Bounce in the Wild: A Deep Dive into Email Delivery Failures from a Large Email Service Provider

Ruixuan Li
Tsinghua University
Beijing, China
liruixuan@mail.tsinghua.edu.cn

Shaodong Xiao
Fuzhou University
Fuzhou, China
youngeast351@gmail.com

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

Yanzhong Lin
Coremail Technology Co. Ltd
Guangdong, China
tim@coremail.cn

Haixin Duan
Tsinghua University
Beijing, China
duanhx@tsinghua.edu.cn

Qingfeng Pan
Coremail Technology Co. Ltd
Guangdong, China
pqf@coremail.cn

Jianjun Chen
Tsinghua University
Beijing, China
jianjun@tsinghua.edu.cn

Jia Zhang
Tsinghua University
Beijing, China
zhangjia2017@tsinghua.edu.cn

Ximeng Liu
Fuzhou University
Fuzhou, China
snbnix@gmail.com

Xiuqi Lu
Huazhong University of Science and Technology
Wuhan, China
xiuqilu.hust@gmail.com

Jun Shao
Zhejiang Gongshang University
Hangzhou, China
chn.junshao@gmail.com

Abstract

Abnormal email bounces seriously disrupt user lives and company transactions. Proliferating security protocols and protection strategies have made email delivery increasingly complex. A natural question is *how* and *why* email delivery fails in the wild. Filling this knowledge gap requires a representative global email delivery dataset, which is rarely disclosed by email service providers (ESPs).

In this paper, we first systematically reveal the scale and root causes of email bounces, and evaluate the email squatting risk in the real world. Through a 15-month passive dataset from a large ESP, we present a unique global view of 298M emails delivered to 3M receiver mail servers in 169 countries. We find that 38M (12.93%) emails fail to be delivered in the first attempt, about one-third of which could be successfully delivered after retrying, while the rest are permanently undeliverable. Delving deeper into bounce reasons, we observe that poor server reputation and network communication quality are significant factors leading to temporary email bounces. In particular, spam blocklists affect many normal email deliveries. The misconfiguration of authentication mechanisms and email address typos result in many permanently undeliverable emails. More seriously, many email addresses with significant residual value can be exploited by squatting attackers. Overall, we call for the community to revisit email delivery failures, especially to improve standards for email bounce reporting and resolution.

CCS Concepts

• **Networks** → *Network measurement*; *Application layer protocols*; *Naming and addressing*; • **Security and privacy** → *Security protocols*; *Privacy protections.*

Keywords

Internet measurement; Email security; Email squatting

ACM Reference Format:
Ruixuan Li, Shaodong Xiao, Baojun Liu, Yanzhong Lin, Haixin Duan, Qingfeng Pan, Jianjun Chen, Jia Zhang, Ximeng Liu, Xiuqi Lu, and Jun Shao. 2024. Bounce in the Wild: A Deep Dive into Email Delivery Failures from a Large Email Service Provider. In *Proceedings of the 2024 ACM Internet Measurement Conference (IMC '24), November 4–6, 2024, Madrid, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3646547.3688425

1 Introduction

Prompt and reliable email delivery is the basic expectation of users, especially when dealing with important information, such as resetting passwords and trading bills. Frustratingly, many users have expressed that their emails bounced abnormally, and they fail to comprehend the reasons, let alone know how to resolve the problems [30, 52].

Security protocols and protection strategies proposed to prevent malicious behavior further increase the complexity of email deliveries [37]. For example, DKIM [25], SPF [42], and DMARC [43] are introduced to guarantee email sender authenticity, and spam blocklists [13] are used to intercept malicious email sources. Currently,

IMC '24, November 4–6, 2024, Madrid, Spain

ACM ISBN 979-8-4007-0592-2/24/11
https://doi.org/10.1145/3646547.3688425

email service providers (ESPs) are progressively tightening their requirements for email security [31, 69]. Improper implementation of various mechanisms can all result in email bounces, which poses a significant challenge to email deliverability.

However, the email community currently lacks a comprehensive understanding of *how* and *why* email bounces in the wild. These knowledge gaps significantly hinder the development of the email ecosystem. Therefore, we aim to deeply explore the scale and root causes of email delivery failures, and analyze the associated security risks in the real world.

Challenges. Research on large-scale email delivery behavior poses several challenges. At first, the ethical risks of actively sending emails to numerous ESPs are significant, so it is difficult to obtain a representative dataset. In addition, the reasons for email bounces in the wild are complex and the community lacks systematic analysis methods. Even if a sufficient amount of bounce messages are collected, accurately identifying the types of bounce reasons is not simple.

Our approach. By collaborating with the largest ESP (i.e., Coremail [24]) of China, we obtain the most comprehensive global email delivery dataset ever. To improve domestic and international email communication, Coremail utilizes 34 mail proxy servers distributed across six countries/regions to deliver emails to non-domestic ESPs. In total, we collected 298M emails delivered from mail proxy servers to 3M receiver mail servers distributed across 169 countries/regions within 15 months. Notably, our dataset does not involve cases of receiving emails. We did not collect any email subject or content; see Section 6.1 for specific ethical considerations.

Based upon the above unique dataset, we further automatically classify 190M bounce messages by building the email bounce reason classifier (EBRC). Specifically, we first employ the text clustering algorithm (Drain [33]) to cluster all bounce messages into 10K templates. After that, we analyze the top 200 templates with Coremail's professionals and define 16 types of major bounce reasons. Finally, we use the BERT language model [17] to train the EBRC and then label the types of all bounce messages. Through evaluation, the EBRC achieves 93.85% recall and 91.24% precision.

Major findings. We divide the email delivery status into three cases, i.e., success on the first delivery (`non-bounced`), success after multiple attempts (`soft-bounced`), and constant failure despite repeated tries (`hard-bounced`). Among the 298M emails, 259M (87.07%) are `non-bounced`, 14M (4.82%) are `soft-bounced`, and 24M (8.11%) are `hard-bounced`. After excluding ambiguous bounce messages, we reveal the following five major root causes of the 32M bounced emails.

16M (51.84%) email delivery failures are *active protective bounces* by ESPs, which can be attributed to two root causes. *1) Malicious email delivery.* We observe that attackers send emails to targeted victims by carefully generating 4K email addresses. Most emails are bounced due to non-existent users, while 0.91% of usernames are successfully guessed. Furthermore, some attackers utilize leaked datasets to distribute mass spam, of which 2M (7.71%) emails are rejected by receiver ESPs. *2) Spam blocking policy.* Poor server reputation is the main factor causing `soft-bounced`. 10M (31.10%) emails are delivery failures due to outgoing servers hitting spam blocklists. About half of Coremail's mail proxy servers are blocklisted daily by Spamhaus [13], which is employed by 288K domains. ESPs block numerous normal emails due to the use of Spamhaus blocklists. In addition, the huge differences in spam filter rules among ESPs negatively impact email deliverability.

11M (34.73%) email delivery failures are *passive accidental bounces*, which can be attributed to three root causes. *1) Server manager misconfiguration.* Due to the incorrect deployment of authentication mechanisms (DKIM/SPF/DMARC) and MX records, 701K (2.19%) and 4M (11.37%) emails are `hard-bounced`, respectively. The DKIM/SPF errors last an average of 12 days. Moreover, we find that 11K domains mandate TLS, so outgoing servers without STARTTLS support will experience email bounces. *2) Improper user operation.* The username typos (before @) in receiver email addresses result in 2M (6.85%) `hard-bounced` emails. About half of users require one month to fix full mailbox issues, and inactive account issues are even worse. *3) Poor email infrastructure.* Network quality issues cause 3M (10.20%) `soft-bounced` emails. About one-third of emails delivered to specific regions, mostly in Africa, experienced SMTP session timeouts. The geographic location of outgoing servers also affects email deliverability.

Our dataset contains observations of unregistered email address typos typed by real users. Digging deeper, we find that email address squatting poses persistent and realistic security threats. Specifically, 3K domain names that can be exploited by squatting attackers. 9K users have historically sent 158K emails to these domain names. Some domain names have been purchased by new registrants and launch email services. Furthermore, more than one-third of the usernames we tested are vulnerable to squatting attacks, some of which are associated with popular websites, such as GitHub and Adobe. Attackers can intercept numerous emails by exploiting the residual trust of these vulnerable email addresses.

Overall, our study highlights the significance of thoroughly revisiting email bounces, which can help the email community evaluate the deployment of security strategies and prevent malicious activities. In particular, we propose to standardize bounce message templates, which can improve the understanding and resolution of email delivery failures.

Contributions. Contributions of this paper are as follows.

- Our work first presents the landscape of global email delivery. We evaluate the security mechanism deployment and real user behavior from the email bounce perspective.
- For the first time, we systematically explore the root causes of email delivery failures in the wild, providing new insights for communities to improve the email ecosystem.
- We first widely reveal the squatting risks of email domain names and usernames in the real world, and report security threats to relevant entities.

2 Background

The original design of the email system lacked encryption and authentication, allowing attackers to modify and disguise emails [56]. After more than four decades of evolution, numerous security protocols and protection strategies have been incorporated into the email ecosystem. In this section, we first outline security mechanisms in the email ecosystem. After that, we introduce the process of email delivery, as well as bounce messages and bounce types.

2.1 Email Security Mechanisms

Security protocols. The STARTTLS extension [35] can utilize TLS channels to encrypt SMTP communication, addressing the initial plaintext design flaws of the SMTP protocol. Furthermore, to defend against email spoofing attacks, three security extensions, DomainKeys Identified Mail (DKIM) [25], Sender Policy Framework (SPF) [42], and Domain-based Message Authentication, Reporting, and Conformance (DMARC) [43] were introduced to guarantee email authenticity. Specifically, DKIM relies on digital signatures to ensure the integrity of email content. The receiver can query the public key specified by the sender in the DNS TXT record, and then verify that the signature in the received email is correct. SPF enables senders to publish mail servers through the DNS TXT record that can deliver emails on behalf of their domain. In this way, the receiver can determine whether the source of the email is legitimate. DMARC is a mechanism for detecting email fraud based on DKIM and SPF. The sender can publish DMARC policies through the DNS TXT record to specify how to handle and report emails that do not pass DKIM and SPF verification. Currently, ESPs are becoming more stringent in checking the identity of email senders. For example, Google and Yahoo announced that bulk senders must deploy DMARC after February 2024 [31, 69].

Protection strategies. Since emails can reach a user's mailbox without their consent, spam is a common medium for fraudulent and promotional activities. To mitigate security risks, many ESPs deploy spam filters [64] to check the compliance of email content. Moreover, ESPs also utilize reputation systems to prevent emails from malicious sources. Some ESPs calculate the reputation of email sources based on their internal rules, and some ESPs rely on public Domain Name System Blacklist (DNSBL) [46], such as Spamhaus [13] and Spamcop [12]. DNSBL providers typically build spam blocklists through spamtraps and user reports, then use DNS to indicate whether the IP address or domain name is listed in their blocklists. Furthermore, spammers generally do not understand or comply with email bounce messages to redeliver emails [37]. Therefore, greylisting [5] is another effective spam protection measure that works by rejecting the email sent by a sender on the first attempt and accepting it once the same sender retries.

2.2 Email Delivery and Bounce

In light of the numerous security mechanisms in place, successfully transferring an email is no easy task. Figure 1 illustrates the process of `Alex` delivering an email to `Jun`, which mainly involves **six entities**: the sender, the receiver, the sender domain name server, the receiver domain name server, the sender mail server, and the receiver mail server.

At first, `Alex` needs to configure MX records (<3>) and corresponding IP addresses (<4>) for his/her mail servers on the domain name server (`ns1.a.com`). In addition, to pass email authenticity verification, DKIM records (<5>), SPF records (<6>), and DMARC records (<7>) are also required. After that, the email is transmitted (①) from the `Alex`'s mail user agent (MUA) via SMTP or HTTP protocol to the `Alex`'s mail transmission agent (MTA). For further delivery, the `Alex`'s MTA needs to obtain the `Jun`'s MTA through DNS query (②). Following this, the `Alex`'s MTA transmits the email

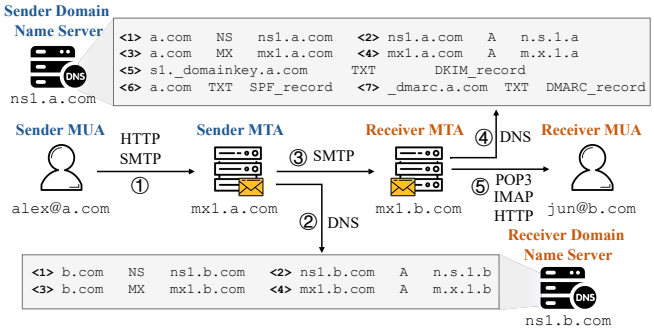

Figure 1: Process of typical email delivery.

to the `Jun`'s MTA through SMTP protocol (③), which can be protected by STARTTLS protocol. Before receiving the email, the `Jun`'s MTA retrieves the DNS records (④) associated with the `Alex`'s domain name, and verifies whether the email satisfies the requirements of DKIM, SPF, and DMARC. Moreover, the `Jun`'s MTA also conducts additional checks, such as email content compliance and email source reputation. After passing all checks, the `Jun`'s MTA transmits the email to the `Jun`'s MUA via HTTP, IMAP, or POP3 protocols (⑤).

Considering the tediousness of the above process, it is not surprising that emails often encounter delivery failures. To represent the email status more intuitively, the ESPs usually provide bounce reasons through non-delivery report (NDR) messages, which primarily consist of the reply code [56] (e.g., 452), the enhanced mail system status code [66] (e.g., 4.2.2), and the specific error text. As an example, when the receiver mailbox is full, the receiver MTA can reply with "*452-4.2.2 The email account that you tried to reach is over quota*". There are eight main types of enhanced mail system status codes [66], which specify more detailed email status compared with email reply codes.

With the proliferation of email security protocols and protection strategies, the reasons for email delivery failures are growing increasingly complex. Even for identical types of bounce messages, their underlying causes may vary. Depending on whether the bounce is permanent or temporary, we can divide the email delivery status into three *bounce degrees*. If the email fails to deliver through any MTA, this email is **`hard-bounced`**. On the contrary, if the email fails to deliver successfully on the first attempt, but is subsequently successfully delivered after retrying, this email is **`soft-bounced`**. Furthermore, the email that is successfully delivered on the first attempt is **`non-bounced`**.

3 Methodology

In this section, we first present the collection process of our email delivery dataset. After that, we introduce our approach to identify various types of reasons for email bounces. Finally, we discuss the limitations of our study.

3.1 Data Collection

This paper aims to explore the root causes of global email bounces. Achieving this task only through active measurement is difficult. The main reasons include: 1) It is unethical to send unsolicited

emails to many real users. Also, creating test email accounts for numerous ESPs is impractical, and many email services are private; 2) Active email delivery is challenging to encompass the diversity and anomalies in real email scenarios. Therefore, we cooperate with our industrial partner Coremail [24], a leading ESP in China, to conduct a comprehensive and longitudinal analysis of email delivery failures in the wild. Coremail provides email delivery services to more than 20K Chinese universities and enterprises. In the following, we present Coremail's email delivery strategy.

Coremail uses its 34 proxy MTAs in six countries/regions to deliver emails, i.e., the **distributed mail proxy strategy**. Figure 2 shows the specific email delivery process. When Coremail receives emails from the sender (❶), it first determines whether the receiver MTA is located outside China. If so, Coremail sends emails to their proxy MTAs through the private encryption protocol (❷). After that, the proxy MTA transmits the email to the receiver MTA via the SMTP protocol (❸). When one proxy MTA fails to deliver the email, Coremail randomly selects another proxy MTA to repeat the above delivery process (❹❺❻❼). To prevent misjudging the maliciousness of emails and thus affecting the customer's email delivery, Coremail sends emails that are determined to be spam once. Conversely, Coremail exerts maximum effort to deliver the normal email to the receiver. Eventually, the receiver MTA transmits the successfully received email to the receiver (❽).

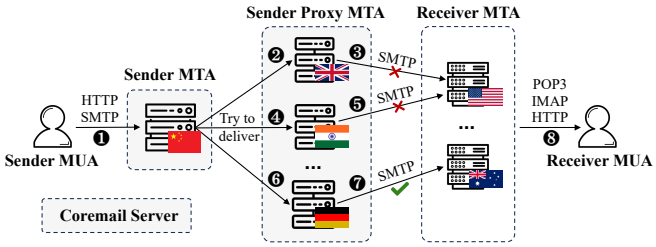

Figure 2: Coremail's distributed mail proxy strategy.

Overall, we can evaluate global email bounces from different countries/regions through the distributed mail proxy strategy. Since this paper focuses on the communication between the sender MTA and the receiver MTA (❸❺❼), we only extract the minimum data required for our research from raw email delivery logs. To avoid exposing personal privacy, we do not collect subject or body information of email. For a more detailed discussion of ethical concerns, please refer to Section 6.1. Figure 3 shows the format of our email delivery dataset. We record proxy MTAs (`from_ip`), receiver MTAs (`to_ip`), NDR messages (`delivery_result`), and latency (`delivery_latency`) associated with each email in all deliveries. In addition, we also record the judgment of email content compliance by Coremail's spam filter (`email_flag`), that is, `Normal` or `Spam`.

Our dataset spans 15 months, from June 14, 2022 to September 6, 2023, and includes a total of 298M emails. We discover 68K sender domains and 3M receiver domains in our dataset. In particular, we build a popularity ranking list based on the number of incoming emails for receiver domains, which we call the `InEmailRank` list. For details of the top 10 receiver domains, see Appendix A. As shown by the orange dots in Figure 4, Coremail's 34 proxy MTAs are

```
{
  "from": "alice@a.com", "to": "bob@b.com",
  "start_time": "2022-06-14 16:30:35",
  "end_time": "2022-06-14 16:45:19",
  "from_ip": ["proxy1_ip", ... , "proxy5_ip"],
  "to_ip": ["dest1_ip", ... , "dest5_ip"],
  "delivery_result": ["550 Mail rejected", ... , "250 OK"],
  "delivery_latency": [54854, ... , 28320],
  "email_flag": "Spam"
}
```

Figure 3: An example of our email delivery dataset.

distributed in six countries/regions: the United States, Hong Kong, Germany, Singapore, the United Kingdom, and India. Considering the receiver MTAs, we observe 574K IP addresses spread across 169 countries/regions and 22K autonomous systems (ASes). The blue heat map in Figure 4 displays the geographic distribution of receiver MTAs. The United States (28.53%), Germany (10.59%), and Canada (5.42%) account for a relatively higher proportion.

Discussion. Because proxy servers are distributed globally, this allows Coremail to deliver emails without experiencing GFW interference, even though Coremail is a Chinese ESP. Consequently, our passive dataset does not introduce limitations because it comes from Chinese ESP.

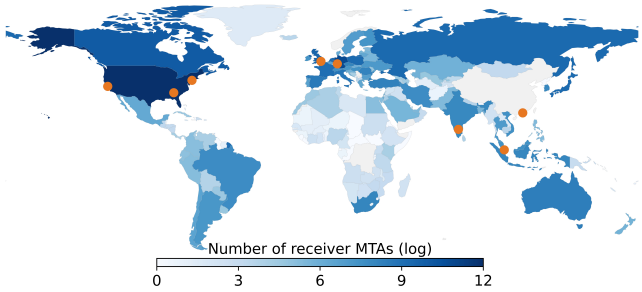

Figure 4: Geographic distribution of receiver MTAs.

3.2 Identifying Bounce Reason Types

An intuitive way to identify types of bounce reasons is based on email status codes. Nonetheless, we find that 28.79% of 190M NDR messages in our dataset do not contain an enhanced mail system status code. In addition, we observe that many NDR messages exhibit inconsistent formats and unclear meanings, as detailed in Appendix B. The above issues mainly arise from the inability of enhanced mail system status codes to cover the various errors of email delivery and the lack of standard NDR message templates in the email community.

To better understand email bounces, we manually determine the main types of bounce reasons, and utilize the language model to build a classifier for automatically identifying NDR message types. The specific process is as follows.

Type determination. At first, we need to determine what types of bounce reasons there are. It is impractical to manually categorize 190M unique NDR messages. Hence, we construct a sufficiently

representative subset of NDR messages to determine the types of bounce reasons. Specifically, we apply the Drain algorithm [33] to cluster 10,089 templates for NDR messages, as done in [70]. For instance, consider the template "*550-5.1.1 (.*) Email address could not be found, or was misspelled (.*)*", wherein the first wildcard signifies email addresses, and the second wildcard signifies vendor-defined codes. Next, we rank all templates based on the number of NDR messages they corresponded to. After that, we manually label the types of the top 200 templates with Coremail's professionals, which cover 68.49% of all NDR messages.

However, we find that some of the top 200 templates obviously could not accurately indicate the reason for email bounces, primarily due to the receiver MTA only returning some ambiguous keywords, such as "*address rejected*" and "*access denied*". In total, we discover ambiguous NDR messages for 6M bounced emails, and the corresponding templates are shown in Appendix B. Since ambiguous NDR messages affect the effectiveness of the classifier, we focus only on 32M of the 38M bounced emails during our classifier training. Below, we introduce six categories and 16 types (T1-T16) of email bounce reasons we defined.

• DNS query failure. The DNS resource record of the sender (T1) or receiver (T2) domain name failed to resolve.
• Violate protocol standard. The sender violates of authentication mechanisms (T3), including DKIM, SPF, DMARC. The sender MTA incorrectly implements STARTTLS (T4).
• Restrict email source. The receiver MTA restricts emails from specific sources, mainly including the sender MTA listed in blocklists (T5), blocked by greylisting (T6), or delivering too fast (T7).
• Refuse email reception. The receiver MTA refuses the receiver email address from receiving emails, primarily due to the nonexistence of the receiver email address (T8), the receiver mailbox is full (T9), excessive (invalid) recipient count of email (T10), the number or rate of incoming emails exceeds the limit (T11), the email is too large (T12), and the email content is considered as spam (T13).
• SMTP connection error. The SMTP session experiences a timeout (T14) or interruption (T15).
• Unknown/other. The reason for the email bounce cannot be determined or does not fall into the above 15 types (T16).

Classifier training. Next, we use the BERT [17] language model to build the email bounce reason classifier (EBRC). Given the training cost, we select 4,000 raw NDR messages for each type as input texts for the language model. The specific steps are as follows. We first select the corresponding templates for each type from the 200 manually labeled templates. Then we use the selected template to match the raw NDR messages. For each type, we try to match a similar number of raw NDR messages for each selected template. For example, we discover eight T12 templates in manually labeled 200 templates, so we use each template to randomly match 500 raw NDR messages. Finally, we fed the language model with various types and corresponding raw NDR messages, enabling it to identify bounce reason types.

Prediction and evaluation. Because of the large volume of raw NDR messages, it is costly to predict them all directly using the EBRC. Therefore, we turn to utilize the EBRC to predict the types of NDR message templates. Specifically, we randomly select 100 raw NDR messages for each unlabeled template separately to constitute a predictive set. Notably, the number corresponding to some templates is fewer than 100. Then, we employ the EBRC to identify the types of raw NDR messages within each prediction set. We take the most frequently occurring type within a prediction set as the type of corresponding template. After that, we label all raw NDR messages through template matching.

Finally, we evaluate the accuracy of the EBRC. We randomly select 100 raw NDR messages for each type of bounce reason and manually analyze the prediction results. Through the confusion matrix, the recall rate of EBRC is 93.85%, and the accuracy rate is 91.24%. Therefore, the EBRC is considered suitable for predicting bounce reason types.

Table 1 shows the distribution of various types of NDR messages for 32M bounced emails. Some bounced emails may be associated with multiple types of reasons. There are 1,367,513 (4.26%) emails corresponding to T16 (unknown/other), which are not shown in Table 1. Through further analysis, most NDR messages associated with T16 are ambiguous. The common NDR templates include "*550 (.*) This message is not RFC 5322 compliant*", "*421 (.*) Intrusion prevention active for (.*)*", etc.

Table 1: Statistics on types of NDR messages for 32M bounced emails. The top five types are highlighted.

T1	T2	T3
575,381 (1.79%)	6,432,448 (20.06%)	851,529 (2.65%)
T4	**T5**	**T6**
597,229 (1.86%)	9,975,329 (31.10%)	843,425 (2.63%)
T7	**T8**	**T9**
816,131 (2.54%)	2,391,760 (7.46%)	660,254 (2.06%)
T10	**T11**	**T12**
248,654 (0.78%)	599,164 (1.87%)	170,987 (0.53%)
T13	**T14**	**T15**
2,986,543 (9.31%)	4,822,598 (15.04%)	2,088,693 (6.51%)

3.3 Limitations

Our study is mainly based on the analysis of a large passive dataset. It is important to highlight the potential limitations of our dataset. At first, our dataset can only cover emails delivered from countries/regions where 34 proxy MTAs are located. Emails received by Coremail are not included in the data we collect. In addition, most senders are staff or students from Chinese companies and educational institutions, so their delivery behavior may not be representative of global users.

Second, the landscape of email bounces from other ESPs may be different from the corpus we collected through the distributed mail proxy strategy. Specifically, the ESP may not repeatedly deliver the bounced email or only use the same MTA to redeliver the email. In addition, our identification of malicious emails depends on Coremail's spam filters, which may differ from the filtering rules used by other ESPs. Although it is difficult to assess how many large ESPs our measurements can represent, our analysis of the most comprehensive email delivery dataset to date can enhance the community's understanding of email bounces.

Third, our determination of the reasons for email bounces is constrained by the authenticity and accuracy of NDR messages

provided by receiver MTAs. Indeed, NDR messages are currently the most prevalent and effective means for the community to understand email bounces. Given that ESPs rarely disclose internal email delivery data, we believe that our study can drive the community to improve email deliverability.

4 Exploring Email Delivery Failures

In this section, we first present an overview of email bounces in the wild. Then, we thoroughly investigate the root causes of email delivery failures from two perspectives, i.e., active protective bounces and passive accidental bounces.

4.1 Overview of Email Bounces

The categories and types of bounces that we defined in Section 3.2 reflect only the literal meaning of NDR messages. We can more accurately infer the root causes of delivery failures based on the bounce degree (see Section 2.2). For example, the `hard-bounced` email due to the MX record resolution error is generally caused by the misconfiguration of the domain name manager, rather than temporary DNS resolution failure or erroneous DNS cache.

Our results show that out of 298M emails, 259M (87.07%) are `non-bounced`, 24M (8.11%) are `hard-bounced`, and 14M (4.82%) are `soft-bounced`. The bar chart in Figure 5 represents the number of emails that experience different bounce degrees per day, aligning with the left y-axis. The `hard-bounced` is the primary reason affecting email deliverability, prompting us to explore the root causes that are most closely associated with email delivery failures. In addition, `soft-bounced` can usually be resolved through additional delivery efforts by the ESP. Our dataset reveals that `soft-bounced` emails experienced an average of three deliveries. Therefore, in cases where the bounce degree of an email cannot be determined, we recommend that ESPs make at least three delivery attempts to improve email deliverability.

The line chart in Figure 5 shows the number of emails delivered on a monthly basis, aligning with the right y-axis. The surge in email deliveries in January 2023 is perhaps due to increased user work and company business ahead of the Chinese New Year (January 22nd). We can also clearly observe a significant decrease in the number of email deliveries on Saturdays and Sundays. This indicates that users depend more on email during the weekdays when study and work tasks are more demanding.

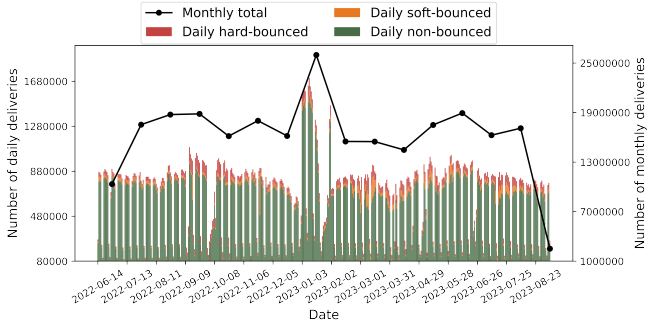

Figure 5: Number of daily/monthly email deliveries.

Furthermore, the email delivery process involves six entities (see Section 2.2). From the perspective of email deliverability, we refer to the entity that caused emails to bounce as the *causative entity*. We consider the attacker as a special causative entity. Through the above analysis, we summarize five root causes for email delivery failures. Table 2 provides the common types, categories, bounce degrees, causative entities, and root causes of 32M bounced emails. Additionally, we show email bounces across different countries, ASes, and receiver domains in Appendix A.

4.2 Active Protective Bounces

Email has long been the primary medium of malicious activities. ESPs typically deploy various protections to detect email content and block malicious sources. Among the 32M bounced emails in our dataset, we find that 16M (51.84%) are active protective bounces by receiver ESPs. These bounces can be attributed to two root causes. The first is malicious email delivery, such as attackers delivering large amounts of spam. The second is the spam blocking policy, primarily implemented by the receiver MTA to prevent spam.

4.2.1 Malicious Email Delivery. We find that 2M (7.74%) email bounces are caused by malicious email deliveries. These bounces are warranted and reflect the ESP's efforts to protect user security. In the following, we introduce two main types of malicious activities.

Username guess. Typically, the initial step for an attacker to commit malicious behaviors is to obtain the victim's email address. From our dataset, we observe that nine sender domains try to deliver emails to many guessed email addresses, which results in 9K (0.03%) emails being `hard-bounced` due to the non-existent receiver. Specifically, these senders focus on specific users or organizations, combining social engineering to create numerous email addresses with mutated usernames (after @) for special strings. This approach is very purposeful and has a high attack success rate. We observe that senders generate 4,273 email usernames that may conform to human habits (e.g., abbreviate, add hyphens) for some victim's names (e.g., a bank's CEO). Unfortunately, 39 (0.91%) email usernames are successfully guessed, causing victims to receive 536 potentially malicious emails, such as spear phishing scams. We reported the relevant risks to the victim's ESPs, and Coremail flagged the attackers as malicious.

Bulk spam. Leaked email datasets are common means for malicious attackers to deliver spam and marketing ads. We investigate the prevalence of the above behaviors in the real world. If more than 80% of the receiver email addresses associated with a sender domain can be queried in the HaveIBeenPwned [6], we consider the corresponding sender to be an attacker who deliberately collects email addresses through the leaked dataset. We found 31 malicious sender domains, and they tried to deliver 3M emails. Among them, 2M (70.12%) emails are `hard-bounced`, and 258K (7.32%) emails are `soft-bounced`. In particular, 23 malicious sender domains are flagged as spammers by Spamhaus [13]. Overall, we recommend that security organizations consider historical delivery behavior when determining malicious senders, especially the source and characteristics of receiver email addresses.

Table 2: Statistics on types, categories, bounce degrees, causative entities, and root causes of common email bounces.

Root Cause					
Type	**Category**	**Bounce Reason**	**Bounce Degree**	**Causative Entity**	**Number**
Malicious Email Behavior					**2,481,639 (7.74%)**
T8	Refuse email reception	Guess victim email addresses	hard	Attacker	9,324 (0.03%)
T8/T13	Refuse email reception	Delivering large amounts of spam	hard	Attacker	2,472,315 (7.71%)
Spam Blocking Policy					**14,143,814 (44.10%)**
T5	Restrict email source	Sender MTA listed in blocklists	hard/soft	Receiver mail server	9,975,329 (31.10%)
T6		Sender MTA blocked by greylisting	hard/soft	Receiver mail server	843,425 (2.63%)
T7		Sender MTA delivers too fast	soft	Receiver mail server	690,137 (2.15%)
T13	Refuse email reception	Email detected as spam	hard	Receiver mail server	2,203,518 (6.87%)
T11		User gets too much email	hard	Receiver mail server	431,405 (1.35%)
Server Manager Misconfiguration					**4,920,494 (15.34%)**
T3	Violate protocol standard	Sender authentication failure	hard	Sender name server	701,347 (2.19%)
T4		Server does not support STARTTLS	soft	Sender mail server	572,324 (1.78%)
T2	DNS query failure	Error MX record for receiver domain	hard	Receiver name server	3,646,823 (11.37%)
Improper User Operation					**2,947,801 (9.19%)**
T2	DNS query failure	Receiver domain name typo	hard	Sender	89,450 (0.28%)
T8	Refuse email reception	Receiver username typo	hard	Sender	2,198,064 (6.85%)
		Receiver email address is inactive	hard	Receiver	12,074 (0.04%)
T9		Receiver mailbox is full	hard	Receiver	648,213 (2.02%)
Poor Email Infrastructure					**3,271,341 (10.20%)**
T14	SMTP connection error	SMTP session timeout	soft	/	3,271,341 (10.20%)

Key Takeaway: We observe that attackers successfully guessed the email addresses of 39 users, with a success rate of 0.91%. In addition, some attackers utilize leaked datasets to send 3M emails, of which 2M (70.12%) emails are blocked by receiver MTAs.

4.2.2 Spam Blocking Policy. Malicious email activities seriously damage the security and privacy of users. To mitigate threats, ESPs deploy various strategies to restrict spam and malicious delivery. Next, we explore the impact of blocklists and spam filters on email delivery, which leads to 14M (44.10%) email bounces.

Blocklists. ESPs can refuse email communication with blocklisted hosts through DNSBLs or their internal reputation systems. We find that 10M (31.10%) emails experience delivery failures due to sender MTAs hitting blocklists. Although 80.71% of emails can be successfully redelivered after changing MTAs, this process requires an average of three attempts, which affects the timeliness of emails. Furthermore, NDR messages can prompt the blocklist hit by the email source.[1] We find that 288K receiver domains adopt Spamhaus [13], including Yahoo [11] and Outlook [8], etc.

Exploring further, we analyze the impact of the Spamhaus blocklist on email bounce. The black line in Figure 6 represents the number of Coremail's proxy MTAs that are blocklisted by Spamhaus, aligning with the right y-axis. On average, half the number of proxy MTAs are blocklisted by Spamhaus every day. In particular, five proxy MTAs are blocklisted by Spamhaus for more than 70% of the days within 15 months. The bar chart in Figure 6 shows the number of spam and normal emails blocked by ESPs using Spamhaus, aligning with the left y-axis. We can see that an average of 16K emails per day fail to be delivered. The number of bounced emails increased significantly after February 2023, mainly due to the addition of 63K domains in February 2023 adopting Spamhaus. Unfortunately, 78.06% of emails bounced due to the Spamhaus blocklist are `Normal` (marked by Coremail). The main reason is that removing the host from the blocklist is not always simple and timely [10, 65], which affects the email delivery of blocklisted servers for some time.

The above results highlight the importance of maintaining a good reputation for email servers, especially for shared MTAs that are responsible for the email delivery of many users. The malicious behavior by some users, such as we show in Section 4.2.1, may damage the reputation of MTAs. Overall, we recommend that ESPs strengthen scrutiny of their users' behavior and periodically monitor the reputation of outgoing servers. In addition, blocklist providers should consider the email delivery history of the host more seriously.

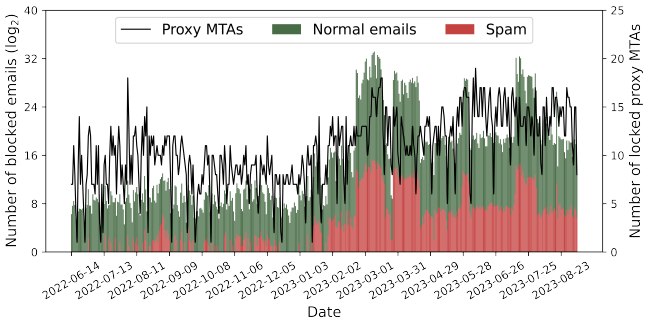

Figure 6: Proxy MTAs and emails blocked by Spamhaus.

Greylisting. Coremail's distributed mail proxy strategy is in direct contradiction to the working principle of the greylisting. Specifically, greylisting [5, 32] rejects the tuple (sender MTA, sender

[1]For example, "*Service unavailable, Client host blocked using Spamhaus*".

address, receiver address) for the first delivery attempt and accepts emails from this tuple after a certain period of time (e.g., 300 seconds). However, Coremail randomly selects a proxy MTA for each email delivery. In our dataset, 783 receiver domains explicitly indicate the adoption of the greylisting. As a result, 843K (2.63%) emails experience delivery failures due to violations of the greylisting policy. Coremail recognized this problem and promised to improve its delivery strategy in the future.

Spam filters. ESPs can deploy spam filters to prevent malicious emails. We find that 2M (6.87%) emails are determined as spam by 11K receiver domains and therefore permanently undeliverable. Furthermore, we discover that rule differences between spam filters can seriously affect email deliverability. Specifically, among emails marked as `Spam` by Coremail, 46.49% are considered not spam by receiver domains. Since Coremail delivers emails marked as `Spam` only once, this may result in potential delivery failure for emails that could have been successful with further delivery attempts. Moreover, for the emails that receiver domains determined as spam, Coremail considers that 39.46% are `Normal`. The consequence is that Coremail redelivers these emails multiple times, causing the reputation of their MTAs to deteriorate and affecting subsequent normal email delivery.

Key Takeaway: The poor reputation of sender MTAs seriously hinders email delivery, resulting in 10M (31.10%) email bounces. In particular, 288K receiver domains rely on Spamhaus to determine host reputation. About half the number of Coremail's proxy MTAs are blocklisted by Spamhaus every day. Unfortunately, 78.06% of emails intercepted by receiver domains through Spamhaus are normal. In addition, the huge differences in spam filter rules between ESPs also seriously affect email deliverability.

4.3 Passive Accidental Bounces

As a complex system involving the collaboration of multiple parties, email experiences many passive accidental bounces. These delivery failures are generally not the responsibility and expectation of the receiver ESP, but are primarily the result of inappropriate behavior by domain managers or users. We find that 11M (34.73%) emails are passive accidental bounces, involving three root causes. The first is server manager misconfiguration, such as incorrect configuration of SPF records. The second is improper user operation, such as the sender entering the wrong email address. The third is poor email infrastructure, which results in SMTP session timeouts.

4.3.1 Server Manager Misconfiguration. Several works [20, 26, 68] have revealed improper implementation of email security protocols in the wild. Different from them, we evaluate the impact of server misconfigurations from the perspective of bounces, which caused 5M (15.35%) email delivery failures.

Authentication mechanisms. Recalling Section 2.1, DKIM, SPF, and DMARC serve as crucial mechanisms to verify email authenticity, and they require the manager of the sender domain to correctly configure the corresponding DNS records. We find that 701K (2.19%) emails are `hard-bounced` due to sender authentication failure, which are attributed to 9K sender domains. According to the NDR messages[2], 42.09% of the emails failed both DKIM and SPF checks, 55.19% failed SPF or DKIM checks. In addition, at least 2.72% of email bounces consider the DMARC mechanism.

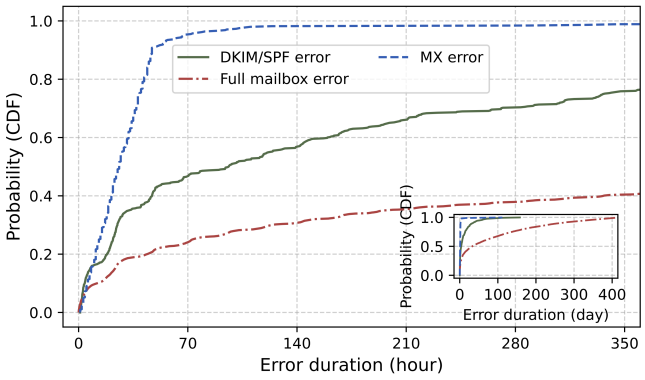

Figure 7: Distribution of the misconfiguration duration for DKIM/SPF, MX records, and mailboxes full.

Next, we analyze the duration of DKIM and SPF implementation errors. Given that NDR messages often mix the terms "DKIM" and "SPF", and ESPs generally only require senders to implement either SPF or DKIM [31, 69], we analyze them together. We find that 25.81% of sender domains maintained incorrect DKIM/SPF records consistently, and 33.72% of sender domains recurrently encountered DKIM/SPF errors. Figure 7 shows the distribution of time required by domain managers to fix DKIM/SPF errors. Unfortunately, the average fix time is 12 days, and the managers of 384 domains take more than a month to fix errors. As leading ESPs strengthen the requirements for DMARC [31, 69], server managers should promptly check their implementation of authentication mechanisms.

STARTTLS. ESPs can implement STARTTLS with three levels of strength, that is, mandatory TLS, support TLS and plaintext, not support TLS. Considering the compatibility of email services, Coremail initially establishes the SMTP session with the receiver MTA without STARTTLS. If the receiver MTA enforces TLS, Coremail immediately switches to using STARTTLS to redeliver the email. Our data reveals that 11K domains mandate STARTTLS, resulting in 572K (1.78%) emails `soft-bounced`. Additionally, we observe that popular domains are more likely to enforce TLS. Specifically, 38% of the top 100 domains in `InEmailRank` list enforce TLS, while the corresponding ratios are 8.53% for the top 10K domains.

MX records. Mismanagement of MX records by receiver domain owners inevitably leads to email delivery failures. Our results show that MX record management errors occurred in 684 receiver domains, resulting in 4M (11.37%) `hard-bounced` emails. Figure 7 plots the distribution of the configuration error duration of MX records. We find that most domains are back to normal within one day. Unfortunately, the MX records of more than 40 domains cannot be successfully resolved for a week. Overall, compared with

[2]Some examples of NDR messages. Both DKIM and SPF checks failed: "*421-4.7.0 This message does not pass authentication checks (SPF and DKIM both do not pass)*". DKIM or SPF checks failed: "*550-5.7.26 This message does not have authentication information or fails to pass authentication checks (SPF or DKIM)*". DMARC check failed: "*550-5.7.26 Unauthenticated email from (.*) is not accepted due to domain's DMARC policy*".

relatively complex DKIM/SPF records, MX record configuration errors require significantly shorter repair times.

Key Takeaway: Due to the improper implementation of email authentication mechanisms, 701K (2.19%) emails are `hard-bounced`. The average duration of DKIM/SPF errors is more than 12 days. In contrast, the vast majority of MX record errors can be fixed in one day. Furthermore, 11K domains mandate STARTTLS, so MTAs that do not support STARTTLS would encounter email bounces.

4.3.2 Improper User Operation. It is often challenging for users to completely avoid improper operations when using email services. Most of the mistakes are unconscious and usually result in `hard-bounced`. We find that 3M (9.19%) email bounces are attributed to common user mistakes, including receiver domain name typos (after @), receiver username typos (before @), full mailboxes, and inactive email addresses.

Domain name typos. The domain name typo in the receiver email address usually leads to three results: 1) the typo domain name does not provide email service; 2) the typo domain name provides email service but the username does not exist; 3) the typo domain name provides email service and the username exists. Since the latter two cases are difficult to identify accurately, we only focus on the first one below. Specifically, we use dnstwist [27] to generate 208K candidate domain name typos for the top 1K domains in `InEmailRank` list. After that, we select 16K domain names from our dataset that *never resolved successfully* via DNS, which are particularly likely to be typo domain names. Finally, we match them with candidate domain name typos.

In total, we discover 2K receiver domain names with typographical errors, of which the most common errors are omission (37.14%, e.g., "yahoo.com.cn" to "yaho.com.cn"), replacement (15.02%, e.g., "icloud.com" to 'icloyd.com'), and bitsquatting (12.34%, e.g., 'hotmail.com' to 'lotmail.com'). In particular, 49K senders typed domain name typos resulting in 89K (0.28%) email delivery failures. Even worse, some typo domain names may be exploitable by squatting attackers to commit malicious behavior (see Section 5.2). Furthermore, we find that certain domain name typos originate from TLD, with the most prevalent error being repetition (e.g., "springer.com" to "springer.comm").

Username typos. Because of the large number of usernames, detecting username typos is not straightforward. Firstly, we select 2.5M email addresses that receiver MTAs determined as non-existent usernames. Secondly, under the premise that the sender is the same, we select the receiver email addresses with more than 90% similarity to non-existent email addresses. Only if the email delivery is successful, the corresponding receiver usernames are our candidates. After the above steps, we obtain 82K corresponding (non-existent, candidate) username pairs. Thirdly, we use dnstwist [27] to verify whether non-existent usernames are in the set of typos generated by candidate usernames. If so, we consider the corresponding non-existent username to be a typo.

In total, we find 28K username typos, of which the most common errors are omission (43.92%), bitsquatting (12.83%), and replacement (10.58%). Compared with domain names, username typing errors are more common. Specifically, 24K senders type username typos resulting in 2M (6.85%) email bounces. We observe that five username typos received over 20K emails. This is most likely due to typing errors in the automatic email delivery or forwarding service.

Full mailboxes. If users do not manage their mailbox storage in time, they can no longer receive emails. We find that 75K mailboxes reached their maximum quota at least once, causing 648K (2.02%) emails to be `hard-bounced`. Furthermore, 58K mailboxes are consistently full, and 2K mailboxes repeatedly experience quota issues. Figure 7 plots the distribution of the quota error duration of receiver mailboxes. In more than 51.07% of cases, the mailbox quota problem lasts for at least 30 days. Moreover, users can only repair the full mailbox state after 86 days on average. To help users periodically clean up useless emails, the email client can remind mailbox status through out-of-band methods, such as SMS.

Inactive email addresses. If there is no use or login activity for an extended period, ESPs typically consider email addresses as inactive, thus restricting their receipt of emails. We find 3K receiver email addresses that became inactive at least once, resulting in 12K (0.04%) emails being `hard-bounced`. More seriously, 2K receiver email addresses are consistently inactive in our dataset. Inactive email addresses are likely to rely on weak or leaked passwords [53, 63], and it is difficult for users to realize that inactive accounts have been stolen by hackers. Therefore, we recommend that ESPs periodically recycle and delete inactive email accounts [7].

Key Takeaway: The improper operation of users results in a significant number of `hard-bounced` emails. 2M (6.85%) emails failed to be delivered due to receiver username typos. In addition, 648K (2.02%) emails are bounced because the receiver mailbox is full. Worryingly, we observed that over half of the mailbox quota issues last for more than a month.

4.3.3 Poor Email Infrastructure. In the following, we evaluate the impact of email infrastructure on email deliverability. If the number of emails sent from Coremail's proxy MTAs to a country/region is *N1*, and the number of `soft-bounced` emails due to SMTP session timeout is *N2*. We define the ratio of *N2* to *N1* as the poor degree of email infrastructure for that country/region. Considering the rationality of results, we exclude countries/regions with an unrepresentative number of incoming emails, as detailed in Appendix A.

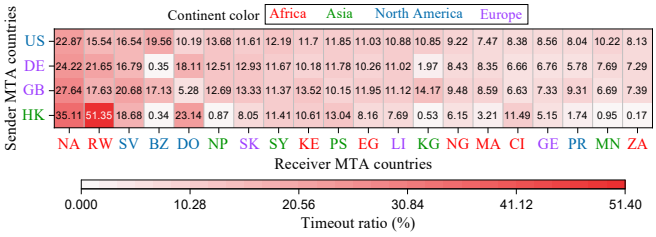

Figure 8: The poor degree of email infrastructure in different countries/regions.

We find that 3M (10.20%) emails experience SMTP session timeout due to network quality issues. Figure 8 shows the top 20 countries with the poorest email infrastructure, eight of which are located in Africa. We do not count the proxy MTAs located in Singapore and Indonesia due to their low email delivery volume. We

can see that poor email infrastructure is one of the important factors leading to the failure of email delivery. In particular, 35.11% of `soft-bounced` emails delivered from Hong Kong to Namibia experience SMTP session timeouts. Furthermore, the geographic location of the outgoing MTA also affects email deliverability. For example, email deliveries to Rwanda from proxy MTAs located in Hong Kong obviously encountered more timeout errors than other proxy MTAs. In Appendix C, we analyze the impact of email infrastructure on email delivery performance.

5 Evaluating Email Address Squatting

So far, we have revealed the root causes of email bounces in the wild. We observe that many email address typos lurk in the email delivery process. This drives us to delve into the security risks of email address squatting.

5.1 Identifying Exploitable Resources

As revealed by many works [34, 38, 49], squatting attacks on network resource names (e.g., domain, APP, container) pose a huge threat to user security. The attacker registers resource names that contain typos or are outdated to capture traffic destined to original names or disguise original services.

Both the domain name (after @) and the username (before @) of an email address could potentially be vulnerable to squatting attacks. Typically, the technical difficulty of domain name squatting is relatively low, because many registrars provide interfaces to check whether domain names are available for registration. Regarding username squatting, although the SMTP VRFY/EXPN command that indicates whether the user exists is generally disabled [48], an attacker can still determine whether a username is registrable via NDR messages or web registration interface prompts. In addition, because many providers offer free mailboxes, the financial investment of username squatting is minimal. Below, we introduce our methods of identifying email addresses that are exploitable for squatting attacks.

As for domain names, we first select 75K receiver domain names that encountered DNS resolution failures in our dataset. After that, we actively query the A records of these domain names and retain 5K domain names that return the NXDOMAIN code. Subsequently, we identified domain names available for purchase using the API provided by GoDaddy [4] on December 3, 2023.

As for usernames, we find that the NDR message containing prompts like "*non-existent user*" does not necessarily mean that the corresponding username is available for registration. Some usernames may only be temporarily frozen or not publicly open for registration. As such, we assess the scope and harm of username squatting attacks through registration prompts on the web UI. Specifically, we first select the top five receiver domains with a high incoming email count and provide registration interfaces, including "gmail.com", "hotmail.com", "yahoo.com", "outlook.com", and "aol.com". Then, we select 875 receiver email addresses with an incoming email count larger than 100 from bounced emails with NDR message type `T8`. After that, we automatically enter these 875 email addresses into the public web registration interface and follow the prompts in the web UI to determine whether the email address is registrable. Notably, no actual registration procedures are executed thereafter.

5.2 Real Risk of Squatting

Unlike Ho et al. [34], who actively generated and registered email domain name typos in 2017, we first evaluate the squatting risk of email domain names and usernames in the real world. Considering that our dataset contains only one sender ESP and the time span is limited, what we reveal is only the lower bound of security risk.

Domain squatting. We find 3k domain names that can be registered for squatting attacks, we call them `vulnerable` domain names. In total, 9K users try to send 158K emails to `vulnerable` domain names. According to the domain category [14], these senders come from governments, universities, technology companies, etc. In addition, some `vulnerable` domain names were once large companies. We find that 169 `vulnerable` domain names correspond to more than 200 email users.

Focusing on the source of the squatting threat, we observe that 38 `vulnerable` domain names are typos. For example, 172 users of a large financial company type typos in their company's domain name ("l" to "i"), resulting in 698 emails direct to a `vulnerable` domain name. This allows squatting attackers to intercept vital transaction information, affecting the company's reputation and causing financial losses. Furthermore, our dataset indicates 592 `vulnerable` domains that have historically successfully received 93K emails from 6K senders. However, owners fail to renew domain names and alert users in time, leaving their domains vulnerable to squatting attacks. For instance, we observe that after the email domain of a literature publisher expires, 148 users from 17 universities still send emails to it. Attackers can exploit the residual trust of `vulnerable` domain names, which are associated with well-known brands or historical businesses, as a means to lure potential victims.

Username squatting. Among the 875 usernames we tested, *more than one-third* (312) can be registered for squatting attacks, we call them `vulnerable` usernames. We find that 672 users try to send 46K emails to `vulnerable` usernames.

Digging deeper, 25 vulnerable usernames were working addresses in the past, and they have successfully received 235 emails. In particular, 21 of the 25 `vulnerable` usernames belong to Yahoo, suggesting that Yahoo's account re-registration strategy may be more relaxed. What's worse, 14 `vulnerable` usernames have registered for many popular websites [36], such as GitHub, Adobe, Spotify, eBay, etc. Squatting attackers can directly take over the personal information and online services of these 14 `vulnerable` usernames.

Longitudinal analysis. Figure 9 shows the number of senders and emails vulnerable to squatting attacks per week. We can observe that the harm of squatting attacks has remained stable over 64 weeks. In the worst case, the number of vulnerable senders and emails in a week is close to 2K and 25K, respectively. In particular, 45.95% `vulnerable` domain names and 33.79% `vulnerable` usernames received emails within 36 consecutive weeks. Focusing on the peak number of vulnerable emails shown in Figure 9, this is caused by many employees from a logistics service company mistakenly typing typos in their company's domain name.

Furthermore, we check the registration status of 3K `vulnerable` domain names again on February 3, 2024. The results show that

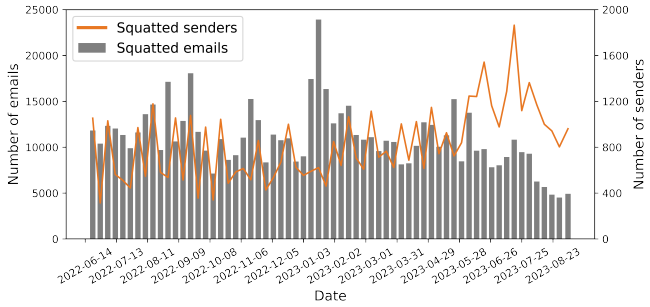

Figure 9: Number of senders and emails vulnerable to squatting attack per week.

751 `vulnerable` domain names have been re-registered, of which 105 are configured with MX records and open the TCP/25 port. Next, we investigate the registrant changes for 105 domain names through WHOIS information [15]. We find that the registrants of 59 (56.19%) domain names remained unchanged. These domain names are likely to be re-registered by previous domain owners, so the associated security risks are limited. However, the registrants of 28 (26.67%) domain names have changed. Although we cannot accurately determine whether the intentions of these registrants are malicious, our results have shown the potential threat of email address squatting.

Protective registration. To prevent squatting attacks, we carry out protective registrations for `vulnerable` domain names. Based on our dataset, we first selected `vulnerable` domain names that have not been successfully resolved. This ensures that we do not discourage the public from registering domain names they are interested in. Then, we registered 30 domain names with the highest number of email receipts. For ethical and privacy reasons, we have not deployed any online services for registered domain names to monitor related traffic. Moreover, we actively contacted the owner of the correct domain name (e.g., hotmail.com) corresponding to the domain name typo, informing them about potential security risks and guaranteeing the transfer of our registered domain names free of charge.

Since registered email addresses would automatically receive emails and transferring ownership is challenging, we opt to inform the corresponding vulnerable sender about the security risks. Notably, we send emails at a rate of one per minute and only send one email per user. In total, we sent reminder emails to 672 users and received 48 replies. Specifically, seven users realized their accidental typing errors, 13 users indicated that they located the bounce causes, and 28 users simply expressed their gratitude.

Key Takeaway: We find that 3K domain names can be exploited by squatting attacks, and they have historically received 158K emails from 9K users. Even worse, some of these domain names have been registered by users different from previous registrants. Furthermore, more than one-third of the usernames we tested are vulnerable to squatting attacks, and some are associated with accounts of many popular websites. Overall, our results reveal that attackers can consistently intercept user emails by squatting email addresses, highlighting the email community should pay more attention.

6 Discussion

6.1 Ethical Considerations

Since our study involves numerous email deliveries from real users, we must minimize any ethical risks associated with this work. While our institution lacks an Institutional Review Board (IRB), our research has been authorized and supervised by the network management department and regulatory authority of our partner. In particular, we carefully design the data collection process and analytical content by referencing previous works with similar data analysis [44, 50, 60] and authoritative principles of research ethics [1, 41]. The specific ethical considerations are as follows.

First of all, we balance the benefits and drawbacks of our research according to the principle of "Beneficence" [41], primarily considering the following four aspects.

- Scope of data collection. Theoretically, detailed data allows us to evaluate email delivery failures more deeply. However, the analysis of sensitive information in email inevitably causes significant harm to user privacy. Therefore, we do not collect the email subject and email content.
- Data analysis and storage. To enhance the protection of our dataset, we complete the analysis and storage of all data on Coremail's secure servers. This server is protected by Coremail's commercial firewall. In particular, we eventually deleted all the relevant data safely.
- Data disclosure. Due to email sensitivities and privacy concerns, we do not publish any datasets.

Regarding the principle of "Respect for Law and Public Interest" [41], we meticulously introduce the research objectives, methods, and potential risks to Coremail. We strictly abide by the privacy protection regulations and server usage guidelines agreed with Coremail. We promise not to share data with anyone outside our research team.

As for the principle of "Respect for Person" [41], we mainly need to pay attention to analyzing the impact of user email addresses. According to China's data protection law and the national laws of relevant authorities, our partners allow us to collect email addresses without user consent. In particular, email addresses are necessary for reporting security risks and implementing protective registration. Except for the notification of security risks, we do not send emails to any of the email addresses in our dataset.

Furthermore, we follow the principle of "Justice" [41] to ensure that the entities involved in our research receive considerable benefits. Our analysis of email bounces can guide the community to improve the availability of email services. In addition, we reveal and report malicious attacks to help email providers enhance their security protection. Overall, our study provides valuable insights for all participants in the email ecosystem, including the email community (e.g., understanding email bounces), ESPs (e.g., security protection), domain managers (e.g., protocol implementation), and email users (e.g., account management).

6.2 Recommendations

Below, we provide recommendations to various entities involved in the email ecosystem to improve email deliverability and prevent related security risks.

Email community. The ambiguity and irregularity of NDR messages are major obstacles to accurately understanding email delivery failures. We suggest that the IETF community build standard reports and solutions for email delivery failures. Specifically, the email community should regulate the use of various email status codes and update them to adapt to the email delivery ecosystem. In addition, the email community should widely discuss the types and standard templates for the specific error text in NDR messages. For example, we can use the "*550-5.7.26 Email from <IP address> violates the SPF policy of <domain name>*" template to indicate that the email was rejected because it did not pass SPF verification. The above efforts can help email providers and users better understand the reasons for bounced emails, thereby improving the success rate of normal email delivery.

Email provider. Our work reveals that server reputation is a key factor affecting email deliverability. The sender ESP should carefully monitor the reputation of their servers through various means, such as public DNSBLs, NDR messages, and user feedback. We recommend that the sender ESP strive to understand and comply with the NDR messages, such as following the greylisting strategy. As for the receiver ESP, we advocate that they carefully consider and evaluate the reasonableness of their blocklists, weighing the effectiveness against spam protection and potential impediments to email delivery. For example, the ESP can combine the historical delivery behavior of the host to determine its reputation.

Domain manager. Mismanagement of domain resource records usually leads to widespread email bounces. Managers should consistently monitor the compliance of their domain resource records to prevent violations of email authentication mechanisms. Moreover, large ESPs should be aware of the squatting risk caused by email address typos. The capable ESP can utilize email delivery datasets to protectively register common domain name typos.

Email user. Users should pay attention to their mailbox quota and typing address compliance to prevent email bounces and squatting risks. In addition, we suggest that users immediately reactivate or properly deactivate inactive accounts [51]. Neglecting to do so might lead to disruptions in email-related services, potential exposure of private information, and malicious exploitation.

7 Related Work

Email Security Mechanism. Mitigating the risk of plaintext transmission of email has been a primary concern of the community. Holzbauer et al. [37] and Lee et al. [45] measured the server deployment of STARTTLS by setting target domain names with different configurations. Poddebniak et al. [55] conducted a structured analysis of STARTTLS implementations and reported more than 40 security issues, such as STARTTLS stripping attacks.

Many studies have comprehensively evaluated the deployment of various email security mechanisms [22, 28, 29, 47, 62, 67]. Their results indicate the grim state of email security and that only a few servers use a combination of security mechanisms. Focusing on email authentication mechanisms, Wang et al. [68] comprehensively investigated the deployment of DKIM in the wild through active scanning and passive data analysis. Czybik et al. [26] highlighted that while the adoption rate of the SPF is on the rise, it still exhibits notable security vulnerabilities. Ashiq et al. [21] conducted large-scale measurements of DMARC and found that improper configuration could lead to traffic amplification.

In addition, some studies focus on the effectiveness of spam protection strategies. For example, [40, 54, 61] evaluate the effectiveness of spam filters and propose improvement methods. [57–59] pointed out that DNSBLs can protect against spam very well, but there are also some false positives. Holzbauer et al. [37] reported that greylisting can greatly reduce spam and does not cause problems with email delivery to large ESPs. In contrast to previous studies, our study aims to explore the interrelationship between security mechanisms and email deliverability.

Email Delivery Failure. The community currently pays insufficient attention to email delivery failures, which are closely related to the daily lives of users. Particularly, most of the work related to email bounces was done about 20 years ago.

In 2004, Afergan et al. [18] actively measured email loss, latency, and errors for 571 domains from a vantage point, with a period of one month. They found that several domains experienced sudden email loss, and some email deliveries were delayed for more than 10 days. However, their research only involves a few regions and email domains. In 2007, Clayton et al. [23] analyzed a four-week email dataset from an ISP. Their results show the number of undelivered emails, the scale of forwarded emails, and the use of mailing lists. Furthermore, Agarwal et al. [19] measured the percentage of emails lost during delivery in 2007, and proposed methods to notify recipients when emails are lost. However, none of the previous work delved into the root causes of email bounces.

Holzbauer et al. [37] analyzed the complexity of email delivery, and tested the support of *outgoing* MTAs from 436 ESPs for security mechanisms. Their results revealed wide variations in ESP adoption of new technologies, which may lead to email delivery failures. Different from them, we evaluate the impact of security policies deployed by global *incoming* MTAs on email delivery.

8 Conclusion

This paper reveals the root causes of email delivery failures in the real world, as well as evaluates the security mechanism deployment and email address squatting risk from the bounce perspective. Of the 298M emails, 38M (12.93%) failed to deliver on the first delivery, of which around one-third of them could be successfully redelivered, while the rest consistently failed to be delivered. Server reputation is a key factor for the temporary bounces, especially affecting the delivery of many normal emails. The misconfiguration of DKIM/SPF causes many permanent bounces, and the server manager needs an average of 12 days to fix them. Furthermore, unintentional email address typos result in many emails failing to reach the expected mailbox. More seriously, the remaining trust of many email addresses could be exploited by attackers, posing significant threats to users. We recommend that the community revisit email delivery failures, and discuss the report and resolution mechanism of email bounces.

Acknowledgments

We thank the shepherd and 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 (No. 62102218, 62272413), CCF-Tencent Rhino-Bird Young Faculty Open Research Fund (CCF-Tencent RAGR20230116), the "Pioneer" and "Leading Goose" R&D Program of Zhejiang, China (Grant No. 2024C03288).

Baojun Liu, Jia Zhang, and Jun Shao are both corresponding authors. Jun Shao is with the School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou 310018, China, and the Zhejiang E-Commerce Key Lab, Hangzhou 310018, China (e-mail: chn.junshao@gmail.com).

References

[1] 1979. 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.
[2] 2021. Federal Trade Commission. https://www.fcc.gov/consumers/guides/broadband-speed-guide.
[3] 2023. Fix NDR error 550 5.4.1 in Exchange Online. https://learn.microsoft.com/en-us/exchange/troubleshoot/email-delivery/ndr/fix-error-code-550-5-4-1-in-exchange-online?source=recommendations.
[4] 2023. Godaddy. Documentation: Domains API. https://developer.godaddy.com/doc/endpoint/domains.
[5] 2023. Greylisting.org. https://www.vpnreactor.com/greylisting-org-acquistion/.
[6] 2023. have i been pwned? https://haveibeenpwned.com/.
[7] 2023. Inactive Google Account Policy. https://support.google.com/accounts/answer/12418290?hl=en.
[8] 2023. Postmaster. https://sendersupport.olc.protection.outlook.com/pm/troubleshooting.aspx.
[9] 2023. Recipient address rejected: Access denied. AS(201806281). https://answers.microsoft.com/zh-hant/outlook_com/forum/all/recipient-address-rejected-access-denied/dc0e60b1-6072-473a-939a-edb6957e3cb0.
[10] 2023. SBL Delisting Procedure. https://www.spamhaus.org/sbl/delistingprocedure/.
[11] 2023. SMTP Error Codes. https://senders.yahooinc.com/smtp-error-codes/.
[12] 2023. Spamcop. https://www.spamcop.net/.
[13] 2023. Spamhaus. https://www.spamhaus.org/.
[14] 2023. Web Filter Lookup. https://www.fortiguard.com/webfilter.
[15] 2023. WHOIS History API. https://whois-history.whoisxmlapi.com/api.
[16] 2023. World Population Review. Internet Speeds by Country 2023. https://worldpopulationreview.com/country-rankings/internet-speeds-by-country.
[17] 2024. BERT community. https://huggingface.co/google-bert/.
[18] M. Afergan and R. Beverly. 2004. The state of the email address. *Comput. Commun. Rev.* 35, 1 (2004), 29–36.
[19] S. Agarwal, V. Padmanabhan, and D. Joseph. 2007. Addressing Email Loss with SureMail: Measurement, Design, and Evaluation. In *USENIX Annual Technical Conference*. 281–294.
[20] M. Ashiq, W. Li, T. Fiebig, and T. Chung. 2023. You've Got Report: Measurement and Security Implications of DMARC Reporting. In *USENIX Security*. 4123–4137.
[21] M. Ashiq, W. Li, T. Fiebig, and T. Chung. 2023. You've Got Report: Measurement and Security Implications of DMARC Reporting. In *USENIX*. 4123–4137.
[22] B. Blechschmidt and B. Stock. 2023. Extended Hell(o): A Comprehensive Large-Scale Study on Email Confidentiality and Integrity Mechanisms in the Wild. In *USENIX*. 4895–4912.
[23] R. Clayton. 2007. Email traffic: a quantitative snapshot. In *CEAS*.
[24] Coremail. 2023. https://www.coremail.cn/.
[25] D. Crocker, T. Hansen, and M. Kucherawy. 2011. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376.
[26] S. Czybik, M. Horlboge, and K. Rieck. 2023. Lazy Gatekeepers: A Large-Scale Study on SPF Configuration in the Wild. In *IMC*. ACM, 344–355.
[27] dnstwist. 2023. https://github.com/elceef/dnstwist.
[28] Z. Durumeric, D. Adrian, A. Mirian, J. Kasten, E. Bursztein, N. Lidzborski, K. Thomas, V. Eranti, M. Bailey, and J. Halderman. 2015. Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security. In *IMC*. ACM, 27–39.
[29] I. Foster, J. Larson, M. Masich, A. Snoeren, S. Savage, and K. Levchenko. 2015. Security by Any Other Name: On the Effectiveness of Provider Based Email Security. In *CCS*. ACM, 450–464.
[30] Gmail. 2021. Not receiving emails from Russian Domains that I work with. https://support.google.com/mail/thread/113761181/not-receiving-emails-from-russian-domains-that-i-work-with?hl=en.
[31] Gmail. 2024. Email sender guidelines. https://support.google.com/a/answer/81126.
[32] Evan Harris. 2003. The Next Step in the Spam Control War: Greylisting. http://projects.puremagic.com/greylisting/whitepaper.html.
[33] P. He, J. Zhu, Z. Zheng, and M. Lyu. 2017. Drain: An Online Log Parsing Approach with Fixed Depth Tree. In *ICWS*. IEEE, 33–40.
[34] G. Ho, A. Sharma, M. Javed, V. Paxson, and D. Wagner. 2017. Detecting Credential Spearphishing in Enterprise Settings. In *USENIX Security*. 469–485.
[35] P. Hoffman. 2002. SMTP Service Extension for Secure SMTP over Transport Layer Security. RFC 3207.
[36] holehe. 2024. Holehe OSINT - Email to Registered Accounts. https://github.com/megadose/holehe.
[37] F. Holzbauer, J. Ullrich, M. Lindorfer, and T. Fiebig. 2022. Not that Simple: Email Delivery in the 21st Century. In *USENIX ATC*. USENIX Association, 295–308.
[38] Y. Hu, H. Wang, R. He, L. Li, G. Tyson, I. Castro, Y. Guo, L. Wu, and G. Xu. 2020. Mobile App Squatting. In *WWW*. ACM / IW3C2, 1727–1738.
[39] ip api. 2023. IP Geolocation API. https://ip-api.com/.
[40] J. Isacenkova and D. Balzarotti. 2011. Measurement and evaluation of a real world deployment of a challenge-response spam filter. In *IMC*. ACM, 413–426.
[41] E. Kenneally and D. Dittrich. 2012. The Belmont report: ethical principles and guidelines for the protection of human subjects of research.
[42] S. Kitterman. 2014. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. RFC 7208.
[43] M. Kucherawy and E. Zwicky. 2015. Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489.
[44] D. Lain, K. Kostiainen, and S. Capkun. 2022. Phishing in Organizations: Findings from a Large-Scale and Long-Term Study. In *SP*. IEEE, 842–859.
[45] H. Lee, A. Gireesh, R. Rijswijk-Deij, T. Kwon, and T. Chung. 2020. A Longitudinal and Comprehensive Study of the DANE Ecosystem in Email. In *USENIX Security*. 613–630.
[46] J. Levine. 2010. DNS Blacklists and Whitelists. RFC 5782.
[47] R. Li, Z. Zhang, J. Shao, R. Lu, X. Jia, and G. Wei. 2024. The Potential Harm of Email Delivery: Investigating the HTTPS Configurations of Webmail Services. *IEEE Transactions on Dependable and Secure Computing (TDSC)* 21, 1 (2024), 125–138.
[48] G. Lindberg. 1999. Anti-Spam Recommendations for SMTP MTAs. RFC 2505.
[49] G. Liu, X. Gao, H. Wang, and K. Sun. 2022. Exploring the Unchartered Space of Container Registry Typosquatting. In *USENIX*. 35–51.
[50] G. Liu, L. Jin, S. Hao, Y. Zhang, D. Liu, A. Stavrou, and H. Wang. 2023. Dial "N" for NXDomain: The Scale, Origin, and Security Implications of DNS Queries to Non-Existent Domains. In *IMC*. ACM, 198–212.
[51] Y. Liu, Y. Jia, Q. Tan, Z. Liu, and L. Xing. 2022. How Are Your Zombie Accounts? Understanding Users' Practices and Expectations on Mobile App Account Deletion. In *USENIX Security*. 863–880.
[52] Microsoft. 2022. My email sent to Gmail account is bounced back. https://answers.microsoft.com/en-us/outlook_com/forum/all/my-email-sent-to-gmail-account-is-bounced-back/e326204a-435b-4e17-87ac-d72810a0e013.
[53] The Hacker News. 2024. Key Lesson from Microsoft's Password Spray Hack: Secure Every Account. https://thehackernews.com/2024/03/key-lesson-from-microsofts-password.html.
[54] A. Pitsillidis, C. Kanich, G. M. Voelker, K. Levchenko, and S. Savage. 2012. Taster's choice: a comparative analysis of spam feeds. In *IMC*. ACM, 427–440.
[55] D. Poddebniak, F. Ising, H. Böck, and S. Schinzel. 2021. Why TLS is better without STARTTLS: A Security Analysis of STARTTLS in the Email Context. In *USENIX*. 4365–4382.
[56] J. Postel. 1982. SIMPLE MAIL TRANSFER PROTOCOL. RFC 821.
[57] A. Ramachandran, D. Dagon, and N. Feamster. 2006. Can DNS-Based Blacklists Keep Up with Bots?. In *CEAS*.
[58] S. Sinha, M. Bailey, and F. Jahanian. 2010. Improving Spam Blacklisting Through Dynamic Thresholding and Speculative Aggregation. In *NDSS*. 57–64.
[59] T. Sochor. 2014. Overview of e-mail SPAM elimination and its efficiency. In *IEEE RCIS*. 1–11.
[60] C. Stransky, O. Wiese, V. Roth, Y. Acar, and S. Fahl. 2022. 27 Years and 81 Million Opportunities Later: Investigating the Use of Email Encryption for an Entire University. In *SP*. IEEE, 860–875.
[61] G. Stringhini, M. Egele, A. Zarras, T. Holz, C. Kruegel, and G. Vigna. 2012. B@bel: Leveraging Email Delivery for Spam Mitigation. In *USENIX*. 16–32.
[62] D. Tatang, F. Zettl, and T. Holz. 2021. The Evolution of DNS-based Email Authentication: Measuring Adoption and Finding Flaws. In *RAID*. ACM, 354–369.
[63] TechRepublic. 2021. Billions of passwords leaked online from past data breaches. https://www.techrepublic.com/article/billions-of-passwords-leaked-online-from-past-data-breaches.
[64] TitanHQ. 2023. Cybersecurity platforms delivering layered security. https://www.titanhq.com/.
[65] uceprotect. 2024. Removal of Level 1 Records. https://www.uceprotect.net/de/index.php?m=7&s=6.
[66] G. Vaudreuil. 2003. Enhanced Mail System Status Codes. RFC 3463.
[67] C. Wang, Y. Kuranaga, Y. Wang, M. Zhang, L. Zheng, X. Li, J. Chen, H. Duan, Y. Lin, and Q. Pan. 2024. BreakSPF: How Shared Infrastructures Magnify SPF Vulnerabilities Across the Internet. In *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*.

[68] C. Wang, K. Shen, M. Guo, Y. Zhao, M. Zhang, J. Chen, B. Liu, X. Zheng, H. Duan, Y. Lin, and Q. Pan. 2022. A Large-scale and Longitudinal Measurement Study of DKIM Deployment. In *USENIX Security*. 1185–1201.
[69] Yahoo. 2024. Sender Requirements and Recommendations. https://senders.yahooinc.com/best-practices.
[70] Y. Zhang, B. Liu, C. Lu, Z. Li, H. Duan, J. Li, and Z. Zhang. 2021. Rusted Anchors: A National Client-Side View of Hidden Root CAs in the Web PKI Ecosystem. In *CCS*. 1373–1387.

A Bounces Across ESPs, ASes, Countries

ESPs. We analyze the email bounces across the top 10 domains in `InEmailRank` list. As shown by Table 3, the bounces of emails delivered to different ESPs vary greatly. As for Gmail, `hard-bounced` emails are mostly due to the mailbox being full. We find that numerous spam were sent to Hotmail, Yahoo, and Outlook, which led to a high ratio of `hard-bounced`. Moreover, Hotmail and Outlook rejected many emails from Coremail's MTAs through Spamhaus, resulting in a high ratio of `soft-bounced`.

Table 3: Statistics of emails delivered to the top 10 domains.

Domain	Email volume	hard/soft-bounced
gmail.com	23,733,906	21.37%/3.95%
hotmail.com	4,849,243	18.24%/9.63%
yahoo.com	3,114,139	26.28%/4.41%
apple.com	2,943,240	7.39%/3.45%
bbva.com	2,912,151	2.13%/0.35%
cma-cgm.com	1,939,385	0.81%/2.57%
outlook.com	1,744,463	19.41%/12.99%
dbschenker.com	1,493,125	7.53%/3.38%
dhl.com	1,368,682	6.24%/3.46%
amazon.com	1,300,748	1.70%/2.63%

ASes. We also investigate email delivery failures across ASes. Table 4 shows the top 10 ASes in terms of the number of received emails. We can see that the bounce ratio for most ASes is about 10%. The high ratio of `hard-bounced` for AS16509 and AS714 is primarily due to receiver ESPs determining that numerous email content is not compliant.

Table 4: Statistics of emails delivered to the top 10 ASes.

AS	Email volume	hard/soft-bounced
AS8075 Microsoft Corporation	97,736,054	5.21%/5.96%
AS15169 Google LLC	40,783,693	6.74%/2.74%
AS16509 Amazon.com, Inc.	15,151,247	10.56%/8.12%
AS52129 Proofpoint, Inc.	9,071,799	1.91%/2.15%
AS22843 Proofpoint, Inc.	6,860,926	2.01/2.14%
AS26211 Proofpoint, Inc.	5,729,157	1.66%/2.15%
AS3462 Data Communication Business Group	5,402,427	2.04%/2.37%
AS714 Apple Inc.	3,839,283	11.56%/4.90%
AS16417 Cisco Systems Ironport Division	3,336,579	4.63%/3.18%
AS30238 Cisco Systems Ironport Division	3,198,858	4.36%/3.83%

Countries/regions. Next, we analyze the email bounces from different countries/regions. Considering the credibility of our results, we set thresholds to exclude unrepresentative data from our dataset. Specifically, we use ip-api [39] to obtain geographic information of all receiver MTAs. Then, we set thresholds of 1,000 for incoming emails as the unrepresentative standard for country/region data. Ultimately, we excluded 31 countries/regions, involving 21K emails. Table 5 shows the top 10 countries with the highest percentage of `hard-bounced` and `soft-bounced`. Please note that the major categories, types, and root causes correspond to the most frequently encountered bounce reasons for `hard-bounced` or `soft-bounced` emails. In addition, the email volume is the total number of emails sent by Coremail to the corresponding country.

We can see that high percentages of `hard-bounced` are mainly the responsibility of a small set of domains/attackers, which can be attributed to three reasons. Firstly, misconfigurations by the manager result in the email service unavailable. For example, we observe that all 2K emails sent to an receiver domain in Venezuela are timed out. Secondly, users send emails to numerous non-existent or inactive addresses. Confirm email subjects through Coremail's professionals, some of which are companies sending promotional or subscription emails to users. Others are some organizations recording periodic business correspondence via email, but receiver email addresses is unavailable. This emphasizes that companies and users should periodically check the availability of their mailing lists. Thirdly, attackers performe targeted email username guessing attacks (see Section 4.2.1). Considering `soft-bounced`, many email bounces are caused by Coremail not adhering well to the greylisting policy, as detailed in Section 4.2.2. In addition, the poor email infrastructure is another major cause of `soft-bounced`, which is mainly reflected in some countries located in Africa.

B NDR Message Analysis

By analyzing a passive dataset of email deliveries, we observe that many NDR messages are confusing in terms of format and meaning.

First, although the email community has standardized a series of reply codes and enhanced mail system status codes, it is actually difficult to accurately judge the bounce reason based on them. For example, RFC 821 [56] interprets the reply code 550 as "*Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access)*", and RFC 3463 [66] interprets the enhanced mail system status code 5.7.1 as "*The sender is not authorized to send to the destination. This can be the result of per-host or per-recipient filtering*". However, we observe that ESPs use 550-5.7.1 to represent many types of email bounces[3], such as the user does not exist, the source hits blocklists, the email with non-compliant content.

Second, some NDR messages contain only ambiguous keywords. Table 6 shows the templates corresponding to 6M ambiguous NDR messages in the top 200 templates. The first template accounts for 76.99% of these ambiguous NDR messages, which is a bounce reason defined by Microsoft. According to some reports [3, 9], this template may be related to the non-existent receiver address, incorrect MX record of the sender domain, etc.

Third, even if the NDR messages belong to the same type, there are huge differences in their textual expressions. For example, some

[3]Some examples of NDR message templates: "*550-5.7.1 Recipient address rejected: user (.*) does not exist*", "*550-5.7.1 This email was rejected because it violates our security policy. Remotehost is listed in the following RBL lists: SpamCop*", "*550-5.7.1 Message contains spam or virus. (.*)*", "*550-5.7.1 Messages missing a valid messageId header are not accepted.*"

Table 5: Statistics on the top 10 countries/regions with the highest percentage of `hard-bounced` and `soft-bounced`.

Rank	Country/Region	Email volume	hard/soft-bounced	Major category	Major type	Major root cause
Top 10 countries/regions with the highest hard-bounced						
1	Venezuela	4,913	57.66%/12.54%	SMTP connection error	T14 (94.95%)	Server Manager Misconfiguration
2	Tajikistan	3,275	44.31%/15.39%	Refuse email reception	T8 (51.96%)	Malicious Email Delivery
3	Belize	1,099	37.03%/24.02%	SMTP connection error	T14 (77.29%)	Server Manager Misconfiguration
4	Qatar	166,862	35.84%/2.76%	Refuse email reception	T8 (75.20%)	Improper User Operation
5	Romania	185,311	35.41%/4.33%	SMTP connection error	T14 (56.45%)	Malicious Email Delivery
6	Kyrgyzstan	5,765	30.20%/12.60%	Refuse email reception	T8 (35.73%)	Malicious Email Delivery
7	New Zealand	56,301	28.46%/7.16%	Refuse email reception	T8 (47.00%)	Malicious Email Delivery
8	Latvia	77,677	27.65%/5.29%	Refuse email reception	T8 (68.93%)	Improper User Operation
9	Iran	1,039,562	25.03%/4.84%	Refuse email reception	T8 (70.03%)	Improper User Operation
10	Myanmar	46,879	24.83%/1.41%	Refuse email reception	T8 (77.89%)	Improper User Operation
Top 10 countries/regions with the highest soft-bounced						
1	Montenegro	1,258	23.04%/33.70%	Restrict email source	T6 (96.60%)	Spam Blocking Policy
2	Zimbabwe	2,779	14.11%/26.66%	Restrict email source	T6 (72.00%)	Spam Blocking Policy
3	Belize	1,099	37.03%/24.02%	SMTP connection error	T14 (98.62%)	Poor Email Infrastructure
4	Namibia	2,113	17.61%/23.00%	SMTP connection error	T14 (93.11%)	Poor Email Infrastructure
5	Madagascar	5,714	6.81%/22.03%	Restrict email source	T6 (72.00%)	Spam Blocking Policy
6	Syria	3,136	18.37%/17.73%	SMTP connection error	T14 (56.83%)	Poor Email Infrastructure
7	Rwanda	1,567	8.93%/17.04%	SMTP connection error	T14 (89.14%)	Poor Email Infrastructure
8	Tajikistan	3,275	44.31%/15.39%	Restrict email source	T6 (51.30%)	Spam Blocking Policy
9	Slovakia	72,731	18.34%/15.14%	SMTP connection error	T14 (55.02%)	Poor Email Infrastructure
10	Brunei	1,306	8.65%/14.39%	Restrict email source	T6 (84.23%)	Spam Blocking Policy

Table 6: Top five ambiguous NDR message templates.

NDR message template	Number
(.*) 5.4.1 Recipient address rejected: Access denied. AS(201806281) (.*)	4,987,322 (76.99%)
554 5.7.1 (.*) Message rejected due to local policy. (.*)	569,243 (8.79%)
550 (.*) Mail is rejected by recipients (.*)	463,816 (7.16%)
(.*) Not allowed.(CONNECT)	335,554 (5.18%)
(.*) Relay access denied (.*)	275,957 (4.26%)

NDR message templates that represent that the receiver's mailbox is full: "*452-4.1.1 (.*) mailbox full*", "*552-5.2.2 The email account that you tried to reach is over quota and inactive*", "*501-5.0.1 (.*) has exceeded his/her disk space limit.*"

C Performance of Email Delivery

In the following, we present the global email delivery latency in the real world. Particularly, we only analyze the performance of successful email delivery and exclude countries/regions with unrepresentative data (see Appendix A).

Our results indicate that the average/medium latency of global email delivery is 19.37s/14.03s. Figure 10 shows the median delivery latency of emails delivered from Coremail MTAs to receiver MTAs in each country/region. We find that for 85.82% of countries/regions, the median delivery latency is less than 30s. Singapore (5.96s) is the country with the smallest median latency. However, the performance of email delivery in some countries/regions is poor. Cambodia (83.81s), Tanzania (77.49s), Chile (76.29s), Greenland (66.85s), and Angola (64.92s) are the five countries with the highest latency.

Delving deeper, we find that email delivery performance is correlated with Internet infrastructure investment. We determine the country bandwidth level based on the statistics of the World Population Review [16]. Bandwidth less than 25 Mbps is defined as slow Internet speed, and vice versa as fast Internet speed [2]. Our results show that the average/medium email delivery latency in countries with slow Internet speed is 16.73s/12.54s, and the corresponding latency for countries with fast Internet speed is 9.74s/6.97s. Furthermore, the geographic location of the outgoing server also affects the latency of email delivery. The average difference of median latency between Coremail proxy MTAs in different locations and receiver MTAs is 3.77s, but the corresponding difference for three countries is particularly high, i.e., Cambodia, Angola, and Bolivia. For example, the median email delivery latency from Hong Kong to Cambodia is 8.93s, while for countries/regions where other Coremail MTAs are located, it is as high as 79s.

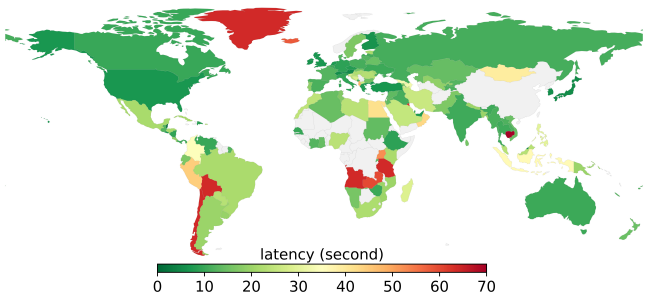

Figure 10: The median latency in email delivery for each country/region.