. Building on these insights, we measure DNSBL deployment by extensively collecting active DNSBL providers from public websites and matching their names in the passive NDR dataset.

At first, we collect 946 DNSBL zones from eight public websites; see Appendix A for detailed sources. Following this, we select DNSBL zones that conform to the RFC 5782 specification [37]. Specifically, DNS-IBL zones must blocklist "127.0.0.2" and exclude "127.0.0.1"; DNS-DBL zones must blocklist "`TEST`" and exclude "`INVALID`". In total, we find 100 active DNS-IBL zones and 26 active DNS-DBL zones. According to Second-Level Domains (SLDs) of DNSBL zones, we identify 60 DNSBL providers. After that, we cooperate with *Coremail* [22], a large ESP in China, to inspect 190M NDRs in the 15-month email delivery log (June 2022 to September 2023). *Coremail* serves more than 20K enterprise customers, and the delivery log involves 68K customer domains and 3M recipient domains. If the recipient domain returns NDRs containing DNSBL provider names, we consider it to deploy corresponding DNSBLs.

Limitations. The effectiveness of our passive measurement relies on NDRs explicitly reporting DNSBL usage. However, we find that certain domains do not indicate the DNSBL they employ in NDRs, such as "*554 5.7.1 Data End Rejected: Listed in Many RBLs*". As a complement, we conducted an active measurement of DNSBL deployment by popular ESPs, as detailed in Section III-B.

B. *Landscape of DNSBL Adoption*

We find that 307,244 domains utilize DNSBLs to block malicious sources. These domains adopt blocklists published by a total of 29 DNSBL providers. Table I presents the number of domains using various DNSBL providers. We can see that *Spamhaus* is the most popular DNSBL provider, deployed by 288,514 domains (90.06%).

Next, we analyze the popularity of domains that deploy DNSBLs. Through the *Coremail*'s email delivery log, we rank 3M recipient domains according to the number of emails they received. We find that 53% of the top 100, 45% of the top 1K, and 46% of the top 10K domains deploy DNSBLs. All 53 domains in the Top 100 domains utilize the blocklist of *Spamhaus*. In addition, 295,550 (94.82%) domains rely on one DNSBL, while a minority (0.36%) utilize more than five DNSBLs. We also investigate the email providers associated with domains deploying DNSBL by querying their `MX` records and extracting SLDs. Table II illustrates the top 10 email providers and the prevalence of the domains deployed on them. Most of these email providers offer hosting services, of which `outlook.com` accounts for 60.54%.

Table I. STATISTICS ON ADOPTION, EMPIRICAL STUDY RESULTS, AND MANIPULATION RISKS OF 29 DNSBL PROVIDERS.

	DNSBL Provider	# Zone[1]	Type	Removal Mode[2]			# Domains Adopted	HADES	Manipulation Source[4]		
				Auto	Early	Free			Trap	Relay	Sharing
1	spamhaus.org	zen dbl	IP Domain	●	●	●	288,514 (90.06%)	✓	✓	✗	✗
2	spamcop.net	bl	IP	●	●	●	15,825 (4.94%)	✓	✓	✗	✗
3	uceprotect.net	dnsbl-1 dnsbl-2 dnsbl-3	IP IP (block) IP (AS)	◐	◐	○	3,304 (1.03%)	✓	✓	✓	✓
4	junkemailfilter.com	black hostkarma	IP/Domain IP/Domain	●	●	●	3,157 (0.99%)	✓	✓	✓	✗
5	sorbs.net	dnsbl spam.dnsbl	IP IP	●	●	●	2,466 (0.77%)	✗	✗	✗	✗
6	manitu.net	ix.dnsbl	IP	●	●	●	1,624 (0.51%)	✗	✗	✗	✗
7	surriel.com	psbl	IP	●	●	●	1,575 (0.49%)	✓	✓	✗	✗
8	barracudacentral.org	b	IP	●	●	●	798 (0.25%)	✗	✗	✗	✗
9	senderscore.com	bl.score	IP	●	●	●	751 (0.23%)	✓	✓	✗	✗
10	spfbl.net	dnsbl	IP	◐	◐	○	366 (0.11%)	✓	✗	✗	✓
11	s5h.net	all	IP	○	◐	●	359 (0.11%)	✓	✓	✓	✗
12	gbudb.net	truncate	IP	●	○	●	295 (0.09%)	✓	✓	✗	✗
13	abuseat.org	cbl	IP	●	●	●	285 (0.09%)	✓	✓	✗	✗
14	beetjevreemd.nl	dnsbl	IP	–	–	–	184 (0.06%)	✗	✗	✗	✗
15	justspam.org	dnsbl	IP	◐	◐	●	168 (0.05%)	✗	✗	✗	✗
16	spamrats.com	all	IP	●	●	●	167 (0.03%)	✗	✗	✗	✗
17	surbl.org	multi	Domain	●	●	●	84 (0.02%)	✓	✓	✗	✗
18	backscatterer.org	ips	IP	●	◐	○	78 (0.02%)	✗	✗	✗	✗
19	zapbl.net	dnsbl	IP	●	●	●	55 (0.02%)	✗	✗	✗	✗
20	virusfree.cz	bad bip	IP IP	●	●	●	52 (0.02%)	✗	✗	✗	✗
21	pte.hu	singular.ttk	IP	●	●	●	49 (0.01%)	✗	✗	✗	✗
22	mailspike.net	rep bl	IP IP	●	●	●	42 (0.01%)	✓	✓	✓	✗
23	spameatingmonkey.net	bl	IP	●	●	●	41 (0.01%)	✗	✗	✗	✗
24	scrolloutf1.com	reputation-ip.rbl reputation-domain.rbl	IP Domain	–	–	–	30 (0.01%)	✗	✗	✗	✗
25	fmb.la	bl	IP	●	●	●	27 (0.01%)	✓	✓	✓	✗
26	redhawk.org	access	IP	–	●	–	18 (<0.01%)	✗	✗	✗	✗
27	brukalai.it	black.dnsbl	IP/Domain	●	●	●	14 (<0.01%)	✓	✓	✓	✓
28	0spam.org	bl rbl dbl	IP IP IP	●	●	●	9 (<0.01%)	✗	✗	✗	✗
29	blocklist.de	bl	IP	●	●	●	9 (<0.01%)	✗	✗	✗	✗

[1] Append the domain name of the DNSBL provider to get the complete DNSBL zone.
[2] ● means DNSBL providers support the automatic/early/free removal of listed hosts; ○ means not support; ◐ means partial support; – means unknown.
[3] ✓ means the DNSBL provider is vulnerability to HADES; ✗ means not vulnerability.
[4] ✓ means that spamtraps (Trap)/email relay servers (Realy)/data sharing sources (Sharing) can be exploited to perform HADES; ✗ means cannot.

Table II. Top 10 email providers deploying DNSBL.

Email Provider	# Domains	Highest/Median Rank
outlook.com	185,996 (60.54%)	2/442,425
mimecast.com	12,314 (4.01%)	57/453,599
secureserver.net	5,581 (1.82%)	125/1,157,999
emailsrvr.com	3,307 (1.08%)	1,028/845,421
one.com	3,043 (0.99%)	3,169/2,310,414
ppe-hosted.com	2,045 (0.67%)	158/820,194
rzone.de	1,823 (0.59%)	6,164/662,150
google.com	1,393 (0.45%)	754/562,702
mimecast.co.za	1,168 (0.38%)	1,317/432,645
loopia.se	1,127 (0.37%)	4,910/2,336,630

Active measurement for popular ESPs. We compensate for the limitations of passive data analysis through active measurement. Based on previous research [20], [57], we first select 15 popular ESPs for which we can successfully register accounts. Then, we configure an experimental email server and ensure that 15 ESPs can properly receive emails from it. Following this, we inject the IP address of our experimental email server into DNS-IBLs (see Section V for the inject method). If almost all of the subsequent emails are rejected by the ESP, we assume it deploys DNS-IBLs. Similarly, we use blocklisted domains to send emails from IP addresses with normal reputations to measure the deployment of DNS-DBLs. Considering the popularity of *Spamhaus*, our focus was specifically on measuring its adoption by 15 popular ESPs.

Appendix B shows the results of the active measurement. When our outgoing IP address hits the blocklist of *Spamhaus*, more than 99% of emails sent to seven ESPs are rejected. This indicates that these ESPs rely heavily on *Spamhaus* to determine IP reputation. In addition, the spam filtering system of three ESPs integrates the domain blocklist of *Spamhaus*. These ESPs only use *Spamhaus* to evaluate the maliciousness of domain names in `MAIL FROM` headers, rather than those in the `FROM` and `DKIM` headers.

In conclusion, our findings demonstrate the importance of DNSBLs for combating spam in the real world. ESPs deploying DNSBLs account for a sizable market share [40], [71], including *Outlook*, *Hotmail*, *Yahoo*, *iCloud*, etc. In particular, the number of domains deploying DNSBLs as indicated by our study represents only a lower bound.

C. Empirical Study on End-to-End DNSBL Operation

DNSBL providers typically build DNSBLs according to their proprietary rules, and the DNSBL-related RFCs lack guidance for the blocklist construction [37], [38]. Consequently, the operations of various DNSBLs are diverse and often opaque. To further investigate the reliability and security of DNSBLs, it is crucial to understand their inclusion and delisting strategies. To this end, we conducted an empirical study on 29 DNSBL providers by reviewing their websites and public reports, as summarized in Table I.

Blocklist types. Almost all DNSBL providers offer IP address blocklists, while *Uceprotect* [3] extends this by listing entire IP blocks and autonomous systems (ASes). In addition, five DNSBL providers offer domain blocklists. When DNSBL providers combine multiple DNSBL zones into a single large zone, we focus only on the combined zone. For example, `zen.spamhaus.org` includes both `sbl.spamhaus.org` and `xbl.spamhaus.org`. Moreover, some DNSBL zones include not necessarily malicious entries and these zones are excluded from our study. For instance, `pbl.spamhaus.org` includes IP addresses typically assigned to users by their ISPs, which are not intended for sending emails.

DNSBL Construction. *This paper proposes the concept of capture servers, which refers to servers that report abusive hosts in the wild to DNSBL providers.* Capture servers are a crucial resource for DNSBL construction. If they are compromised, attackers can easily manipulate DNSBLs. We mainly find three types of capture servers, as illustrated in Figure 2.

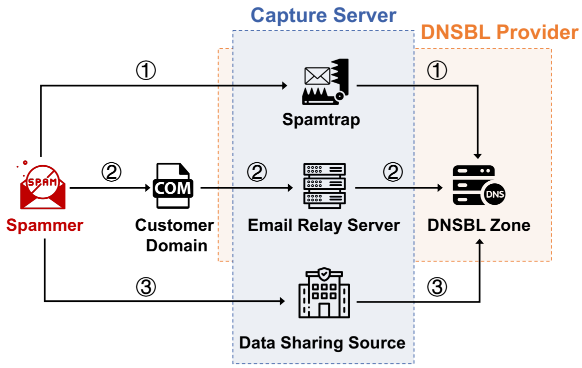

Figure 2. Three types of capture servers.

- **Spamtraps**. The spamtrap is the most important way to capture spammers. Their appearance is a regular email address, but `MX` records of the domain point to the DNSBL provider. Spamtraps can be categorized into two main categories. One is the pristine spamtrap. DNSBL providers use newly registered or unused domains/email addresses to build spamtraps. The other is the recycled spamtrap. DNSBL providers use email addresses that were once valid but are currently expired or frozen as spamtraps. Typically, spamtraps are placed where spammers can collect but are not accessible to normal users. For example, the source code of web pages or leaked email datasets. Therefore, DNSBL providers believe hosts that send emails to spamtraps as high-confidence spammers.

- **Email relay servers**. DNSBL providers publish email servers on their websites and require customer domains to set `MX` records pointing to these servers. This enables DNSBL providers to help customers filter spam and relay legitimate emails. Simultaneously, DNSBL providers detect spammers by the email traffic of customer domains. We find that *Uceprotect* [4] and *Junkemailfilter* [34] offer email relay servers.

- **Data sharing sources**. DNSBL providers compile blocklists through email logs or threat intelligence from other organizations, including collaborators, contributors, cloud service providers, etc. We find that *Uceprotect* [5], *Spfbl* [23], and *Blocklist* [18] publish their collaborators on websites.

Removal modes. Permanently blocklisting IP addresses and domain names often leads to user complaints. Most DNSBL providers automatically delist blocklisted hosts and allow users to request early removal. In addition, DNSBL providers delay automatic removal cycles or disable early removal to penalize repeatedly blocklisted hosts.

Early removal typically requires the ownership proof and abuse explanation of the blocklisted host. However, requesting six DNSBL providers to remove blocklisted hosts is difficult. Specifically, *Gbudb* [28] does not offer active removal services. *Justspam* [35] requires that blocklisted hosts not be included in the other 14 popular DNSBLs. *Uceprotect* [68], *Spfbl* [65], and *Backscatterer* [16] require a fee for the early removal, with costs reaching up to $500. To prevent spammers who abuse open relays from removing blocklisted entries, *S5h* [55] mandates applicants to send removal requests from blocklisted IP addresses. This requirement also prevents customers of email hosting platforms from actively requesting removal, as they do not own the shared server.

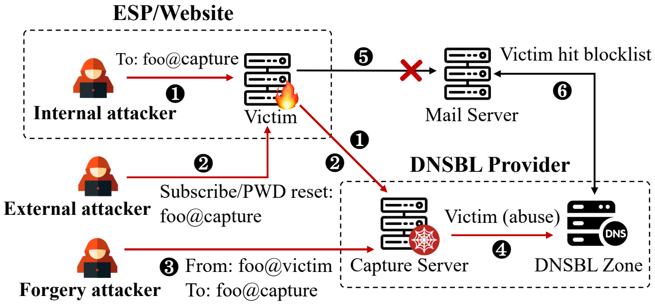

Figure 3. Workflow of the HADES attack.

IV. The HADES Attack: Manipulating DNSBLs

In this paper, we discover adversaries can manipulate DNSBLs to include IP addresses and domains, ultimately leading to a range of serious attacks, including blocking email delivery and deleting domain names. Such attacks are analogous to victims being dragged into the underworld, thus isolated from the outside world. We refer to them as HADES (greek god of the underworld) attacks. In this section, we first describe the threat model of the HADES, and then introduce the detailed attack workflow.

A. Threat Model

HADES attacks aim to destroy the email delivery capability of victims. The key idea is that the adversary instructs the victim to send emails to capture servers. As a result, the DNSBL provider decides the victim is malicious based on false feeding and includes it in blocklists. Ultimately, incoming mail servers that deploy DNSBLs block emails from the victim.

In this paper, we assume that the attacker's capabilities are limited. First, the attacker cannot intercept or monitor mail servers to deliver emails. Second, the attacker does not need to forge IP addresses/domains and craft spam content. Third, the attacker obtains a list of capture servers. According to Section V-A, we can collect capture servers in public datasets with minimal effort. In particular, some spamtrap detection tools launched by commercial companies assist attackers in identifying capture servers [19], [32].

Victims of HADES attacks fall into two main categories. The first includes IP addresses with email delivery capability, including outgoing mail servers belonging to ESPs or websites, which are often shared by numerous entities. For instance, email users of ESPs and customers of web services can all be affected. The second category comprises domain names. If the DNSBL provider does not verify the authenticity of the sender domain, HADES attacks can target arbitrary domain names. Moreover, if registries use DNSBLs to delete abusive domains, victim domains may cease to exist on the Internet.

B. HADES Workflow and Attack Variants

The HADES attack consists of two steps. In the first step, the attacker identifies the capture servers of DNSBL providers. We introduce the detailed methodology in Section V-A. In the second step, the attacker instructs victims to send emails to these capture servers. Attackers can accomplish this through three variants: Internal, External, and Forgery attacks. Figure 3 illustrates the workflow of the HADES attack.

- **Internal attacks.** The adversary is inside the victim provider, holding legitimate accounts from the ESP or website. The adversary sends emails directly to capture servers through the victim's outgoing mail server (❶).

- **External attacks**. The adversary is outside the victim provider, and they can only indirectly induce the victim to send emails to capture servers (❷). Effective inducement methods include abusing the email subscription service and the password reset function of websites. Many websites require authentication for registration, typically involving clicking a link in verification emails. In the case of subscription services, the adversary first generates many email addresses with controlled domains and initiates email subscription applications on the target website, such as news updates and product pushes. Subsequently, the adversary receives verification emails and completes registrations. After that, the adversary configures MX records of controlled domains as capture servers. Eventually, the victim permanently sends automated mass emails to capture servers. Similarly, the adversary can register many accounts and actively request password resets, causing recovery emails of victim websites directed to capture servers.

- **Forgery attacks**. The main target of Forgery attacks is the domain name. The adversary uses controlled servers to send emails to capture servers and set domain names in the Mail From, From, and DKIM headers as the victim (❸).

As the number of false feeds received by capture servers increases, DNSBL providers blocklist victims for their perceived participation in malicious activities (❹). Eventually, the incoming mail server queries the DNSBL zone (❺) and rejects emails from the victims (❻). Registries that integrate DNSBL into their domain reputation systems delete victim domains. In particular, registries often provide interfaces for abuse reports [51], [73], allowing attackers to deliberately report blocklisted domains and expedite their deletion.

V. Performing HADES and Evaluation

An essential capability for adversaries to perform HADES is the identification of capture servers. In this section, we first introduce the characteristics of capture servers and the process of discovering them. After that, we evaluate the effect of exploit capture servers to manipulate DNSBLs.

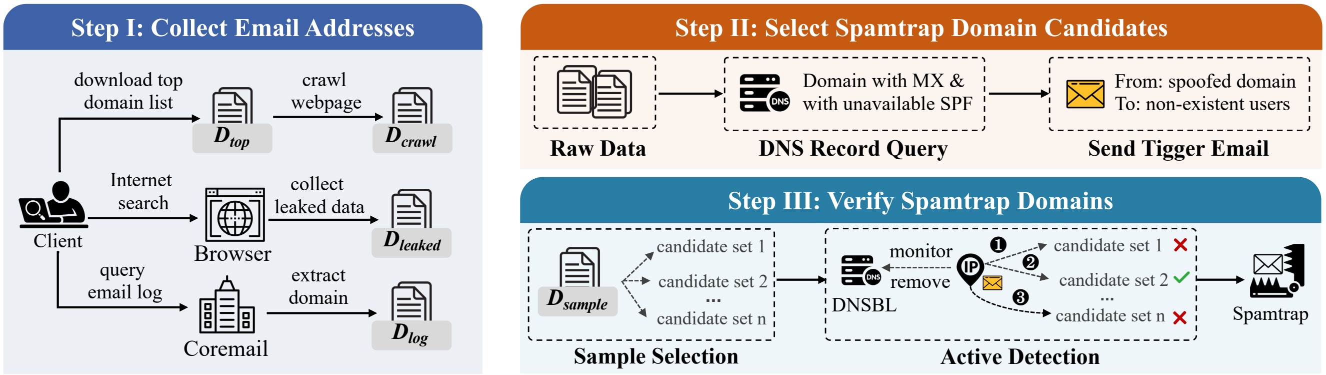

Figure 4. Workflow of discovering spamtraps.

A. Discovering Capture Servers of DNSBL Providers

The key to identifying capture servers is to discover their exposed characteristics in the wild. The identification of email relay servers and data sharing sources is relatively straightforward, primarily relying on information disclosed by DNSBL providers on their websites. Recalling the empirical study in Section III-C, we find eight domains of email relay servers and 104 domains of collaborators.

DNSBL providers never publish, divulge, and sell their list of spamtraps, as this would undermine the value and credibility of blocklists. Although spamtraps and normal email addresses appear extremely similar, their operation exhibits distinctive characteristics. Figure 4 illustrates the workflow for discovering spamtraps. We first extensively collect email domains, then select spamtrap domain candidates through the characteristic filter, and finally verify spamtrap domains.

Step I: Collect email addresses. Because the purpose of spamtraps is to entice spammers into sending emails to them, they must exist in public datasets available to spammers. Therefore, we begin by extensively collecting email addresses from the following four sources to maximize the inclusion of spamtraps. For ethical reasons, we only collect domain names.

- D_{top}: We downloaded and merged three lists of Top 1M popular domains on March 1, 2024, including Tranco [67], Umbrella [69], and Majestic [43]. Finally, we got 2,430,940 unique domains.

- D_{crawl}: Embedding spamtraps in web pages is a common method used by DNSBL providers to attract spammers. On March 5, 2024, we crawled the web pages of domains in D_{top} and extracted email addresses. In the end, we collected 208,847 unique domains.

- D_{leaked}: Historically, vast quantities of email addresses have been leaked on the Internet. The DNSBL provider may retrieve expired domains or email addresses as their spamtraps. We obtained three large email address leakage datasets [8], including Adobe, Anti Public, and Collection #1. Ultimately, we collected 26,845,147 unique domains.

- D_{log}: The email delivery volume of large ESPs is immense, and they may unknowingly send emails to spamtraps. We obtained 3,350,518 unique recipient domains from *Coremail*'s 15-month email delivery log.

Step II: Select spamtrap domain candidates. This step leverages the exposed characteristics to select spamtrap domain candidates. To bootstrap our exploration, we searched the Internet to collect information about spamtraps established by researchers and security agencies [25], [53], [60]. Our key observation is that spamtraps function as honeypot systems that passively capture spammers without delivering emails. Therefore, spamtraps typically accept all emails (*non-rejection*) and do not send emails (*non-sendable*), which can be manifested in the following five characteristics.

1) The domain configures MX records to direct spam to the spamtrap server (*non-rejection*). 2) The domain accepts email delivered to non-existent users (*non-rejection*). 3) The domain accepts emails from spoofed domains with failed SPF and DKIM authentication (*non-rejection*). 4) The domain does not return any bounced emails (*non-sendable*). 5) The domain configure unavailable SPF records (*non-sendable*). In addition to the absence of SPF records, we consider "v=spf1 -all" is also an unavailable SPF record, because it does not allow any host to deliver email on behalf of the domain.

Next, we select spamtrap domain candidates through characteristic matching. We acknowledge that spamtraps may exhibit only some of the above five proposed characteristics. To minimize measurement costs and ethical risks, we retain domains that meet all five characteristics to reduce the number of spamtrap domain candidates. Specifically, we first conduct large-scale DNS scans to select domains with configured MX records and unavailable SPF records. Following this, we send three trigger emails to each selected domain. These trigger emails are sent from our domains configured with invalid SPF and DKIM records, and point to non-existent email addresses.[1] In addition, all emails do not contain malicious content and explain the purpose of our experiment. Finally, we retain domains that do not return any bounced emails and NDRs as spamtrap domain candidates.

Step III: Verify spamtrap domains. Finally, we verify spamtrap domains through active detection. A feasible method is to use different IP addresses to send emails to each spamtrap candidate. If the sender is blocklisted, the corresponding recipient is confirmed as an accurate spamtrap. Although it is possible to obtain numerous IP addresses from cloud platforms and

[1] We use 20 random characters to build non-existent email addresses of the recipient domain, e.g., k4TgsNL2vxE5jAf5gD1b@foo.com.

VPN providers [71], it is ethically unacceptable to send emails from multiple IP addresses and blocklist them. To minimize the impact on the real network, we use a dedicated cloud server and controlled domain to test a sample dataset.

Specifically, we first select a *Coremail*'s outgoing mail server that is most frequently listed in DNSBLs based on NDRs. Following this, we extract 13,172 recipient domains from the email delivery logs of this outgoing mail server. These domains are a subset of D_{log}, and we refer to them as D_{sample}. After that, we select spamtrap domain candidates through Step II and divide them into several candidate sets according to their `MX` records. We then send trigger emails from our outgoing server to domains in each candidate set sequentially, at a rate of one email per minute for five hours. Within one day, if our outgoing server is blocklisted, we consider the domains in the corresponding candidate set as spamtraps. We actively request DNSBL providers to remove our IP address until all candidate sets are tested. The entire experiment lasted eight days.

Limitations. The opacity of the DNSBL construction imposes limitations on our study. First, we mainly analyze the operation of DNSBLs based on empirical studies and external tests. We may miss some loopholes in the DNSBL construction process and cannot accurately know the details of the blocklist rules. Second, we did not accurately identify the spamtraps of all DNSBL providers for ethical reasons. Third, our mail server is specially deployed for experiments, which is different from the reputation and operation mode of the real email server. However, the blocklisting of numerous prominent servers suggests that server age is not a significant factor in the generation of abusive entries by DNSBLs, as detailed in Section VI-A. Therefore, new servers do not greatly affect the accuracy of our spamtrap identification.

B. Finding DNSBLs Prone to Manipulation

In the following, we examine DNSBLs that can be manipulated by three types of capture servers. We find that the security considerations of the DNSBL construction process are generally insufficient, and the blocklists of 14 DNSBL providers can be easily manipulated.

Spamtraps. As a critical defense against spammers, many spamtraps exhibit behavior markedly different from normal email services. Table III shows the results of our selection of spamtrap domain candidates from raw datasets. By applying the five characteristics, we can reduce the raw domain dataset to less than 1%. In particular, 90% of domains can be filtered out only through `MX` and `SPF` records.

Table III. STATISTICS ON THE IDENTIFICATION RESULTS OF SPAMTRAP CANDIDATES.

	D_{top}	D_{crawl}	D_{leaked}	D_{log}	D_{sample}
	Spamtrap Candidates Selection				
Domain	2,430,940	208,847	26,845,147	3,350,518	13,172
# with `MX`	1,064,761 (43.80%)	203,102 (97.25%)	11,385,214 (42.41%)	2,939,404 (87.73%)	13,015 (98.81%)
# unavailable `SPF`	219,380 (9.02%)	36,058 (17.17%)	3,093,727 (11.52%)	500,403 (14.94%)	948 (7.20%)
Domain candidate	17,282 (0.71%)	702 (0.33%)	233,868 (0.87%)	31,102 (0.92%)	21 (0.16%)
Server candidate	3,760	866	9,116	4,734	25
	DNSBL Providers Hit				
DNS-IBL	11	3	12	12	2
DNS-DBL	3	2	3	3	1

To analyze DNSBLs that are vulnerable to manipulation, we send trigger emails from a dedicated server to spamtrap domain candidates at a rate of one per second within five hours. Considering that the number of candidate domains we need to verify is small, if our outgoing IP address or domain name is blocklisted, we consider the DNSBL prone to manipulation. In other words, attackers can conduct `HADES` at an acceptable cost using the list of spamtrap candidates. As shown by Table III, the spamtraps of 13 DNSBL providers expose obvious characteristics, resulting in attackers being able to manipulate 16 DNS-IBL zones and three DNS-DBL zones. The specific 13 DNSBL providers are shown in Table I, and they account for more than 98% of the domains deploying DNSBLs. Furthermore, the number of IP addresses of `MX` records of spamtrap domain candidates, i.e., spamtrap server candidates, is significantly smaller. For example, 31,102 spamtrap domain candidates in D_{log} correspond to only 4,734 spamtrap server candidates. Therefore, the attacker is fully capable of finding the spamtraps of all 13 DNSBL providers. In particular, the attacker can rent many IP addresses at a very low cost, i.e., less than $0.01 for a single IP address [71], to reduce detection time.

Next, we present the results of discovering accurate spamtraps from the D_{sample}. Among 21 spamtrap domain candidates, we find five spamtrap domains of *Spamhaus*. The `MX` records for these five domains are configured with 19 unique IP addresses. Further, we match the `MX` records of domains in our raw dataset with 19 IP addresses to discover more *Spamhaus* spamtraps. Eventually, we identify 140,449 spamtrap domains, which are distributed as follows: 4,069 from D_{top}, 77 from D_{crawl}, 131,724 from D_{leaked}, and 10,350 from D_{log}.

Surprisingly, upon analyzing the `NS` records of *Spamhaus* spamtrap domains, we find that most of them are associated with parking domain providers. This prompted us to further investigate the relationship between 30 popular parking providers and spamtraps. Our result reveals that the `MX` records of numerous domains managed by five parking providers point to spamtraps of *Spamhaus*, including `parkingcrew.net`, `parklogic.com`, `above.com`, `epik.com`, and `fastpark.net`. Moreover, we find that many spamtrap domains are typographical variations of popular domains, such as `gmail.com.com`, `hotmial.com`, and `qqa.com`. This results in normal users often inadvertently directing their emails to spamtraps, resulting in damage to the reputation of ESPs. We observed Coremail customers attempting to send 187,990 emails to *Spamhaus* spamtraps in 15 months. While representing a small portion of *Coremail*'s total email delivery volume, they can significantly impact the server's email delivery success rate. In Appendix C, we provide more details about spamtraps.

Email relay servers. We send trigger emails at a rate of one per second to eight domains of email relay servers within five minutes. As we can see in Table I, our outgoing IP address is blocklisted by six DNSBL providers, and the domain is blocklisted by one. This indicates that these six DNSBL providers are vulnerable to manipulation. By querying the `MX` records of domains in our raw datasets, we find that 1,561 domains are configured with the email relay servers.

Data sharing sources. We send trigger emails at a rate of one per second to 104 collaborator domains of DNSBL providers within 10 minutes. Our IP address is blocklisted by two DNSBL providers, *Uceprotect* and *Spfbl*, indicating that attackers could manipulate them, as shown in Table I.

C. Effect of Successful Attacks

This section explores the effect of manipulating DNSBLs. We monitor the time required to inject IP addresses and domains into DNSBLs using capture servers from various sources. Additionally, we track the duration that hosts remained blocklisted, which is related to the automatic removal policies of DNSBL providers.

Experiments setup. Our experiment involves the capture servers from four sources. The first is spamtrap server candidates, which are used to evaluate the effect of an attacker manipulating DNSBLs only through the characteristic-filtered dataset. For ethical reasons, we select the minimum number of servers needed to cover the spamtraps of 13 DNSBL providers, i.e., 4,734 spamtrap server candidates in D_{log}. We refer to them as C_{sc}. In addition, we randomly select a spamtrap domain of *Spamhaus*, termed C_{sd}. We refer to the eight email relay domains as C_{rd}, and 104 contributor domains of DNSBL providers as C_{cd}.

Next, we inject our IP addresses and domains into DNSBLs by sending emails to capture servers at a limited rate, i.e., as fast as one email per second. The specific send rate and duration are shown in Table IV. After stopping email delivery, we monitor when DNSBLs automatically remove blocklisted entries. To avoid experimental errors introduced by the DNS cache, we directly query authoritative name servers responsible for DNSBL zone updates instead of DNS recursive resolvers. We repeated the above experiment three times, each with a new outgoing IP address and domain. We take the average of all measurements as the final result.

We also investigate whether DNSBL providers strictly verify sender identity when blocklisting domains. Specifically, we set the `Mail From`, `Form`, and `DKIM` headers for trigger emails to three different domains, all configured with bogus `SPF` and `DKIM` records. We then send trigger emails with the incorrect `DKIM` signature from an IP address that violates the `SPF` policy to servers in C_{sc}. Finally, we monitor whether our domains are still blocklisted.

Injection effect. When we send trigger emails to servers in C_{sc}, we can inject our IP addresses into most DNSBLs within three hours and up to one day. In particular, we can inject IP addresses into *Spamhaus* blocklist by sending only three emails to one domain in C_{sd}. Using email relay servers and data sharing sources, we send emails for five to 10 minutes, and our IP addresses can be injected into five DNSBLs within two hours, as fast as three minutes. Moreover, we realize that the time when the IP address appears in DNSBLs is closely related to the frequency of DNSBL zone updates. We infer that *Spamhaus*, *Junkemailfilter*, *Gbudb*, and *Abuseat* update their blocklists at a higher frequency, so attackers can manipulate them more quickly.

Table IV. THE COST AND TIME OF MANIPULATING DNSBLS.

Dataset	C_{sc}	C_{sd}	C_{rd}	C_{cd}	**Remove time**
Rate;Duration	1/s;5h	1/m;3m	1/s;5m	1/s;10m	
	DNS-IBL Listed Time				
Spamhaus	3m	3m	/	/	7d
Spamcop	3h	/	/	/	3d
Uceprotect	1.5h	/	1.5h	1.5h	7d
Junkemailfilter	10m	/	3m	/	7d
Surriel	24h	/	/	/	14d
Senderscore	12h	/	/	/	10d
Spfbl	/	/	/	5m	7d
S5h	10h	/	18h	/	✗
Gbudb	15m	/	/	/	7d
Abuseat	3m	/	/	/	7d
Mailspike	2h	/	1h	/	9d
Fmb	24h	/	8h	/	9d
Brukalai	1h	/	2h	2h	30d
	DNS-DBL Listed Time				
Spamhaus	6h	/	/	/	14d
Junkemailfilter	10h	/	30m	/	7d
Surbl	10h	/	/	/	7d

Compared to IP addresses, it takes longer to inject domains into DNSBLs, usually more than six hours. Seriously, we find that three DNSBL providers still blocklist spoofed domains, which further exacerbates the damage caused by the `HADES` attack. Specifically, *Spamhaus* and *Surbl* blocklist all spoofed domains in the `Mail From`, `From`, and `DKIM` headers, and *Junkmailfilter* blocklist spoofed domains in the `Mail From` field. In addition, we observe that these three DNSBL providers extract SLDs as DNSBL entries when blocklisting domains. This allows attackers to damage the reputation of SLDs by abusing a subdomain.

Removal time. Except for *Spamcop*, the automatic deletion cycle of other DNSBL providers is greater than seven days, with a maximum of 30 days. *Spamcop* lists and delists our IP addresses frequently within three days, with an alternate interval ranging from approximately five minutes to an hour. Furthermore, we actively request *S5h* to remove our IP address.

Overall, sending emails to capture servers is an effective method to manipulate DNSBLs. The attacker can keep victims in DNSBLs for about a week with just a few minutes of effort. In particular, the attacker can repeatedly blocklist victims (e.g., once a week) to increase penalties imposed on them by DNSBL providers, such as prohibiting active or free removal of listed entries.

VI. PRACTICAL CONSIDERATIONS OF HIGH-PROFILE VICTIMS

Theoretically, `HADES` could affect all IP addresses with outgoing email capability and arbitrary domains. However, some practical factors may mitigate the harm for high-profile victims. In this section, we first investigate outgoing mail servers and domains for popular ESPs and websites that can be blocklisted and then evaluate the harm of `HADES` on them. Finally, we reveal how DNSBL usage strategies of domain registries exacerbate the damage of `HADES` attacks.

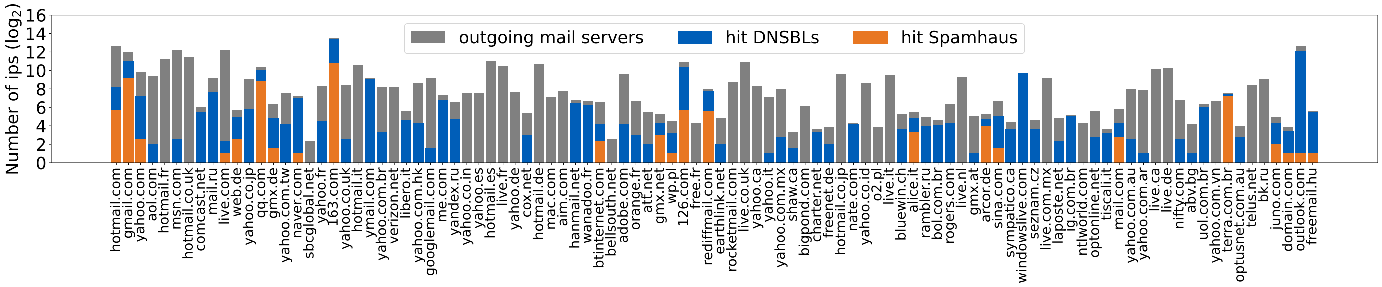

Figure 5. Number of outgoing mail servers of the 100 domains in D_{AT} that have been listed in the least one DNSBL and *Spamhaus* blocklist in two months.

A. Finding Blocklistable Victims

DNSBL providers may maintain allowlists to prevent accidental inclusion of popular email servers [37], so IP addresses and domains in allowlists can not be attacked. However, we find that only four of the 29 DNSBL providers publish their allowlists, i.e., `dnswl.spfbl.net`, `wl.mailspike.net`, `white.dnsbl.brukalai.lt`, and `wl.0spam.org`, excluding the top five most popular providers. Therefore, we turn to the use of blocklists to infer the attack surface of `HADES`. Specifically, email servers that have historically been blocklisted must not be in allowlists, so they can be injected into DNSBLs.

Experimental setup. The key idea of the experiment is to monitor whether popular domains and their outgoing mail servers hit DNSBLs. However, collecting outgoing mail servers is not a simple task. Typically, there are two main ways to obtain email servers. One is through `MX` records, but the results may not contain outgoing mail servers. The other is through `SPF` records, which usually include outgoing mail servers. However, the IP range specified by many `SPF` records is too large [24], resulting in them containing many servers that are not actually working. For example, the `SPF` record of `gmail.com` contains more than 300K IP addresses.

We collect outgoing mail servers through passive email reception logs. Specifically, we first select the Adobe and Tranco Top 1K domains, then extract their outgoing mail servers from *Coremail*'s one-year global email reception log, from May 2023 to May 2024. We find 6M email deliveries from these popular domains, covering 888 in the Adobe Top 1K domains and 476 in Tranco Top 1K domains, which we refer to as D_{AT} and D_{TT}, respectively. We only extract SPF-verified IP addresses to ensure that they belong to ESPs and websites. In total, we collect 50,987 outgoing mail servers. Finally, we monitor DNSBLs every 12 hours for two months to detect whether domains and their outgoing mail servers in D_{AT} and D_{TT} are included.

Blocklistable victims. We find that 39,201 (76.88%) outgoing mail servers of popular ESPs and websites have been included in at least one DNSBL. In particular, *Spamhaus* once blocklisted 8,826 (17.31%) outgoing mail servers in two months. For popular ESPs, outgoing mail servers for 727 (81.87%) domains in D_{AT} once hit DNSBLs, with 106 (11.94%) hitting the *Spamhaus* blocklist, including `hotmail.com`, `gmail.com`, `yahoo.com`, etc. Figure 5 shows the number of outgoing mail servers of the top 100 domains in D_{AT} that hit at least one DNSBL and *Spamhaus* blocklist. Almost all outgoing mail servers of `terra.com.br` have historically been blocklisted by *Spamhaus* at least once. For popular websites, the outgoing mail servers under 379 (79.62%) domains in D_{TT} once hit DNSBLs, such as `google.com`, and `microsoft.com`. Additionally, 139 domains in the Adobe Top 1K list and 56 domains in the Tranco Top 1K list were once included in DNSBLs, and more than 90% of these domains appear in blocklists of *Junkemailfilter*, *Brukalai*, and *Surbl*.

In summary, our results highlight that allowlist coverage of DNSBL providers is not broad enough, resulting in many popular outgoing mail servers and domains that can be victims of the `HADES` attack.

B. Attacking Popular Email Service Providers

The primary way to target ESPs is through `Internal` attacks, typically executed by users of free mailboxes or employees of organizations. Since capture servers look no different from normal email servers, ESPs cannot prevent their users from delivering emails to them. The primary limitation for `Internal` attacks is the email delivery rate imposed by the ESP. Through the survey of popular ESPs [9], [76], we find that free mailboxes usually allow users to send more than 500 emails per day, and some mailboxes only limit the number of emails sent per minute or hour. Moreover, the rate limits for paying subscribers, enterprise customers, and hosting platform users are more relaxed. Recalling Section V-C, the number and rate of emails attackers need to manipulate DNSBLs usually fall below the ESP's limit.

We realize that popular ESPs may deploy many outgoing mail servers, so only a small part of outgoing mail servers hitting DNSBLs have minimal impact on large ESPs. However, we find that many ESPs depend on a limited number of outgoing servers, it is feasible for an attacker to destroy their email delivery capabilities. As shown by Figure 6, about half of the popular domains in D_{AT} and D_{TT} have fewer than 20

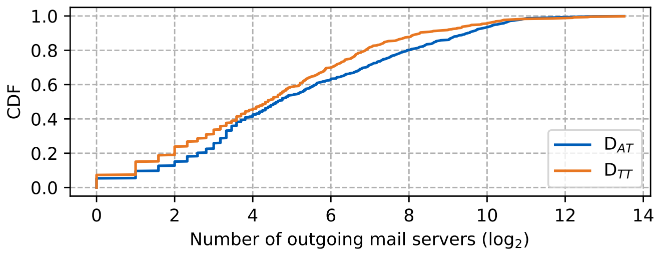

Figure 6. Distribution for the number of outgoing mail servers of domains.

outgoing mail servers. In particular, Figure 5 illustrates that a subset of the outgoing mail servers of domains has been included in DNSBLs, this further reduces the cost of attacks against ESPs. Overall, most outgoing servers of many popular ESPs can be blocklisted by `HADES` with meager attack costs, causing victims significant time and expense for delisting.

C. Attacking Important Websites

Websites typically do not make their email services exposed to the public, so adversaries use `External` attacks against important websites. Common methods to instruct websites to send emails to capture servers include email subscriptions and password resets. We manually investigate the service strategy of the Tranco Top 100 domains. Password reset is a necessary function for almost all websites, and 53 domains that support users receiving news, blogs, and product updates via emails. After successfully subscribing to email services, websites will send emails to capture servers regularly without any further effort from attackers. Many websites frequently send subscription emails, such as *Microsoft* about once a day. Attackers can register many accounts to enhance the damage of `HADES`. Compared with email subscriptions, attackers should actively trigger password resets at a low rate.

Furthermore, we find that many government domains offer email subscription services. We collect 150,306 government SLDs from one year of passive DNS datasets [12] through domain suffix matching [50] (e.g., `.gov.cn`). Then, we use the Google Search API [56] to query whether domains provide the email subscription service and save the top five links in the results.[2] Following this, we use the Chromium web driver [1] to access links and look for input boxes in the web page (e.g., input tags). We exclude links that do not contain the "email" keyword in the properties of the input box. Ultimately, we discover that 528 government domains support email subscriptions, so they are vulnerable to the `HADES` attack. We randomly selected 50 domains for manual verification and found that 90% of them did offer subscription services. These 528 government websites belong to 59 countries, of which the United States accounts for 36.51%.

As the Internet becomes increasingly centralized and shared, many websites outsource their email services to third-party hosting providers [71]. However, even websites that rely on large ESPs are also affected by `HADES`, as mentioned in Section VI-B. In addition, attacking outgoing mail servers of websites through `External` attacks can also disrupt the email delivery service of hosting providers.

D. Escalated Damage by Domain Registries

In Section V-C, we demonstrate that attackers can inject spoofed domains into DNSBLs. For `Forgery` attackers, there are almost no restrictions on their ability to perform the `HADES` attack. Instead of constructing malicious emails and carefully forging domains, they only need a server that can send emails. More seriously, if registries use DNSBLs to delete malicious domains, the damage of `HADES` escalates further, potentially causing the victim domain to disappear from the Internet.

If the status of a newly registered domain in the WHOIS information is `serverHold`, this strongly indicates that it has been deleted by the registry [14], [17]. We collected 219,961 new domains under 401 TLDs from zone files [31] that were added on June 2, 2024 over June 1, 2024. Then, we monitor whether these domains are included in DNSDBLs and whether the status of the blocklisted domains is `serverHold`. The above process was repeated every three hours for one month. We observed that 7,019 (3.19%) domains were blocklisted, of which 6,487 were included in the *Spamhaus* blocklist. Among these blocklisted domains, the status of 398 (5.67%) domains is `serverHold`.

Next, we explore the DNSBL usage policy of domain registries. We exclude TLDs with fewer than 20 blocklisted domains and calculate the percentage of blocklisted domains that are deleted. As shown in Figure 7, the deletion rate of blocklisted domains under 11 TLDs is higher than 50%, with four TLDs reaching 100%. The deletion rate under other TLDs is less than 10%. These 11 TLDs belong to four registries [30], including *DOTSTRATEGY* (`.buzz`), *Shortdot* (`.cyou`, `.sbs`, `.cfd`, `.icu`), *XYZ.COM* (`.lol`, `.xyz`), and *Radix* (`.site`, `.tech`, `.online`, `.store`). *Spamhaus* serves as the primary basis for registries to delete abusive domains. In addition, these four registries operate 51 TLDs, so domains under them are vulnerable to `Forgery` attacks. According to reports, some registries clearly indicate the integration of *Spamhaus* or *Surbl* [51], [58], [74], and many users have experienced domain deletions due to hitting DNSBLs [42], [47]. Overall, attackers can perform `HADES` against domains under at least 51 TLDs, causing them to fail to resolve and associated network services to be terminated.

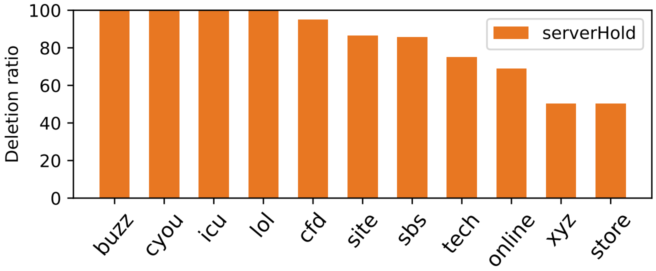

Figure 7. Percentage of blocklisted domains that are deleted by registries.

We also analyze when registries perform deletions of blocklisted domains. Figure 8 shows the time interval between the first time the domain is included in DNSBLs and the

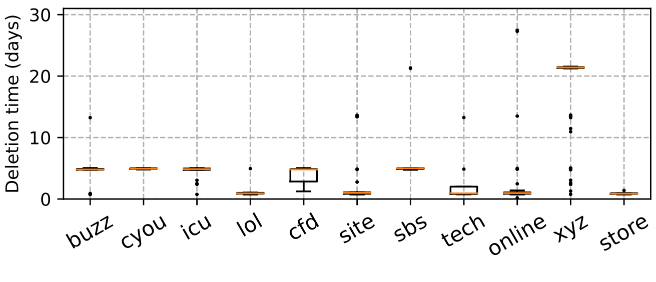

Figure 8. The time interval from when the domain is first listed in DNSBLs to when it is deleted by registries.

[2] For example, we use the "*subscribe email site:.*foo.gov.cn*" syntax to search for `foo.gov.cn`.

deletion. We can see that *DOTSTRATEGY CO* and *Shortdot SA* delete the blocklisted domains in about five days, and *Radix* in about one day. Furthermore, *XYZ.COM* exhibits varying deletion times for blocklisted domains, possibly due to its reliance on multiple sources to assess domain abuse.

VII. Discussion

A. Understanding the Risks of DNSBL Manipulation

As a long-standing and widely deployed mechanism for combating malicious activities, the reliability of DNSBLs is critical to the email ecosystem. The security community has realized the threat posed by the way DNSBL works, i.e., an IP address serving multiple domains can be blocklisted because of the behavior of one domain. For example, a *Cloudflare* shared IP address was flagged as malicious by *Spamhaus*, which ultimately affected a company's email service [21]. However, the security risks of DNSBL manipulation are not well understood within the security community, which potentially amplifies the aforementioned threats of DNSBLs.

The primary reason attackers can execute the `HADES` attack with minimal effort is that capture servers are easy to identify. In addition to the flaws we revealed in Section V, certain capture servers are exposed to attackers in other ways. For example, *Surriel* [2] and *Manitu* [44] offer emails received from the blocklisted host when users request the inclusion details. By constructing numerous emails with unique sender email addresses and delivering them to various domains, an attacker can identify capture servers through the emails offered by these two DNSBL providers. Moreover, aside from the methods mentioned in Section IV, there are numerous other tactics an attacker can use to compel victims to send emails. For example, attackers can post messages in mailing discussion lists to lure victims into replying, use email forwarding services to automatically send emails to capture servers, and redirect bounced emails from victim servers to capture servers. Given that email services are deeply integrated into the Internet, it is challenging for victims to defend against `HADES` once attackers identify capture servers.

Furthermore, the attitudes of participants in the email community towards DNSBLs may create a breeding ground for `HADES` attackers. Specifically, some DNSBL providers may view their role as merely providing threat intelligence and perceive email rejection as a behavior of ESPs. Consequently, they lack sufficient motivation to guarantee DNSBL against manipulation. In the case of ESPs, the majority of them lack the capability and resources to construct blocklists, leaving them with no choice but to rely on DNSBLs. Additionally, ESPs face challenges in identifying whether their customers are sending emails to capture servers. To ensure the availability of email services, they cannot impose too strict restrictions on customer email delivery.

Overall, our study reveals the vulnerability of DNSBLs to malicious manipulation. The injection of numerous legitimate hosts into DNSBLs will severely pollute blocklists and lead to many serious consequences. In particular, the unreliability of blocklists will exacerbate conflicts between users and DNSBLs [10], [45], resulting in damage to the reputation of DNSBL providers. All threat intelligence providers should recognize the importance of not only effectively capturing malicious sources, but also preventing manipulation of blocklists.

B. Mitigation and Disclosure

Combined with our end-to-end investigation of the DNSBL operation, we propose strategies to mitigate the risk of `HADES` attacks for each stage of the DNSBL workflow.

- **Spammer capture**. DNSBL providers should avoid exposing the specific characteristics of their capture servers. For spamtraps, DNSBL providers can enhance their resemblance to normal email addresses, such as configuring valid `SPF` and `DKIM` records for the spamtrap domain. With the email community increasingly emphasizing sender identity authenticity [29], [75], spamtrap domains with invalid `SPF` records will become more conspicuous. Moreover, we recommend that DNSBL providers exercise caution when using domains that closely resemble popular ones as spamtraps. Failing to do so may cause emails from legitimate users to inadvertently reach spamtrap servers, ultimately harming the reputation of legitimate outgoing mail servers. For email relay servers, DNSBL providers can make them invisible on their public websites and only provide services to trusted customers. Additionally, DNSBL providers should not use partner domains directly as their capture servers but rather use internal or private domains from other organizations.

- **Blocklist generation**. DNSBL providers should, within their capabilities, extract email samples for content compliance review. In addition, DNSBL providers should consider email intelligence gathered from multiple capture servers when blocklisting hosts. The above effort prevents the erroneous inclusion of hosts with no malicious behavior. Importantly, DNSBL providers need to carefully check email authenticity through SPF and DKIM mechanisms to avoid blocklisting spoofed domains. Moreover, DNSBL providers can develop more comprehensive allowlists through passive email datasets and active detection to reduce false positives from blocklists.

- **DNSBL zone release**. DNSBL providers should mitigate the additional harm of listing shared IP addresses to legitimate domains. One possible approach is to publish DNSBL zones that support entries for (domain, IP) pairs. For example, the `test.com.1.0.0.127.dnsbl.zone` entry indicates whether `test.com` on 127.0.0.1 is blocklisted. Furthermore, DNSBL zones can help mail servers assess the maliciousness of hosts by indicating details of included entries via TXT records, such as duration and frequency of inclusions.

- **DNSBL delisting**. DNSBLs should allow listed hosts to exit early, which mitigates the impact on misincluded hosts. When processing delisting requests, DNSBL providers must not disclose the header and content of emails they receive, otherwise, attackers can exploit them to detect capture servers.

- **DNSBL usage**. Email providers, registries, and security systems should consider multiple threat intelligence to determine host reputation instead of relying on a single DNSBL.

Following the ethical policy, we have responsibly reported the risk of blocklist manipulation to all 14 affected DNSBL providers. To date, we have received confirmation from five DNSBL providers and engaged in detailed discussions regarding mitigation measures. Specifically, *Spfbl* acknowledged the threat of the HADES attack and promised to fix it in the future. *Spamcop* and *Mailspike* have confirmed the HADES attack but expressed difficulties in implementing defensive measures due to cost considerations. *Gbudb* and *Surbl* have recognized the potential harm of the HADES attack but stated they were not significantly affected. We are currently awaiting responses from the remaining DNSBL providers.

VIII. Ethical Considerations

Our study involves actively sending emails to detect DNSBL construction defects, necessitating a thorough consideration of experimental design to minimize ethical risks. As our institution does not have an Institutional Review Board (IRB), we sought authorization and supervision from our network management department for our research. We conducted a meticulous review of previous works involving similar experiments [14], [41], [49] and adhered to authoritative principles of research ethics [6], [36].

First of all, we follow the principle of "Beneficence" [6] to balance the potential benefits and harms of our study, mainly considering the following three experiments.

- **Measuring DNSBL deployment**. Sending emails to real users from blocklisted sources could provide insights into the DNSBL deployment across global domains. However, this would introduce substantial ethical risks. We measure DNSBL deployment through the passive NDR dataset. NDRs are privacy-insensitive as they do not contain email content, and the sensitive information in the NDRs (e.g., user email addresses) has been anonymized by *Coremail*.

- **Discovering capture servers**. Theoretically, we can identify accurate spamtraps among all spamtrap candidates to better understand DNSBL constructions. However, this process would result in the blocklisting of numerous hosts. Therefore, we limited our verification of spamtraps to an extremely small dataset consisting of 21 domains. We also strictly control the rate of sending emails and monitoring DNSBLs.

- **Practical considerations of attacks**. We know that testing real outgoing mail servers and domains can more comprehensively evaluate the risk of our proposed attack. However, to mitigate ethical risks, we refrained from performing the HADES attacks against hosts outside our control.

Our research aligns with the principle of "Respect for Law and Public Interest" [6]. To mitigate the potential consequences of blocklisting the email delivery source, we exercise caution when selecting the IP addresses and domains for our experiments. Our experimental IP addresses and domains are provided by Alibaba Cloud [13]. Before commencing the experiment, we informed Alibaba of the purpose and method of our study and obtained their permission. More importantly, we actively remove all blocklisted IP addresses and domains after the experiment, and no other users share the experimental host with us. In total, we only use eight cloud servers for testing. Given the considerable number of IP addresses available on Alibaba Cloud, the impact of our experiment on its company reputation is minimal.

We recognize the significance of the principle of "Respect for Person" [6]. We only collect domains to discover capture servers and build non-existent email addresses as recipients. Therefore, emails sent during our experiments do not appear in real user mailboxes. In addition, the email content contains research explanations and contact information to avoid confusion for server administrators and allow them to opt-out.

Finally, we adhere to the principle of "Justice" [6] to ensure that relevant entities benefit from our research. We demonstrated the manipulation risk of DNSBLs and responsibly disclosed the vulnerability to DNSBL providers. We believe our efforts can help them fix loopholes in the DNSBL construction. Furthermore, our study can provide valuable insights to email providers and domain registries regarding DNSBL usage, and promote the security community to improve DNSBLs.

IX. Related Work

Numerous studies investigated the effectiveness of DNSBLs in preventing spam. The results revealed that DNSBLs have both false positives and false negatives. Jung et al. [33] examined seven DNSBLs in 2004 and discovered that 20% of spam sources were not listed. Ramachandran et al. [52], [53] reported that only 5% of IP addresses in the Bobax botnet were included in *Spamhaus*, and 35% of the spam sources found in their spamtraps were not listed in *Spamhaus* or *SpamCop*. Sinha et al. [59], [60] highlighted that targeted and low-rate spam is the root cause of inaccurate blocklisting, and some of *Google*'s servers were blocklisted. Sochor et al. [61] analyzed the email log of a university's SMTP server in 2014. They found that the average monthly spam detection rate of DNSBLs was 74.35%, but false positives were also prevalent. Li et al. [39] observed that some outgoing mail servers of large ESPs were frequently included in *Spamhaus* blocklists, which affects the deliverability of many normal emails.

To optimize the accuracy of email blocklists, many studies have proposed schemes from various perspectives. Ramachandran et al. [53] and Stringhini et al. [66] introduced to capture spammers based on email delivery patterns. They analyzed email delivery logs to identify other hosts that behaved similarly to the malicious hosts. Sinha et al. [60] emphasized the importance of considering the relevance of IP addresses to the local network when constructing blocklists. Markoff et al. [26] proposed the utilization of zone files and WHOIS data to infer malicious domains. Ramanathan et al. [54] suggested aggregating blocklists and extending blocklisted IP addresses to IP prefixes to identify spammers.

In contrast, very little work focuses on the security risks of DNSBLs. Očkay et al. [48] demonstrated the simple concept of risk, i.e., attackers can blocklist victims by submitting fake email reports. However, they did not provide attack verification and evaluation. Furthermore, some scholars explored the manipulation risk of other types of public lists. Pochat et al. [49] analyzed the operation of popular domain lists (e.g., Alexa) and revealed that adversaries can manipulate lists with just one HTTP request. They subsequently introduced Tranco, a more reliable ranked list of popular domain lists. In addition, Xie et al. [72] investigated the causes of the vulnerability of popular domain lists to manipulation and proposed a voting-based approach to construct manipulation-resistant ranked lists. This paper deeply explores the risk sources and hazards of DNSBL manipulation, which can guide the community in improving the reliability of email blocklists.

X. Conclusion

This paper is the first systematic study of the adoption and end-to-end operations of DNSBLs. Leveraging a passive NDR dataset, we find that 30K domains use DNSBLs to block spam sources. Exploiting vulnerabilities in the DNSBL construction process, we propose the HADES attack, a novel threat model that disrupts the email delivery service by manipulating popular DNSBLs. We confirm that the attack is efficient and prevalent, popular ESPs and websites fall victim to attacks. With just three emails, we can inject our email servers into the *Spamhaus* blocklist for a week. In particular, we find that domains under 51 TLDs will be deleted by registries due to blocklisting, compounding the damage of the HADES attack. We responsibly disclose the risk of DNSBL manipulation and provide feasible mitigation strategies. Overall, our study calls the attention of the email community to the security of DNSBL operation and blocklist usage.

Acknowledgment

We thank all anonymous reviewers for their valuable and constructive feedback. This work is supported by the National Key Research and Development Program of China (No. 2023YFB3105600), the National Natural Science Foundation of China (Grant No. 62102218, 62272413), and the "Pioneer" and "Leading Goose" R&D Program of Zhejiang, China (Grant No. 2024C03288). Baojun Liu and Jun Shao are both corresponding authors. This research was partially completed while Ruixuan Li was at Zhejiang Gongshang University.

References

[1] "Chromedriver," https://sites.google.com/chromium.org/driver/home.

[2] "Passive spam block list," https://psbl.org/.

[3] "Uceprotect," https://www.uceprotect.net/.

[4] "Uceprotect faq," https://www.uceprotect.net/en/index.php?m=2&s=0.

[5] "Uceprotect sponsors," https://www.uceprotect.net/en/index.php?m=11&s=0.

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

[7] "Comparison of mail servers," https://en.wikipedia.org/wiki/Comparison_of_mail_servers#Antispam_features, 2024.

[8] "Largest breaches," https://haveibeenpwned.com/, 2024.

[9] "Lists of hosts and their email sending limits," https://kb.mailpoet.com/article/150-lists-of-hosts-and-their-email-sending-limits, 2024.

[10] "The spamhaus project," https://en.wikipedia.org/wiki/The_Spamhaus_Project, 2024.

[11] "Yahoo smtp error codes," https://senders.yahooinc.com/smtp-error-codes/, 2024.

[12] 114dns, https://www.114dns.com/.

[13] Alibaba, https://cn.aliyun.com/, 2024.

[14] E. Alowaisheq, P. Wang, S. A. Alrwais, X. Liao, X. Wang, T. Alowaisheq, X. Mi, S. Tang, and B. Liu, "Cracking the wall of confinement: Understanding and analyzing malicious domain takedowns," in *NDSS*, 2019.

[15] C. Ansari, "Which blacklists does gmail use? new infographic and webinar," https://web.archive.org/web/20140707102925/http://blog.returnpath.com/blog/cherie-ansari/which-blacklists-does-gmail-use, 2014.

[16] Backscatterer, https://www.backscatterer.org/.

[17] T. Barron, N. Miramirkhani, and N. Nikiforakis, "Now you see it, now you don't: A large-scale analysis of early domain deletions," in *RAID*, 2019, pp. 383–397.

[18] Blocklist, "Partners/sponsors from www.blocklist.de," https://www.blocklist.de/en/partners.html.

[19] F. L. Check, "A free health scan for your email addresses," https://freelistcheck.com/, 2024.

[20] J. Chen, V. Paxson, and J. Jiang, "Composition kills: A case study of email sender authentication," in *USENIX Security Symposium*, 2020, pp. 2183–2199.

[21] C. community, "Cloudflare ip on spamhaus.org," https://community.cloudflare.com/t/cloudflare-ip-on-spamhaus-org/269982, 2021.

[22] Coremail, https://www.coremail.cn/, 2024.

[23] S. Customers, https://spfbl.net/en/colaboradores/.

[24] S. Czybik, M. Horlboge, and K. Rieck, "Lazy gatekeepers: A large-scale study on SPF configuration in the wild," in *IMC*. ACM, 2023, pp. 344–355.

[25] DomainTools, "Spamtraps: Creating and seeding," https://www.domaintools.com/resources/blog/spamtraps-creating-and-seeding/, 2024.

[26] M. Felegyhazi, C. Kreibich, and V. Paxson, "On the potential of proactive domain blacklisting," in *LEET*. USENIX Association, 2010.

[27] T. Galloway, K. Karakolios, Z. Ma, R. Perdisci, A. Keromytis, and M. Antonakakis, "Practical attacks against dns reputation systems," in *IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 233–233.

[28] Gbudb, "How ips are removed," http://www.gbudb.com/truncate/how-ips-are-removed.jsp.

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

[30] IANA, "Root zone database," https://www.iana.org/domains/root/db.

[31] ICANN, "zone-requests," https://czds.icann.org/, 2024.

[32] IPQS, "Free spam trap email test," https://www.ipqualityscore.com/spamtrap-email-address-test, 2024.

[33] J. Jung and E. Sit, "An empirical study of spam traffic and the use of dns black lists," in *IMC*. ACM, 2004, pp. 370–375.

[34] Junkemailfilter, "Free mx email server backup service," https://www.junkemailfilter.com/spam/free_mx_backup_service.html.

[35] Justspam, "Removal policy," http://www.justspam.org/policy.

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

[37] J. Levine, "DNS Blacklists and Whitelists," RFC 5782, 2010.

[38] C. Lewis and M. Sergeant, "Overview of best email dns-based list (dnsbl) operational practices," RFC 6471, 2012.

[39] R. Li, S. Xiao, B. Liu, Y. Lin, H. Duan, Q. Pan, J. Chen, J. Zhang, X. Liu, X. Lu, and J. Shao, "Bounce in the wild: A deep dive into email delivery failures from a large email service provider," in *IMC*. ACM, 2024, pp. 659–673.

[40] E. Liu, G. Akiwate, M. Jonker, A. Mirian, S. Savage, and G. Voelker, "Who's got your mail?: characterizing mail service provider usage," in *IMC*. ACM, 2021, pp. 122–136.

[41] M. Liu, Y. Zhang, X. Li, B. L. C. Lu, H. Duan, and X. Zheng, "Understanding the Implementation and Security Implications of Protective DNS Services," in *NDSS*, 2024.

[42] LowEndTalk, "Spamhaus - refusing to delist false positives, pompous/rude attitudes, whats your experience?" https://lowendtalk.com/discussion/193980/spamhaus-refusing-to-delist-false-positives-pompous-rude-attitudes-whats-your-experience, 2024.

[43] Majestic, "Top 1m domains," https://majestic.com/reports/majestic-million, 2024.

[44] Manitu, https://www.manitu.de/.

[45] J. Markoff and N. Perlroth, "Firm is accused of sending spam, and fight jams internet," https://www.nytimes.com/2013/03/27/technology/internet/online-dispute-becomes-internet-snarling-attack.html, 2013.

[46] Microsoft, https://answers.microsoft.com/en-us/outlook_com/forum/all/550-571-service-unavailable-client-host-51xx2xx1x/79a129be-ab20-413b-abf9-659a93c7eec7, 2024.

[47] namePros, "Somebody reported my blog to phishtank and they suspended my domain name," https://www.namepros.com/threads/somebody-reported-my-blog-to-phishtank-and-they-suspended-my-domain-name.1094518/, 2018.

[48] M. Očkay and M. Javurek, "Domain name system black list false reporting attack," in *KIT*, 2013.

[49] V. Pochat, T. Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *NDSS*, 2019.

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

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

[52] A. Ramachandran, D. Dagon, and N. Feamster, "Can dns-based blacklists keep up with bots?" in *CEAS*, 2006.

[53] A. Ramachandran, N. Feamster, and S. S. Vempala, "Filtering spam with behavioral blacklisting," in *CCS*. ACM, 2007, pp. 342–351.

[54] S. Ramanathan, J. Mirkovic, and M. Yu, "BLAG: improving the accuracy of blacklists," in *NDSS*, 2020.

[55] S5h, "how do i delist," http://www.usenix.org.uk/content/rbl.html#how-do-i-delist.

[56] serper, "Google search api," https://serper.dev/.

[57] K. Shen, C. Wang, M. Guo, X. Zheng, C., B. Liu, Y. Zhao, S. Hao, H. Duan, Q. Pan, and M. Yang, "Weak links in authentication chains: A large-scale analysis of email sender spoofing attacks," in *USENIX Security Symposium*, 2021, pp. 3201–3217.

[58] ShortDot, "Shortdot and dns abuse mitigation," https://shortdot.bond/shortdot-and-dns-abuse-mitigation/, 2024.

[59] S. Sinha, M. D. Bailey, and F. Jahanian, "Shades of grey: On the effectiveness of reputation-based "blacklists"," in *MALWARE*. IEEE Computer Society, 2008, pp. 57–64.

[60] ——, "Improving spam blacklisting through dynamic thresholding and speculative aggregation," in *NDSS*. The Internet Society, 2010.

[61] T. Sochor, "Overview of e-mail SPAM elimination and its efficiency," in *RCIS*. IEEE, 2014, pp. 1–11.

[62] A. SpamAssassin, "Enterprise open-source spam filter," https://spamassassin.apache.org/, 2024.

[63] Spamhaus, https://www.spamhaus.org/, 2024.

[64] Spamresource, "Does gmail use spamhaus? or any other blocklists?" https://www.spamresource.com/2023/01/does-gmail-use-spamhaus-or-any-other.html, 2024.

[65] Spfbl, "Query and delist," https://spfbl.net/en/delist/.

[66] G. Stringhini, T. Holz, B. Stone-Gross, C. Kruegel, and G. Vigna, "BOTMAGNIFIER: locating spambots on the internet," in *USENIX Security Symposium*, 2011.

[67] Tranco, "Top 1m domains," https://tranco-list.eu/, 2024.

[68] UCEPROTECT-NETWORK, "Removal of level 3 records," https://www.uceprotect.net/en/index.php?m=7&s=8, 2024.

[69] C. Umbrella, "Top 1m domains," https://s3-us-west-1.amazonaws.com/umbrella-static/index.html, 2024.

[70] T. Vissers, W. Joosen, and N. Nikiforakis, "Parking sensors: Analyzing and detecting parked domains," in *NDSS*, 2015.

[71] C. Wang, Y. Kuranaga, Y. Wang, M. Zhang, L. Zheng, X. li, J. Chen, H. Duan, Y. Lin, and Q. Pan, "BreakSPF: How Shared Infrastructures Magnify SPF Vulnerabilities Across the Internet," in *NDSS*, 2024.

[72] Q. Xie, S. Tang, X. Zheng, Q. Lin, B. Liu, H. Duan, and F. Li, "Building an open, robust, and stable voting-based domain top list," in *USENIX*, 2022, pp. 625–642.

[73] XYZ.COM, "we say no to abuse," https://gen.xyz/account/submitticket.php?step=2&deptid=6.

[74] ——, "The xyz team," https://www.spamhaus.org/authors/the-xyz-team/, 2024.

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

[76] M. Yon, "Email sending limits of various email service providers," https://growthlist.co/email-sending-limits/, 2023.

[77] J. Zirngibl, S. Deusch, P. Sattler, J. Aulbach, G. Carle, and M. Jonker, "Domain parking: Largely present, rarely considered!" in *TMA*, 2022.

Appendix

A. DNSBL Zone Sources

Table V lists the eight sources from which we collect the DNSBL zones.

Table V. Eight source websites for the DNSBL zones.

#	Source
1	https://multirbl.valli.org/list/
2	https://whatismyipaddress.com/blacklist-check
3	https://www.dnsbl.info/dnsbl-list.php
4	https://ipsaya.com/en/ip-blacklist
5	https://rblmon.com/monitored-rbls/
6	https://docs.hetrixtools.com/monitored-blacklists/
7	https://docs.kickbox.com/docs/blocklists-monitored
8	https://rbltracker.com/docs/which-rbls-do-you-currently-monitor

B. The Deployment of Spamhaus Blocklists

Table VI shows the deployment of Spamhaus blocklists by 15 popular ESPs, resulting from our active measurement.

Table VI. The deployment of Spamhaus blocklists by 15 popular ESPs.

ESP	DNS-IBL	DNS-DBL
gmail.com	○[1]	○
outlook.com	●[1]	●
hotmail.com	●	●
yahoo.com	●	○
icloud.com	●	●
qq.com	○	○
tom.com	●	○
yeah.net	○	○
sina.com	●	○
sohu.com	●	○
163.com	○	○
126.com	○	○
139.com	○	○
naver.com	○	○
cock.li	○	○

[1] ● means adopt *Spamhaus*; ○ means not.

C. Analyzing Spamhaus Spamtraps

Recalling Section V-B, we find 140,449 spamtrap domains of *Spamhaus*. Surprisingly, most of the spamtraps are parking domains. We further investigate the relationship between domain parking providers and spamtraps. Specifically, we first selected 30 common providers offering parking services from previous studies [70], [77]. Subsequently, we collected domains hosted on their authoritative name servers from zone files [31], which were obtained on June 1, 2024.

We first analyze the number of parking domains configured with `MX` records. As shown in Table VII, many domains of the five parking providers are configured with `MX` records, of which the corresponding proportion for `parklogic.com` exceeds 96%. For other parking providers, less than 0.1% of their domains have `MX` records. Furthermore, we query the IP addresses of the email servers of all parking domains with `MX` records. The results indicate that 90.03% of them belong to spamtraps of *Spamhaus* that we verified in Section V-B. Parking domains are typically used for advertising and sales [70], not for actual email services. Therefore, emails sent to these domains are likely from spammers. We infer there may be a partnership between domain parking and DNSBL providers.

Table VII. TOP FIVE DOMAIN PARKING PROVIDERS BY NUMBER OF DOMAINS CONFIGURED WITH MX RECORDS.

Parking provider	# Domain	# Domain with MX
parkingcrew.net	684,465	471,090 (68.82%)
parklogic.com	174,308	168,598 (96.72%)
above.com	29,069	26,129 (89.88%)
epik.com	123,462	13,533 (10.96%)
fastpark.net	14,313	11,846 (82.76%)

We also observe that many spamtraps are easy to enter inadvertently by normal users. In particular, the MX records of all domains under 11 SLDs point to *Spamhaus* spamtraps, including `.com.com`, `.edu.edu`, `.cd.cd`, `.tennis.tennis`, `.rocks.rocks`, `.coach.coach`, `.life.life`, `.ninja.ninja`, `.cab.cab`, `.money.money`, and `.dance.dance`. When a user mistakenly types the recipient email address `foo@gmail.com` as `foo@gmail.com.com`, the email is sent to the spamtrap. Moreover, we find that the web pages of 38 domains in the Tranco top 1M list contain *Spamhaus* spamtraps, including embedded locations such as homepages and source codes.

Normal users sending emails to spamtraps will greatly damage the reputation of outgoing mail servers. In the 15-month email delivery business of *Coremail*, we discovered that 64,785 emails were sent to *Spamhaus* spamtraps, and they were delivered 187,990 times. Figure 9 shows the number of emails that *Coremail* sends to spamtraps every day. The most common recipient addresses are typos, such as `gmai.com` (correctly `gmail.com`) and `12.com` (correctly `126.com`). Furthermore, about one-third of undelivered emails are rejected because of outgoing mail servers in DNSBLs. In particular, on February 6, 2023, *Coremail* sent 1,690 emails to *Spamhaus* spamtraps. As a result, on that particular day, 476,108 emails were rejected due to outgoing mail servers hitting *Spamhaus* blocklists, while the average number of rejected emails per day was 18,847.

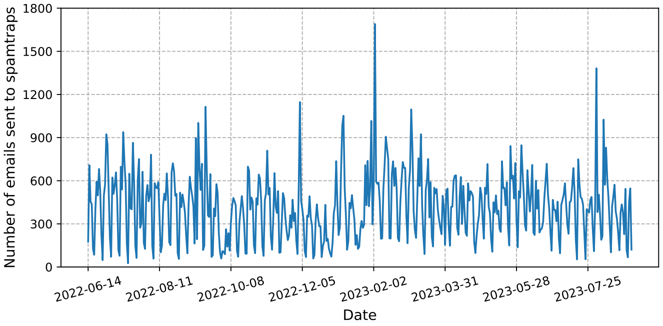

Figure 9. Number of emails delivered by *Coremail* customers to *Spamhaus* spamtrap domains per day.