You've Got Report: Measurement and Security Implications of DMARC Reporting

Md. Ishtiaq Ashiq
Virginia Tech

Weitong Li
Virginia Tech

Tobias Fiebig
Max-Planck-Institut für Informatik

Taejoong Chung
Virginia Tech

Abstract

Email, since its invention, has become the most widely used communication system and SMTP is the standard for email transmission on the Internet. However, SMTP lacks built-in security features, such as sender authentication, making it vulnerable to attacks, including sender spoofing.

To address the threat of spoofing, several security extensions, such as SPF or DKIM, have been proposed. Domain-based Message Authentication Reporting and Conformance (DMARC) was introduced in 2012 as a way for domain name owners to publish desired actions for email receivers to take through a DNS record if SPF or DKIM validation fails. The DMARC record can also request email receivers to send machine-generated reports back to the specified addresses to aid domain name owners in detecting and evaluating the risk of spoofed emails.

However, DMARC's complexity creates opportunities for mismanagement that can be exploited by attackers. This paper presents a large-scale and comprehensive measurement study of DMARC reporting deployment and management. We collected data for all second-level domains under the `.com`, `.net`, `.org`, and `.se` TLDs over 13 months to analyze deployment and management from the domain name owner's perspective. Additionally, we investigated 7 popular email hosting services and 2 open-source DMARC reporting software to understand their reporting practices.

Our study reveals pervasive mismanagement and missing security considerations in DMARC reporting. For example, we found that a single email from an attacker can make a victim SMTP server receive a large number of reports with a high amplification factor (e.g., 1,460×) by exploiting misconfigured SMTP servers. Based on our findings of several operational misconfigurations for DMARC reporting, we provide recommendations for improvement.

1 Introduction

Email, also known as electronic mail, has become the most utilized form of communication worldwide since its inception in the 1970s. Despite the availability of alternative forms of communication, email usage persists to grow as it provides access to multiple platforms.

However as prevalent as email is today, it is still not robust due to its inception in a different time under different requirements. The simple mail transfer protocol (SMTP) in its original form lacks built-in security features for ensuring sender's authenticity, making it vulnerable to various security attacks such as email spoofing [7, 18].

To mitigate these threats, numerous email security protocols have been introduced (e.g., SPF [24] and DKIM [6]) to authenticate the sender by leveraging DNS; for example, domain name owners can publish `SPF` records that contain a list of IP addresses authorized to use the sender's domain name. Given the rising deployment status of these two protocols [4, 15, 38], however, there had been a lack of standard, domain-specific policies for receivers to decide how to handle messages when either SPF or DKIM validation *fails*.

Domain-based Message Authentication Reporting and Conformance (DMARC) [22] solves this problem by allowing domain name owners to publish a policy as a DNS record (i.e., `DMARC` record) specifying how email receivers should handle emails that fail either SPF or DKIM validation. Additionally, DMARC also provides a *reporting* mechanism so email senders can learn how receivers applied the selected policy to received mails via XML reports sent back to the addresses specified in the `DMARC` record. Domain name owners often face challenges in understanding the number of bad actors and how frequently they launch sender-spoofing attacks; for example, when email senders detect spoofing attempts by examining reports, they can examine the contained meta-information, including the attackers' IP addresses and the number of emails. This information helps identify and address threats promptly, even if they result from misconfigurations by the domain owner.

Unfortunately, DMARC is complex, creating many opportunities for mismanagement, often opening doors for attackers to exploit; for example, DMARC reports are usually big and compressed, and generated *automatically* from SMTP servers.

Hence, for example, attackers can put arbitrary addresses as report recipients in their DMARC record and trigger the SMTP servers to send reports, consuming resources.

Surprisingly, the DMARC reporting ecosystem has not yet been empirically analyzed. While there have been several studies on DMARC *record* deployment, we find that no prior efforts investigated how email senders and receivers handle DMARC reporting.

In this paper, we present a comprehensive study of the DMARC reporting ecosystem. To study the sender-side (i.e., report receiver), our work uses 13 months of daily DMARC record snapshots for all second-level domains of .com, .net, .org, and .se. To study the receiver-side (i.e., report sender), we perform a controlled experiment on the top 7 email hosting providers as well as two open-source software.

Our analysis reveals common misconfigurations in operational practice and missing security considerations in the specification [22] of the DMARC reporting ecosystem. Our contributions are as follows:

- First, we find that 49% of domains that use DMARC records use reporting features to receive the report, 70% of which are configured to forward the report to external domains. However, 26% of them are misconfigured, thus not able to receive the report, which also happens in the most popular domains (e.g., 10% of the top 10K most popular domains).
- Out of 7 popular email hosting providers that support DMARC reporting, we identify 6 of them do not follow the security recommendations from RFC7489 [22], which allows attackers to turn them into *reflectors* to send reports to arbitrary addresses.
- We introduce multiple attack schemes that make victims either experience (1) a large influx of unwanted traffic (i.e., reports) or (2) receive a high volume of SMTP connections from multiple sources within a brief time period.
- The survey of SMTP administrators shows a wide-spread use of misconfigured SMTP servers, even for those that manage more than 1,000 accounts.
- Finally, we provide guidelines to the community that can help protect from abusive scenarios and enable the improvement of existing standards.

Additionally, on a more positive note, our findings demonstrate several uncomplicated areas of improvement where DMARC reporting can be safely used as intended. To this end, we publicly release all of our analysis code and data to the research community at

https://dmarc-study.github.io/

thereby allowing other researchers, mail system administrators, and other stakeholders to reproduce and extend our work.

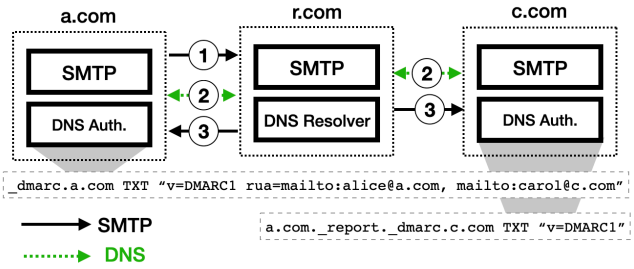

Figure 1: Overview of how DMARC reports can be delivered through SMTP along with DNS; r.com fetches DMARC records from a.com and sends report to the email addresses listed in the rua tag. When it finds an external domain in the rua tag (c.com), it should fetch a DMARC authorization record to confirm the report delegation.

2 Background

2.1 SMTP

Simple Mail Transfer Protocol (SMTP) is the standard protocol for email communication over the Internet. The process starts with the user typing his email in his Mail User Agent (MUA) which then is transmitted to the sender's Mail Transfer Agent (MTA) via SMTP itself or HTTP. Sending MTA then looks up recipient MTA's address via DNS, establishes a TCP connection with the corresponding host, and sends the email via SMTP by sending a set of commands to the recipient MTA, which includes:

- HELO (EHLO), which provides the domain name of the sender; EHLO is an alternative to HELO for servers that support the SMTP service extensions (ESMTP).
- MAIL FROM, which specifies the sender's email address.
- RCPT TO, which specifies the recipient's email address.
- DATA, after which the actual body of the email is sent.

Afterward, the email is delivered to the receiving user by the Mail Delivery Agent (MDA) via HTTP, IMAP, POP3 [23] protocols.

2.2 SMTP Security Extensions

2.2.1 Sender Authenticity

SMTP has no built-in security mechanisms; theoretically, a sender can specify any address in the MAIL FROM command to spoof the sender domain [35]. To mitigate these attacks, various security extensions have been proposed.

SPF: Sender Policy Framework (SPF) [24] allows a domain owner to publish a list of IP addresses in TXT records that are allowed to send emails for the domain; for example, a domain a.com can publish an SPF record in the DNS authoritative

server so that the receiving email server can fetch and validate the sender's IP address. The email can be rejected if the specified IP address in the record is different from the source IP address of the SMTP connection.

DKIM: DomainKeys Identified Mail (DKIM) allows a email receiver to verify the integrity of the message; an email sender can include a digital signature in the email header signed by its private key [6]. The email receiver can verify the signature by fetching the sender's corresponding public key (`TXT` records) from DNS.

DMARC: Even though SPF and DKIM offer the authenticity of the email sender, it does not tell what actions the receiver has to take when validation fails. Thus, Domain-based Message Authentication Reporting and Conformance (DMARC) [22] was proposed to allow domain owners to publish a policy that tells the receiver to follow a certain action when SPF or DKIM validation fails. The domain name owner can publish its `DMARC` record as a `TXT` record in a subdomain named "`_dmarc`"; as shown in Figure 1, the `DMARC` record of `a.com` has to be published at `_dmarc.a.com`. The domain owner can customize the DMARC policy with a combination of tags and values in the `DMARC` record. The two required tags are "v", which identifies the version (currently only "DMARC1" exists) and "p", which defines how to treat the email when validation fails; for example, the domain owner wishes to treat emails that fail in validation as spam by adding "p=quarantine" to its `DMARC` record.

2.2.2 Receiver Authentication and Confidentiality

SMTP also has no built-in security feature for encrypting messages in transit and authenticating recipient. An SMTP extension called STARTTLS was proposed in 2002 to encrypt the message using a TLS session; to do so, the receiver has to send a plain-text command "`STARTTLS`" to the sender at the initial stage of the SMTP connection. Unfortunately, a man-in-the-middle can strip out the `STARTTLS` command to force the client communicate over an unencrypted connection.

To mitigate this attack, MTA-STS and DANE were proposed to let the receiver explicitly tell the clients that (1) it supports TLS for secure email transmissions and (2) validate the receiver's identity using a policy file served through HTTPS (i.e., MTA-STS [27]) or `TLSA` records served through DNS (i.e., DANE [12]).

2.3 Reporting mechanisms

SMTP servers can send automated machine-generated reports via two mechanisms: (1) email receivers can send a DMARC report to the sender so that it can monitor authentication and judge threats. (2) Email senders also can send a TLS-RPT report to the receivers when it encounters TLS validation failure during the TLS handshake to help them debug their TLS configurations. They will also send a TLS-RPT report if any problems related to MTA-STS or DANE is encountered such as, no MTA-STS policy is found.

2.3.1 DMARC Report

DMARC aggregate feedback report (or DMARC Report) can contain useful information about authentication results; a domain owner who wishes to receive such feedback from email receivers can specify where and when to receive the report in their `DMARC` record using the three tags:

(a) `rua`, which is a list of email addresses[1] to receive an aggregated report about all emails sent from the domain, which are typically sent daily.

(b) `ruf`, which is a list of email addresses to receive a message-specific forensic report when the DMARC validation fails; since this report is (1) a more detailed containing forensics of why DMARC validation fails, thus usually bigger, and (2) generated per email, the report sender[2] may decide not to support this tag.

(c) `ri`, which is the number of seconds elapsed between sending aggregate reports to the sender.

An email receiver can fetch `DMARC` records from the sender and generate reports during the DMARC validation process of the email sender. The domain name owner may publish a time interval with a `ri` tag [22] to let the email receiver send the report after the interval, but the email receiver can override it by generating reports on their own schedule.

External Destination Verification (EDV) : A DMARC report contains useful information such as SPF and DMARC validation failures, and even, corrective action that needs to be taken by the domain owner. Reports are sent in a compressed format to the recipients while the raw report is in Extensible Markup Language (XML) which includes various types of metadata and hence, not reader-friendly. Thus, there are many third-party services that parse and analyze the report on behalf of the domain owner such as `dmarcian` [40].

In such case, an email sender `a.com`, can put an email address with a domain outside the domain of the email sender `report@r.com` in its `rua` or `ruf` tag. Thus, it is possible for a bad actor to intentionally specify a `rua` tag to redirect reports to an external email address (i.e., victim) and make it flooded with unwanted reports. To prevent this attack, a mail

[1]The RFC 7489 standard [22] permits the use of various URIs including `http`, `file`, and `ftp`. However, since our focus is on email, we will concentrate solely on the `mailto` URI. Even though it does not specify a limit on the number of addresses, it mandates that email receivers *must* support a minimum of two recipients.

[2]Throughout the paper, when referring to email receivers that send DMARC reports, we also call them *report senders*.

receiver (b.com) *should* [3] check whether the external domain has agreed to receive the report by looking up a certain DMARC record on the external domain, which is called a DMARC authorization record. More specifically, the external domain c.com needs to publish a *DMARC authorization record* at a.com._report._dmarc.c.com with value 'v=DMARC1' to tell the world that it is okay to send a.com's report to c.com.

In the DMARC authorization record, c.com may also specify a rua tag to redirect a report to another email address, which is called *rua tag overriding*; this is useful when c.com wants to update the email address without asking each of their customers (i.e., email senders) to update their rua tags. However, in order to prevent loops or indirect abuse, the domain name in the rua tag must remain same [22]. In other words, it only permits the user name in the email address to change. For example, if the rua tag of a.com forwards the report to admin@b.com, b.com cannot redirect it to other domain.

2.3.2 TLS-RPT

A DMARC report is generated and sent by the email receiver. In contrast, SMTP TLS Reporting (TLS-RPT) [28] is a reporting mechanism for *senders* to provide feedback to receivers when TLS validation fails such as STARTTLS negotiation errors or policy validation errors for DANE [20] and MTA-STS [27]. Managing TLS can often be challenging [25, 26], as a single error, such as an expired certificate, can lead to authentication failure and cause the sender to halt the SMTP connection. In such situations, the TLS-RPT mechanism can assist receivers in promptly addressing these issues.

For receivers who wish to receive TLS-RPT reports, they can publish a TLS-RPT record under a prefixed domain name: _smtp._tls; for example, a TLS-RPT policy for a domain b.com can be retrieved from _smtp._tls.b.com. The policy consists of many directives; similar to a DMARC record, it includes the rua or ruf tag that specifies a list of email addresses to receive the report. Since the report is only generated when the email cannot be delivered properly because of TLS errors, the TLS-RPT report should be delivered to an external domain. Thus, unlike the DMARC Reporting, TLS-RPT does NOT have any destination verification mechanism [28].

2.4 Threat Models

Due to the ability to send DMARC reports to multiple recipients, including external domains, email receivers can be susceptible to exploitation by attackers. This vulnerability can be leveraged to make the email receivers send multiple reports to a victim, leading to what is known as a *reflection attack*. Email receivers that support DMARC reporting can be misused as *reflectors* to achieve two goals: (1) inundating a victim SMTP server with an overwhelming volume of reports, or (2) establishing a substantial number of TCP connections within a short time window, which we call Report Reflection (RR) attack. This misuse can occur when email receivers are misconfigured, such as when they fail to perform proper email domain validation (EDV) checks or neglect to limit the number of reports they send.

[3] The EDV mechanism is not currently mandatory ("MUST") in the RFC [22], but strongly recommended ("SHOULD").

TLD	Domains MX Records	Domains with MX records		
		DMARC	DMARC	
			Report	Report from Ext.
.com	75.6 M	5.0 M (6.6%)	2.4 M (49.4%)	1.7 M (68.8%)
.net	6.5 M	453 K (6.9%)	245 K (54.1%)	172 K (70.2%)
.org	5.8 M	390 K (6.7%)	213 K (54.5%)	152 K (71.4%)
.se	848 K	81 K (9.6%)	30 K (37.4%)	24 K(80.1%)

Table 1: Overview of the dataset captured on January 8, 2023; Overall, 49% of DMARC records have rua tags to receive DMARC reports. 70% of them use external domains to forward the reports.

Preliminaries: To mount a Report Reflection (RR) attack on an SMTP email receiver, attackers should:

(a) have a domain name and control (1) a DNS authoritative to serve crafted DMARC records and (2) an SMTP server to send an email for their scanning.

(b) obtain the list of SMTP servers that they can exploit as reflectors. They can do so by sending an email to each of the interested SMTP servers to examine whether (1) it sends a DMARC report, (2) it does not aggregate the report receivers, and (3) the attacker can predict when it sends reports. The possible scenarios of misconfigurations are detailed in §4.

Goals: By launching this attack, the attackers can achieve goals or benefits, which include:

(a) **Volumetric DDoS Attack:** the attackers can initiate a volumetric DDoS attack, causing the victim to either experience (1) a large influx of unwanted traffic (i.e., reports) or (2) receive a high volume of SMTP connections from multiple sources within a brief time period.

(b) **Hiding behind the reflectors:** the attackers can remain undetected as all SMTP connections and their reports are sent by the reflectors.

(c) **Low economic barrier:** the attacker may (1) register a domain name under the TLDs that provide domain names for free such as .ml or .ga [33], (2) uses third party DNS operators such as Cloudflare DNS [9], and (3) email hosting services such as Sendinblue [36] to take advantage of the low economic barrier.

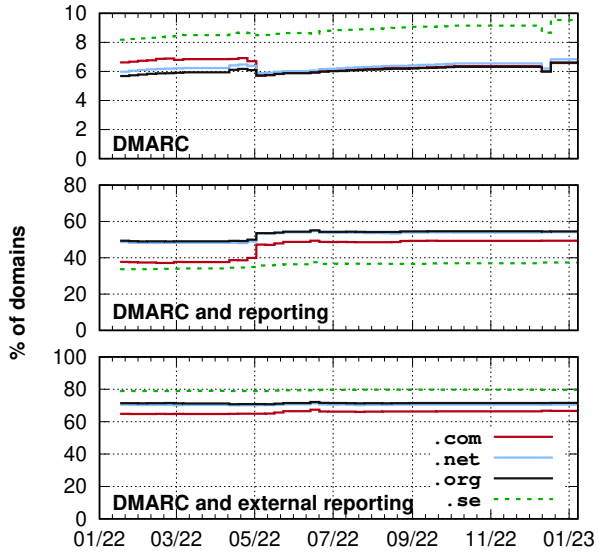

Figure 2: The deployment rate of DMARC records and their reporting feature from the dataset. Throughout our measurement period, we have observed that, on average, over 63% of domains with DMARC records utilize external domains as their reporting address.

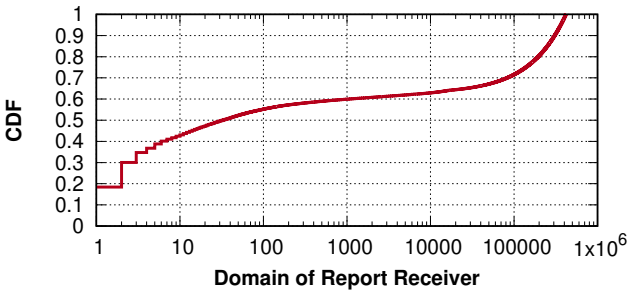

Figure 3: As of January 8, 2023, only 7 domains receive 43% of DMARC reports on behalf of domain owners.

3 Deployment for Outbound Emails

The domain owners can list email address(es) in their `rua` or `ruf` tags to receive DMARC reports from the SMTP servers that generate report in their DMARC validation process. We begin our analysis by focusing on the deployment and management of DMARC by domains.

3.1 Methodology

To cover a large number of registered domains, we use DNS scans from four TLDs: the `.com`, `.org`, and `.net` gTLDs, which are the most popular gTLDs and `.se` ccTLD, which is well known for deploying email security protocols such as DANE [26]. For each of the four TLDs, we first obtain daily zone files from their registries (`.com` and `.net` from Verisign, `.org` from Public Internet Registry, `.se` from Internetstiftelsen) to obtain Name Server (`NS`) for all second-level domains (SLDs). For each of these SLDs, we construct and fetch `DMARC` records for each domain; to avoid potential harm to small DNS authoritative servers caused by frequent DNS measurements, we capture weekly snapshots of DMARC records for each domain. The scanning process typically takes approximately 24 hours to cover all four top-level domains (TLDs). In total, our snapshots span 12 months from January 3, 2022 to January 8, 2023, which is summarized in Table 1.

3.2 Prevalence

We now examine how DMARC reporting has been deployed by domain owners by focusing on the number of second-level domains with `MX` records that (1) serve a DMARC record and (2) also have either a `rua` or `ruf` tag.

Figure 2 (top) plots the percentage of domains with `MX` records in `.com`, `.org`, `.net`, and `.se` second-level domains that have a DMARC record; We observe that the DMARC deployment rate for domains under `.com` is over 6% and 8% for `.se` domains. Domains under `.se` ccTLD have a higher adoption rate compared to other three TLDs; previous studies [8, 26] reported similar findings that `.se` (and `.nl`) domains show higher deployment rates of security protocols like DNSSEC and DANE. During our measurement period, we observe one incident that impacted on the overall `DMARC` record deployment status; on May 3rd, 2022, 1.3 M domains managed by a DNS provider, `dan.com` retracted the `DMARC` records. These records did not have any reporting addresses and therefore, we see a spike in the corresponding graph for reporting (middle).

When focusing on the domains that also have a `rua` or `ruf` tag, we see that its deployment rate is over 35% across all four TLDs; for example, 55% of `DMARC` records in `.org` and `.net` listed email addresses to receive the report. Interestingly, we also find that the vast majority of them point to external domains, implying that domain name owners outsource third party organizations to receive and analyze the report; for example, a domain, `dmarcadvisor.com` accounts for 14% of the `DMARC` records in `.se` domains.

To deep investigate the skewness of the external domains that receive the reports, Figure 3 shows the number of `DMARC` records on which each domain appears; we find a highly skewed distribution, with only 7 domains receive the 43% of `DMARC` reports on behalf of domain owners while most of them (402K (96%)) domains only receive one domain.

We next look at how DMARC reporting is deployed as a function of domain popularity. Figure 4 shows the percentage of the domains with `MX` records in the Alexa top 1M domains that have `DMARC` records (top). Among them, we also show the percentage of the ones with `rua` or `ruf` tags (bottom). We first observe that popular domains tend to have `DMARC` records; these are more likely to have a `rua` or `ruf` tag, most of which are configured to use external domains. For example, 4,530

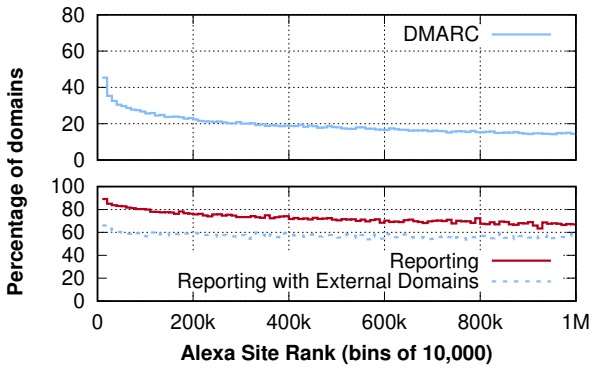

Figure 4: The percentage of domains with DMARC records and their percentages with reporting feature as a function of domain popularity from our latest snapshot.

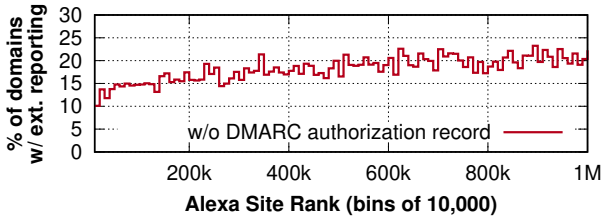

Figure 5: According to our latest snapshot, less popular domains tend to be misconfigured when their DMARC records are configured to forward DMARC reports to external domains.

(45.3%) domains have DMARC records where 87% of them have rua or ruf tag among the 10,000 most popular domains while only 1,397 (14%) domains have DMARC records where 63% of them have the reporting tags among the 10,000 least popular domains.

Key Takeaways DMARC Reporting is widely used with a predominant practice of employing external reporting addresses in the majority of cases.

3.3 DMARC Reporting Conformance

We have observed that most of the domains forward their reports to external domains. However, for external domains to receive the reports on behalf of the domain owners, they *should* publish the corresponding DMARC authorization record as described in §2. For those DMARC records of which rua tags have external domains, we send additional DNS queries to their authoritative name servers to fetch the DMARC authorization records from our latest snapshot. We also calculate the percentage of the domains with external DMARC reporting that do not have the DMARC authorization record as a function of domain popularity in Figure 5. We make a number of observations.

First of all, out of 2M DMARC records that have external domains, we find that 520K (26%) of them do not have the authorization DMARC records by getting NXDOMAIN or the responses that do not have the tag "v=DMARC1". When we look at the relationship between such misconfiguration and the domain popularity, we find prevalent misconfiguration on even most popular domains; for example, 10% of domains that forward their DMARC reports to external domains do not have the matched authorization DMARC records.

Interestingly, we also observe an increasing trend of such misconfiguration as the ranking increases; for example, almost 20% of the least 10K popular Alexa top-1M domains with DMARC that use external domains are misconfigured compared to about 10% for the most 10K popular domains. These domains would not be able to receive any DMARC reports *from the report senders that perform EDV*; in the next section, we examine how popular DMARC report senders (i.e., email receivers) perform EDV.

Key Takeaways Despite the widespread use of external domains in the rua tag, 26% of the records with external domains do not have their corresponding authorization records.

4 Deployment for Inbound Emails

In the previous section, we have analyzed the deployment of DMARC reporting and TLS-RPT for outbound emails and discovered a prevalent absence of DMARC authorization records. Now, we shift our focus to the SMTP servers that send DMARC reports for incoming emails. It is crucial for these report senders to be correctly configured from a security standpoint; for example, if they fail to perform EDV, an adversary can redirect all reports to a victim by changing their rua tags to the victim's domain.

Characterizing the report senders without sending unsolicited emails is challenging. Therefore, we focus on the top email hosting providers that send DMARC reports and popular open-source software measured from dmarcian's DMARC Data Reporters [14], which lists popular email hosting providers that support DMARC reporting. dmarcian receives and analyzes reports on behalf of domain name owners, and the top report senders are made publicly available.

We selected the top 7 email hosting providers and included recent popular providers that support DMARC reporting: Google Workspace, Amazon, Fastmail, etc. The list of the email hosting providers is shown in Table 3. It is worth mentioning that this list is distinct from the list of popular email *service* providers, as some providers only send DMARC reports when domain owners purchase a specific plan to outsource their email services. Additionally, we also tested the two most popular open-source DMARC software, OpenDMARC, and Rspamd, as per our survey, which will be described in detail in section §6.

4.1 Methodology

The purpose of these experiments is to identify the characteristics of report senders. To achieve this, we purchase a domain name, `a.com`, for our DNS authoritative server and SMTP server, which will serve as the email sender. We use BIND [1] for the authoritative server and Postfix [32] for the SMTP server. Additionally, we purchase another domain name, `r.com`, and delegate its SMTP server to a third-party email hosting provider, which will receive emails from `a.com` and examine its DMARC record to send reports. We then proceed as follows:

(a) We select an email hosting provider and subscribe to their hosting plan. The DNS settings for `r.com` are configured to delegate its MX record to the email hosting provider. When testing email software, we run a Postfix SMTP server at `a.com` and DMARC reporting software.
(b) The `DMARC` record for `a.com` is configured according to one of the experiment scenarios outlined in Table 2, which will be described in more detail later.
(c) Emails are sent to `r.com` from `a.com`.
(d) All incoming DNS queries are recorded and all reports are stored.

4.2 Characterization

To characterize the behavior of report senders, we configure the DNS authoritative servers to serve a different `DMARC` record for each experiment. A `DMARC` record can define a variety of rules using multiple tags [22], but we specifically focus on `rua` and `ri` tags, which can be exploited by attackers; more specifically, we focus on whether attackers can use them as *reflectors* to generate a massive number of reports to a victim within a very short time window.

We concentrate on four distinct perspectives of implementation vulnerabilities that can be taken advantage of by attackers: (1) *RFC Ambiguity* where the standard lacks specific implementation details, (2) *RFC Exploit* where the implementation is accurate but can still be exploited (3) *Misconfiguration* where the software improperly implements the RFC, and (4) *others*. To this end, we consider the `DMARC` record configurations as shown in Table 2:

4.2.1 RFC Ambiguity

Multiple email addresses can be specified in the `rua` or `ruf` tag; the attackers can easily increase the attack amplification factor by putting multiple email addresses in the tags.

Exp. 1: Multiple report recipients Since the RFC does not limit the number of allowed email addresses in the `rua` tag, theoretically, the attackers can put many addresses as they wish to increase the amplification factor as long as it fits to a DNS response. The attackers can first estimate the limit by specifying multiple addresses belonging to them.

Exp. 2: Duplicated email addresses Extending the first experiment, we intentionally put the same email address multiple times; if the report sender does not merge them together, attacker can easily put a single well-known email address such as `postmaster`.

Exp. 3: Subdomain de-aggregation This scenario examines whether a report sender merges subdomains into one. If the report senders send a report to a unique domain, this can be used to amplify the number of reports.[4]

4.2.2 Misconfiguration

Exp 4. External destination verification We check whether the report sender performs External Domain Verification (EDV) as suggested in the RFC document [22]. In order to do so, we purchase another domain, `b.com`, which also run an SMTP server. We configure our `DMARC` records to forward the report to both `a.com` and `b.com`. After sending an email from `a.com`, we check whether the external domain also receives the report; if so, the report sender can be exploited to make the victim receive unwanted reports.

4.2.3 RFC Exploit

Exp. 5: DMARC EDV overriding DMARC authorization records can also include a `rua` tag, which allows reports to be redirected to multiple email addresses within the same domain. However, this tag can also be used to amplify the number of reports by using multiple email addresses in the `rua` tag. For example, `a.com` in Figure 1 can have multiple email addresses (e.g., n), each of which points to different external domains. These external domains can put multiple email addresses in their `rua` tag in their `DMARC` authorization records; this results in an exponential increase in the number of reports received, as the EDV process is performed for each external domain, causing the victim to receive n^2 reports.

Exp 6. TLS-RPT external destination verification Unlike a DMARC report, a TLS-RPT report is produced by an *email sender* when TLS validation fails. TLS-RPT does not have any external domain verification mechanism since the report should be delivered to a different server that does not share the same TLS configuration as the email receiver. This can be exploited by an attacker who manipulates the recipient's SMTP

[4] Note that merging reports into the second-level domain can present operational challenges, particularly when the managing entities of the reports differ. It is important to consider this potential issue. Regardless, attackers can take advantage of situations where the report sender has not implemented a method to merge the reports.

No.	Name	Type	Rdata
Exp. 1	_dmarc.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin1@a.com, ... , mailto:admin10@a.com
Exp. 2	_dmarc.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin@a.com, ... , mailto:admin@a.com
Exp. 3	_dmarc.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin@s1.a.com, ... , mailto:admin@s10.a.com
Exp. 4	_dmarc.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin@a.com, mailto:admin@b.com
Exp. 5	_dmarc.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin@a.com, mailto:admin@b.com
	a.com._report._dmarc.b.com	TXT	v=DMARC1; rua=mailto:admin1@b.com, mailto:admin2@b.com
Exp. 6	_dmarc.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin@a.com
	_smtp._tls.a.com	TXT	v=TLSRPTv1; mailto:admin@b.com
Exp. 7	_dmarc.s1.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin@s1.a.com ri=86400
	_dmarc.s2.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin@s2.a.com ri=86400
	_dmarc.s3.a.com	TXT	v=DMARC1; p=none; rua=mailto:admin@s3.a.com ri=86400

Table 2: The configurations of DMARC records for experiments to characterize DMARC report senders

server to send DMARC reports to itself. By intentionally configuring a broken TLS setup on their end, the attacker can cause the reflector's SMTP server to send TLS-RPT reports to the victim address specified by the attacker. This will be detailed in 5.1.1. With Exp. 6, we perform whether the SMTP servers that send DMARC reports also support TLS-RPT.

4.2.4 Others

Exp. 7: Synchronous report generation Report senders may respect the ri value on the DMARC record and send the report exactly after the value. Thus, the attackers can leverage multiple report senders to make them send multiple reports *simultaneously.* Alternatively, the report sender may ignore the ri value, but generate and transmit reports to all report receivers on its schedule. To detect this, we send one email each from different subdomain one hour apart; if all the reports arrive simultaneously (or within a very short time window), it could be a strong signal that they do not respect ri values and manage their own clock. We also run this experiment five times to predict the next report generation time.

4.3 Experiment Results

We now mainly focus on the implementations that attackers can potentially exploit. Table 3 shows the results.

Limit of the email addresses in the rua tag: We find that only Fastmail and Gmail impose a limit of email addresses; note that we only list 50 email addresses, thus the other providers may have a higher limit, which attackers can exploit to increase their amplification factor. We also find that OpenDMARC has a hard limit of 255 bytes for the DMARC records, thus the records that exceed the limit raise errors. Rspamd does not have any limit, thus any number of email addresses that the DNS response size permits is allowed.[5]

[5]It depends on the resolver that Rspamd uses. Assuming that each email address takes up 20 bytes, more than 2,000 email addresses can be listed if its resolver supports EDNS0 and retries over TCP.

Report aggregation: We find that except Gmail, all the other 6 email hosting providers do not attempt to perform email address aggregation in the rua tag; for example, Google, Yahoo, and QQ send *n* duplicated emails to the same sender. Even worse, all these three providers do not implement EDV mechanisms, thus attackers can exploit them to send *n* emails to the same address. Other three providers also do not attempt aggregation when the email addresses in the rua tag share the same domain or subdomain. OpenDMARC also does not aggregate the reports while Rspamd does so.

EDV implementation: We find that Google, Yahoo, QQ, and Gmail, and OpenDMARC do not implement EDV, which raise a concern; attackers can put arbitrary email addresses on their domains and simply email these hosting providers so that they can send more than 50 emails to the victim. Amazon, Fastmail, 163, and Rspamd implements EDV correctly while only Fastmail implements the EDV overriding.

Report sending time: We find that *all* email hosting providers do not respect the ri value; each provider manages their own clock to send reports simultaneously; for example, we observe that all reports we receive from Google come at midnight (ET). For both software, we also confirm that they do not respect ri, but users can configure when to send all DMARC reports by using crontab.

5 Report Reflection (RR) Attack

Here, we introduce how email receivers that send DMARC report can be misused to mount new attacks. As shown in the previous section, some of them do not perform EDV correctly (due to their misconfiguration) and do not limit the number of email addresses in the rua tag (due to the ambiguity of the RFC standard). These two vulnerabilities are the core of our attack; attackers can use them to make the victim (1) receive a massive volume of reports or (2) establish a large number of TCP connections within a very short time window. Our attacks both exploits the misconfigured SMTP servers and the ambiguity of the RFC standards.

EHP.	Report Size (B)	# of addr.	EDV		Email Address Aggr.			RI		RUF Support
			Check (Exp. 4)	Overriding (Exp. 5)	Addr. (Exp. 2)	Domain (Exp. 1)	Subdomain (Exp. 3)	Respect (Exp. 7)	Predictable	
Google	3,962	50	✗	-	✗	✗	✗	✗	✓	✗
Yahoo	4,626	50	✗	-	✗	✗	✗	✗	✓	✗
QQ	3,628	50	✗	-	✗	✗	✗	✗	✓	✗
Gmail	3,962	1	✗	-	-	-	-	✗	✓	✗
163	4,034	50	✓	✗	✓	✗	✗	✗	✓	✗
Amazon	4,324	50	✓	✗	✓	✗	✗	✗	✗	✗
FastMail	4,839	10	✓	✓	✓	✗	✗	✗	✓	✗
OpenDMARC	2,238	8-12[6]	✗	-	✗	✗	✗	✗	✓	✓
Rspamd	2,320	50	✓	✗	✓	✗	✗	✗	✓	✗

Table 3: Table showing the 7 popular email hosting providers (EHPs) and two software that support DMARC reporting. Exp. in parentheses indicates which column maps to which experiment in §4.2. Each experiment is evaluated based on security criteria: marks are colored red if the experiment is vulnerable to exploitation by attackers, green if it is not, and black if it is neutral. Note that if email providers do not implement EDV, we do not test whether they override EDV or not (hence the –). In case of Gmail, it only sends one report to the first recipient listed in the rua tag; so, aggregation check is not tested (hence the –).

EHP	Report Size (B)	Email Address Aggr.			Predictable
		Addr.	Domain	Sub.	
Google	5,839	✓	✓	✗	✓
Comcast	5,094	✗	✓	✓	✓

Table 4: Table showing the two email hosting providers (EHPs) support TLS-RPT reporting; the other providers (Amazon, FastMail, Yahoo, 163, QQ, and Gmail) do not support it. We could not find any opensource software for TLS-RPT. The mark color scheme follows the same as in Table 3. Predictable means whether the TLS-RPT report sending time is predictable or not.

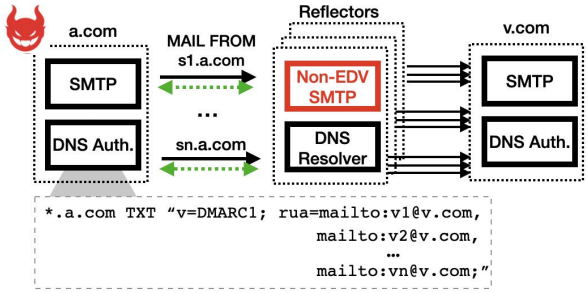

Figure 6: Attack exploiting SMTP servers that do not perform EDV correctly.

5.1 Attack 1: Inbox flooding

When email receivers do not implement EDV mechanism, attackers can exploit them as reflectors and make the victim receive multiple reports. Even if email receivers have EDV mechanisms in place, the attackers can still launch an attack by using TLS-RPT rua tags with the victim's email address.

5.1.1 Email receivers without EDV

Email receivers that do not implement EDV are vulnerable to exploitation as illustrated in Figure 6. The attackers can add multiple email addresses hosted in the victim's SMTP server in their rua tag and send an email to the reflectors. As shown in Table 3, the report size (R) is different across the email hosting providers, all of which are bigger than what is required to send an email to a reflector (approximately 200 bytes to transmit considering SMTP transactions of HELO, MAIL FROM, RCPT TO, DATA commands along with a short email content). If the email receiver does not aggregate the reports or limit the number of email addresses in the rua tag (M), the attackers can easily increase the number of reflected reports. As shown in Table 3, the report size (R) is different across the email hosting providers, all of which are bigger than 200 bytes. Thus, the amplification factor (F) for this attack would be $F = \frac{R}{200} \times M$; Google, Yahoo, and QQ are susceptible to this attack and their amplification factors are 950×, 1150×, and 900× respectively with $M = 50$. This attack can be further amplified if the email receiver also supports TLS-RPT, which is explained in detail in §5.1.2.

Email receivers using OpenDMARC are also vulnerable ($F = 110\times$), as OpenDMARC does not implement EDV nor aggregate the reports. Additionally, attackers can easily identify email servers using OpenDMARC as it puts the string, 'opendmarc-reports' and its version number in the 'X-Mailer' header of every report generated.

5.1.2 Email receivers with EDV

Even if email receivers implement the EDV mechanism, they can still be exploited as reflectors if they support TLS-RPT. When a TLS error occurs during the STARTTLS negotiation due to validation errors with DANE [20] or MTA-STS [27], the email sender will send a TLS-RPT report. Unlike DMARC reports, the EDV mechanism is not present in TLS-RPT, allowing attackers to redirect the report to any email address.

In this attack scenario, the attacker puts email addresses

[6] OpenDMARC restricts DNS records to a maximum of 255 characters.

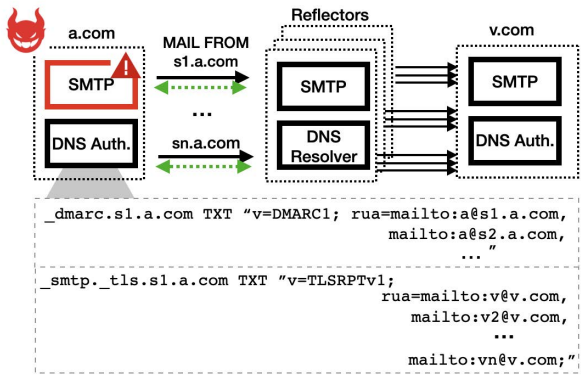

Figure 7: Attack exploiting SMTP servers that support both DMARC and TLS-RPT reporting. The attacker intentionally provides an *invalid* certificate to reflectors when they send a report to the attacker. This triggers them to send a report to the victim as specified in the `rua` tag of the `TLS-RPT` record.

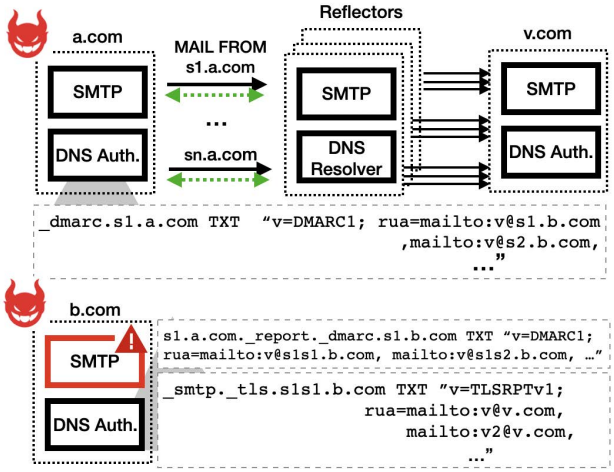

Figure 8: Attack exploiting the `rua` tag override; the reflector finds a `rua` tag, which contains a list of external domains. For each domain, it performs EDV and fetches the `DMARC` authorization record, which also contains a list of email addresses in the `rua` tag; a TLS-RPT report will be sent to each address.

under the same domain in the `rua` tag so that the reflector sends a DMARC report to the attacker. The attacker then causes a TLS error during the STARTTLS negotiation by sending a malformed certificate. The reflector, if supporting TLS-RPT, will fetch the attacker's `TLS-RPT` records, which point to the victim's email address. Since TLS-RPT allows for any email addresses to be listed, even those outside the domain's authority, the reflector will send reports to the victim. The attack scenario is illustrated in Figure 7. Assuming the number of permitted email addresses in both the DMARC `rua` tag and TLS-RPT `rua` tag is n, the attacker can make the reflector send n^2 emails to the victim. The amplification factor becomes $F = \frac{R}{200} \times m^2$ where m is the max no. of email addresses in both DMARC and TLS-RPT `rua` tag.[7] We find that `Google` is susceptible to this attack; however, it aggregates TLS-RPT reports that share the same recipient address. Thus, their amplification factor is 1,460× ($= \frac{5,839}{200} \times 50$).

DMARC EDV overriding: Attackers can amplify the number of reports further by using DMARC `rua` tag overriding. An email receiver retrieves the `DMARC` authorization record when it encounters an external domain in the `rua` tag. This authorization record can also have a `rua` tag, which overrides the email address, which replaces a single email address in the original `rua` tag with all email addresses listed in the `rua` tag of the `DMARC` authorization record.

In this attack, the attacker, `a.com`, puts n email addresses under the domain `b.com`, which is managed by the attacker. Since `b.com` is an external domain, the reflector performs EDV and retrieves its `DMARC` authorization records from `b.com`'s name server. The attacker can override each email address in the `rua` tag with n additional email addresses, all of which are subdomains of `b.com` (e.g. `s1s1.b.com`). As in the previous attack, the attacker's SMTP server intentionally breaks the TLS, causing the reflector to send TLS-RPT reports to the victim. The attacker can then include additional n email addresses in the `rua` tag, generating n^3 reports; in our experiment, however, we could not find any email hosting providers vulnerable to this attack.

[7] Astute readers may notice that the formula does not take into account the cost associated with the TLS handshake; attackers can trigger TLS-RPT reports by intentionally not supporting STARTTLS while serving `TLSA` records [12] or MTA-STS records [27];

5.2 Attack 2: SMTP connection flooding

In a previous attack, attackers could exploit email receivers that either do not support EDV or support both EDV and TLS-RPT to cause the victim to receive reports. However, as shown in Table 4, there are only 2 popular email hosting providers and no open-source software that support TLS-RPT at this moment. Therefore, our focus now shifts to email receivers that do not support TLS-RPT, and we introduce a SMTP connection flooding attack, which causes the reflectors to initiate a large number of TCP (and SMTP) connections to the victim.

All SMTP connections are built on top of TCP connections. Once the SMTP connection is established, two SMTP servers communicate with each other by exchanging SMTP commands. The sender typically sends the `MAIL FROM` and `RCPT TO` commands, and it can transmit the actual email content after receiving an "OK" response (`250`) from the receiver through the `DATA` command. An email receiver typically rejects an email if the address in the `RCPT TO` command is not associated with itself, for security purposes, by emitting a `5XX`

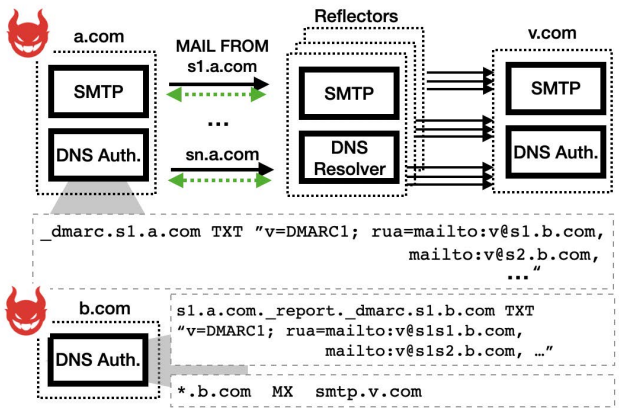

Figure 9: Attack exploiting the rua tag override. Since the TLS-RPT is not supported by the reflector, the attacker can host a wildcard `MX` record that maps to the victim's `MX` record, which makes the reflector initiate n^2 SMTP connections.

error code.[8]

This means that the email receiver (i.e. report sender) *must first establish a SMTP connection, regardless of the email destination*, which is the core of our second attack. Attackers can place many email addresses they control in the `rua` tag and *configure the `MX` record of their domains to point to the victim, causing the reflectors to initiate SMTP connections to the victim.* Additionally, the victim may *block* the incoming IP addresses of the reflectors, not allowing further SMTP connections due to the rate limit on incoming TCP connections on firewall [16].

Reflectors with EDV: Attackers can add n email addresses of their domain in the `rua` tag, all of which `MX` record maps to the victim's `MX` record, which leads to the n amplification factor. Moreover, similar to Figure 8, attackers can leverage DMARC EDV overriding by putting additional n email addresses in the `rua` tag of the `DMARC` authorization record. As shown in Figure 9, attackers can add a wildcard `MX` record to map all subdomains to the victim's `MX` record, `smtp.v.com` so that the reflector initiates n^2 SMTP connections to the victim with a single email. Since the only email hosting provider that implement `rua` tag overriding is `FastMail`; fortunately, we confirm that `FastMail` limits the total number of reports to 10 in total, thus the `rua` tag overriding does not further increase the amplification factor. However, since the `rua` tag overriding is necessary for EDV mechanism in the standard [22], attackers may be able to find such reflectors from their scanning.

[8]However, email receivers may accept an email if they are configured to support SMTP relay [21] (also known as open mail relay), which accepts all emails and forwards them to external recipients. This can be abused by spammers, thus typically disabled.

5.3 Summary

We have found that attackers can improve their attack efficiency (i.e., amplification factor) by putting multiple email addresses on the `rua` tag, redirecting TLS-RPT reports, and `rua` tag overriding. We also have introduced that attackers can induce the reflectors to initiate many SMTP connections to the victims by simply redirecting `MX` records to the victim's SMTP server.

Additionally, the following technique would not impact on the efficiency of the attack (i.e., amplification factor), but they can contribute to the attack;

(a) Attackers can utilize multiple reflectors to launch a DDoS attack; for example, the attackers can only focus on the reflectors that use `OpenDMARC`, which can be easily fingerprinted by extracting the report email headers.

(b) All email hosting providers and open-source software do not respect `ri` values; however, their sending time can be easily measured and predicted, thus attackers can only use the reflectors that send out the reports on a given time.

Responsible Disclosure: Since January 2023, we have disclosed our findings to all the email hosting providers we tested and filed a bug report to the repository of the relevant open-source software. Despite our efforts, only Google and Yahoo have responded back to us. Google classified this bug as P3-level while Yahoo dismissed our bug report as non-critical.

6 Email Sender Validation in Practice

Our passive measurement data and in-lab testing results give us an opportunity to understand the possible attack vectors from email sender validation protocols quantitatively. However, we use only publicly accessible information mainly from DNS and popular email hosting services, and examine open-source software, making it hard for us to understand how operators use these protocols and manage them, and what challenges they face. To bridge this gap to our quantitative data, we conducted a survey in late 2022 to gather a comprehensive view of the email ecosystem with respect to DMARC and TLS-RPT sending.

6.1 Survey Methodology

We recruited participants through three mailing lists: Mail Operators' List (MailOP) [2], in the North American Network Operators' Group (NANOG) [30], and the Email Security Standards for EU (MESSEU) mailing list [29]. In total, 95 participants started the survey and 74 of them answered at least one question. We summarize key demographics in Figure 10: The size of mail-setups ranges from the 16 participants that manage less than 10 accounts to the 25 participants

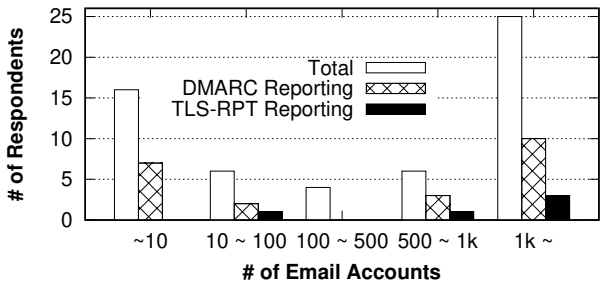

Figure 10: The distribution of how many email accounts each respondent manages and their DMARC and TLS-RPT report sending support are shown.

managing more than 1,000 accounts. Hence, we note that our survey captures the wider spectrum of mail operators. Moreover, we also have 33 participants handling setups that receive more than 1,000 emails per day, which may consider sending DMARC reports to help email senders monitor their authentication status and judge threats.

In summary, our sample contains participants operating systems covering all facets of email systems–from small to large–and hence enables us to better understand the reporting infrastructure in email, especially given the important role of mail operators' input in standardizing protocols in the ecosystem.

Ethical Considerations: Our survey targets organizations and not individuals. We do not collect any personal information, but instead factual information on deployed systems. As confirmed with our Institutional Review Board (IRB), this means that our research does not constitute human subject research. Hence, our IRB did not require an evaluation of our study protocols. Still, even though we are not executing human subject research, we followed best practices on the level required for human subject research surveys, i.e., we informed participants of their subject data access rights, and the right to stop participation and/or withdraw from the study at any point, as long as we are still able to uniquely identify their inputs.

Limitations: As common with survey based research, our survey has limitations, which we note here so our results can be appropriately interpreted. Our sample is comparatively small, and as such should be considered a qualitative perspective enriching our technical result, and not a quantitative study allowing generalization to the wider population. Even though we focused recruiting on operator centric mailinglist, and answering our questions requires domain knowledge, we did not verify respondents answers, i.e., we did not ensure that they actually operate email setups the way they claim to do so. As such, our results may suffer from self-reporting and social desirability bias, especially given the important role of security. Furthermore, participation was voluntary. As such, we may observe a self-selection bias common in operator centric surveys [11].

Hence, even though our results should not be generalized, and might suffer from common biases when handling self-reported response data, they are sufficient to provide a qualitative perspective on the aspects of email operations we are investigating in this paper. Thereby, they contribute to the explainability of our technical contributions.

6.2 DMARC Deployment and Management

DMARC reports: To understand the general demographics of respondents, we first understand how they deploy email security protocols; out of 63 operators, we find that many of them deployed security protocols; SPF (100%), DKIM (60%), DMARC (52%), STARTTLS (74%), MTA-STS (18%), and DANE (25%), which indicates that the respondents are aware of security challenges in SMTP.

Out of 39 operators who published `DMARC` records, we also find that 26 (66%) also send DMARC reports to other mail senders. A closer look at Figure 10 suggests that both large and small operators support DMARC reporting; for example, 10 (40%) operators that manage more than 1,000 accounts and 7 (43%) operators that manage less than 10 accounts both send DMARC report. Among the 26 operators that support DMARC reporting, only 5 (19%) of them also support TLS-RPT, which is in line with our findings that only two popular email hosting providers support it.

Security consideration: Out of 26 operators that support DMARC reporting, 18 (70%) answered the detailed questions regarding their configuration. We find that 6 (out of 18) uses `OpenDMARC` software for their DMARC reporting, which is vulnerable since it violates RFC recommendations in many ways as listed in Table 3; interestingly 50% of them have less than 10 accounts in their infrastructure, which suggests that smaller operators are more inclined to use `OpenDMARC`.

Interestingly, 2 (out of 18) operators that manage more than 1,000 accounts reported that they do not implement EDV mechanism. 9 (out of 18) answered they do not know; however, 4 of these operators reported that they use `OpenDMARC`, thus we find that 6 (33%) operators in total do not implement EDV mechanisms for external domains, which include both small ($\leq$ 10 accounts) and big ($\geq$ 1,000 accounts) SMTP servers.

When focusing on the maximum number of `rua` tags, out of 9 (50%) respondents who answered the question, four of them reported a defined number with 1 (two operators), 3 (one operator), and 5 (one operator). One respondent did not answer, but they use `OpenDMARC`, which is limited to 255 characters on the `rua` tag. The other 4 providers indicated that they do not impose any limit, which can be easily used

as reflectors for SMTP connection flooding attacks regardless of their implementation of EDV.

18 respondents answered the questions regarding the report interval (`ri`); 7 of them answered they respect `ri` tag, which is not common across the popular email hosting providers we surveyed. 4 of them indicated that they ignore the tag and the rest replied that they are not aware of them.

In summary, we find that 26 (41%) email providers currently support DMARC reporting. However, we also confirm the potential targets for reflection attack for those who do not impose the limit the number of report recipients in the `rua` tag (4, 6%), do not implement EDV (6, 9%), and use vulnerable software, `OpenDMARC` (6, 9%).

7 Related Work

In this section, we discuss related studies about security protocols for SMTP sender authentication.

Sender Validation Protocol: As many email security protocols that aim to authenticate senders are proposed, a few studies have focused on the deployment of SPF, DKIM, or DMARC. A few studies have examined the adoption rate of email sender validation protocols. In 2015, Durumeric et al. [13] reported the early stage of DMARC adoption; the study showed that only 1.1% domains with `MX` records have deployed DMARC. Similarly, a recent study on BIMI [39] showed that 19% domains in the Tranco list [31] has a `DMARC` record and Wang et al. [38] reported 11.9% of `DMARC` deployment among Alexa 1M domains. On the validation side, Hu et al. [18] conducted experiments on providers with email spoofing scenarios and revealed many popular providers missing DMARC validation in 2018 and tried to identify the cause behind the slow adoption of the protocol [19]. Casey et al. [10] also measured the adoption rate of SPF and DMARC validation in 2021 by sending legitimate email to user inboxes; they sent vulnerability disclosure emails to 42,924 email addresses encompassing 26K domains and found 54% of them to validate `DMARC` records. Holzbauer et al. [17] also evaluated multiple aspects of email delivery and found that 91.3%, 63%, and 53.5% of domains they tested adopted SPF, DKIM, and DMARC respectively.

Recently, BIMI (Brand Indicators for Message Identification) [5] and ARC (Authenticated Received Chain) [3] were proposed to enhance the spoofing detection, but their deployment is extremely rare [37, 39].

Attacks on Sender Validation Protocol: Recently, several studies [4, 7, 34] examined email spoofing exploits to bypass SPF, DKIM, and DMARC; for example, Bennett et al. [4] identified a buffer overflow vulnerability in one of the SPF libraries called libSPF2. Shen et al. [34] exploited automatic email forwarding service to bypass the security validation and Chen et al. [7] applied black-box fuzzing and discovered 18 types of evasion that bypass DKIM validation, which worked in 10 popular email providers.

Our work extends prior work in two ways. First, we primarily focus on DMARC reporting where none of the prior studies have focused on. We show how SMTP servers deployed for their incoming and outgoing emails using the longitudinal datasets collected by our active measurement and comprehensive survey from popular email hosting providers. Second, we introduce DDoS attack scenarios that exploit misconfiguration and missing security considerations of DMARC reporting.

To our best knowledge, this paper is the first to study focusing on the deployment, management, and their potential vulnerabilities in the DMARC reporting mechanism.

8 Concluding Discussion

We present the first comprehensive study of misconfigurations in the DMARC reporting ecosystem—encompassing 384 M domains (and their 5.9 M `DMARC` records), and 7 popular email hosting providers—focusing on measuring and explaining the security implications of how DMARC reporting is (mis)managed. We found that 49% of domains that use `DMARC` records use the reporting feature, 70% of which forward their reports to external domains. However, we observed that 26% of them are misconfigured by missing matched `DMARC` authorization records on the external domain, which is more prevalent in unpopular domains. On the report sender side, we measured 7 popular email hosting providers and two open source implementations to understand how they parse `DMARC` records and send reports.

In summary, this means that DMARC reporting–and the lived practice of how it is implemented–holds the potential for annoying Denial-of-Service attacks, especially against smaller operators when a larger operator is used as an amplifier. We found that 6 email hosting providers implemented DMARC reporting in a non-standard way which can be exploited by attackers to make them initiate multiple SMTP connections to the victim simultaneously due to them not following EDV mechanisms or falling for `MX` record redirection.

Recommendations Taken together, our results shine a light on the current status of deployment and management of DMARC reporting ecosystem providing important input that domain name owners and report senders can take to improve the DMARC reporting and standardization.

(a) DMARC report software should limit the total number of report recipients; imposing a limit only to the `rua` tag may not be enough because of the `rua` tag overriding feature.

(b) Without EDV, DMARC report senders can be easily misused as a reflector. Thus, they must implement EDV

correctly even though the current DMARC specification currently does not mandate it.

(c) Currently, the TLS-RPT specification [28] does not consider external domain validation since it is natural that the report should be delivered to another domain when TLS failure happens; however, as shown in our attacks, attackers can trigger the report senders to send TLS-RPT reports by not supporting STARTTLS. This can be easily mitigated by introducing a `TLS-RPT` authorization record served by the external domain.

Acknowledgments

We thank Tim Draegen, anonymous reviewers, and our shepherd for their helpful comments. This research was supported in part by NSF grant CNS-2053363 and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2023-00215700, Trustworthy Metaverse: blockchain-enabled convergence research].

References

[1] BIND9. https://www.isc.org/bind/.

[2] Mail Operators' List. https://www.mailop.org/.

[3] K. Andersen, B. Long, S. Blank, and M. Kucherawy. The Authenticated Received Chain (ARC) Protocol. IETF, 2019.

[4] N. Bennett, R. Sowards, and C. D. SPFail: Discovering, Measuring, and Remediating Vulnerabilities in Email Sender Validation. *IMC*, 2022.

[5] S. Blank, P. Goldsten, T. Loder, T. Zinkn, and M. Bradshaw. Brand Indicators for Message Identification (BIMI). IETF, 2021.

[6] D. Crocker, T. Hansen, and M. Kucherawy. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376, IETF, 2011. http://www.ietf.org/rfc/rfc6376.txt.

[7] J. Chen, V. Paxson, and J. Jiang. Composition kills: a case study of email sender authentication. *USENIX Security*, 2020.

[8] T. Chung, R. van Rijswijk-Deij, D. Choffnes, A. Mislove, C. Wilson, D. Levin, and B. M. Maggs. Understanding the Role of Registrars in DNSSEC Deployment. *IMC*, 2017.

[9] Cloudflare. http://www.cloudflare.com.

[10] C. Deccio, T. Yadav, N. Bennett, A. Hilton, M. Howe, T. Norton, J. Rohde, E. Tan, and B. Taylor. Measuring Email Sender Validation in the Wild. *CoNEXT*, 2021.

[11] C. Dietrich, K. Krombholz, K. Borgolte, and T. Fiebig. Investigating system operators' perspective on security misconfigurations. *CCS*, 2018.

[12] V. Dukhovni and W. Hardaker. The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance. RFC 7671, IETF, 2015.

[13] Z. Durumeric, D. Adrian, A. Mirian, J. Kasten, E. Bursztein, N. Lidzborski, K. Thomas, V. Eranti, M. Bailey, and J. A. Halderman. Neither Snow Nor Rain Nor MITM ... An Empirical Analysis of Email Delivery Security. *IMC*, 2015.

[14] DMARC Data Providers. https://dmarcian.com/dmarc-data-providers/.

[15] I. Foster, J. Larson, M. Masich, A. C. Snoeren, S. Savage, and K. Levchenko. Security by Any Other Name: On the Effectiveness of Provider Based Email Security. *CCS*, 2015.

[16] https://access.redhat.com/solutions/396273. https://access.redhat.com/solutions/396273.

[17] F. Holzbauer, J. Ullrich, M. Lindorfer, and T. Fiebig. Not that Simple: Email Delivery in the 21st Century. *USENIX ATC*, 2022.

[18] H. Hu and G. Wang. End-to-End Measurements of Email Spoofing Attacks. *USENIX Security*, 2018.

[19] H. Hu, P. Peng, and G. Wang. Towards Understanding the Adoption of Anti-Spoofing Protocols in Email Systems. *IEEE Cybersecurity Development (SecDev)*, 2018.

[20] P. Hoffman and J. Schlyter. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, IETF, 2012.

[21] J. Klensin. Simple Mail Transfer Protocol. RFC 5321, IETF, 2008. http://www.ietf.org/rfc/rfc5321.txt.

[22] M. Kucherawy and E. Zwicky. Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489, IETF, 2015. https://tools.ietf.org/html/rfc7489.

[23] P. Krumviede, R. Catoe, and D. J. C. Klensin. IMAP/POP AUTHorize Extension for Simple Challenge/Response. RFC 2195, 2195, RFC Editor, 1997.

[24] S. Kitterman. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email. RFC 7208, IETF, 2014. https://tools.ietf.org/html/rfc7208.

[25] H. Lee, M. I. Ashiq, M. Muller, R. van Rijswijk-Deij, T. Kwon, and T. Chung. Under the Hood of DANE Mismanagement in SMTP. *USENIX Security*, 2022.

[26] H. Lee, A. Girish, R. van Rijswijk-Deij, T. T. Kwon, and T. Chung. A Longitudinal and Comprehensive Study of the DANE Ecosystem in Email. *USENIX Security*, 2020.

[27] D. Margolis, M. Risher, B. Ramakrishnan, A. Brotman, and a. J. Jones. SMTP MTA Strict Transport Security (MTA-STS). RFC 8461, IETF, 2018.

[28] D. Margolis, A. Brotman, B. Ramakrishnan, J. Jones, and M. Risher. SMTP TLS Reporting. RFC 8460, 8460, RFC Editor, 2018.

[29] Modern Email Security Standards for EU (MESSEU). messeu@sys4.de.

[30] North American Network Operators' Group. https://www.nanog.org/.

[31] V. L. Pochat, T. V. Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen. TRANCO: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. *NDSS*, 2019.

[32] Postfix. http://www.postfix.org/.

[33] R. Roberts, Y. Goldschlag, R. Walter, T. Chung, A. Mislove, and D. Levin. You Are Who You Appear to Be: A Longitudinal Study of Domain Impersonation in TLS Certificates. *CCS*, 2019.

[34] K. Shen, C. Wang, M. Guo, X. Zheng, C. Lu, B. Liu, Y. Zhao, S. Hao, H. Duan, Q. Pan, and M. Yang. Weak Links in Authentication Chains: A Large-scale Analysis of Email Sender Spoofing Attacks. *USENIX Security*, 2021.

[35] Sabotage! Coping with the Joe Job. https://www.sitepoint.com/sabotage-coping-joe-job/.

[36] sendinblue. sendinblue. https://www.sendinblue.com/.

[37] C. Wang and G. Wang. Revisiting Email Forwarding Security under the Authenticated Received Chain Protocol. *WWW*, 2022.

[38] C. Wang, K. Shen, M. Guo, Y. Zhao, M. Zhang, J. Chen, B. Liu, X. Zheng, H. Duan, Y. Lin, and Q. Pan. A Large-scale and Longitudinal Measurement Study of DKIM Deployment. *USENIX Security*, 2022.

[39] M. Yajima, D. Chiba, Y. Yoneya, and a. T. Mori. A First Look at Brand Indicators for Message Identification (BIMI). *Proceedings of the 24th Passive and Active Measurement (PAM)*, 2023.

[40] dmarcian. https://dmarcian.com.