Unraveling the Complexities of MTA-STS Deployment and Management in Email

Md. Ishtiaq Ashiq
Virginia Tech
USA

Tobias Fiebig
Max-Planck Institute for Informatics
Germany

Taejoong Chung
Virginia Tech
USA

ABSTRACT

Email has been a cornerstone of online communication for decades, but its lack of built-in confidentiality has left it vulnerable to various attacks. To address this issue, two key protocols are being used: MTA-STS (Mail Transfer Agent Strict Transport Security) and DANE (DNS-based Authentication of Named Entities). While DANE was introduced first, MTA-STS has been actively adopted by major email providers like Google and Microsoft, as it does not require the complex DNSSEC chain that poses a significant challenge in deploying and managing DANE. However, despite its significance, there has been limited research on how MTA-STS is deployed and managed in practice. In this study, we present a thorough, longitudinal investigation of the MTA-STS ecosystem. We base our analysis on a dataset capturing over 87 million domains from DNS scans collected across four TLDs over 31 months, along with 10 months of additional component scanning such as TLS certificates, thereby offering a broad perspective on MTA-STS adoption and its management.

Our analysis uncovers a concerning trend of misconfigurations and inconsistencies in MTA-STS setups. In our most recent snapshot, out of 68K domains with MTA-STS record, 29.6% of domains were incorrectly configured, while 3.2% of these should encounter email delivery failure from MTA-STS supporting senders. To gain insights into the challenges faced by email administrators, we surveyed 117 operators. While awareness of MTA-STS was high (94.7%), many cited operational complexity (48.8%) and a preference for DANE (45.4%) as reasons for not deploying the protocol.

Our study not only highlights the growing importance of MTA-STS but also reveals the significant challenges in its deployment and management.

ACM Reference Format:
Md. Ishtiaq Ashiq, Tobias Fiebig, and Taejoong Chung . 2025. Unraveling the Complexities of MTA-STS Deployment and Management in Email. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Email, carried over the Simple Mail Transfer Protocol (SMTP), has been a widely used communication method for decades, but its original design lacked built-in encryption mechanisms, making it vulnerable to various security threats. Even with the later introduction of opportunistic TLS into SMTP via the `STARTTLS` [20] extension, it allowed for downgrade attacks, where an attacker strips the `STARTTLS` command from an intercepted session, preventing the use of encryption [9, 19]. Besides, the common practice of using self-signed certificates may lead to man-in-the-middle traffic interception attacks.

To mitigate these attacks, DANE (DNS-based Authentication of Named Entities) [12] has proven to be a robust protective measure. DANE relies on DNSSEC-signed [3–5] `TLSA` records, securely mapping a mail server's public key to its domain and verifying the recipient's intent to engage in TLS encryption. Despite its advantages, DANE adoption is hindered by its dependency on DNSSEC, which maintains a low global implementation rate (around 4% [18]).

To circumvent these challenges and assure SMTP transport-level encryption, MTA-STS (Mail Transfer Agent Strict Transport Security) [26] has been introduced. MTA-STS wards off `STARTTLS` downgrade or traffic interception attacks by permitting recipient domains to declare `MX` host(s) with valid PKIX certificates through DNS and HTTPS. Additionally, MTA-STS outlines a policy specifying how compliant senders should proceed if secure TLS setup cannot be established. Leading email providers, such as Google [22] and Microsoft [15], employ `MTA-STS` and enforce MTA-STS for outgoing mail.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

However, MTA-STS also introduces *additional complexity*. Not only do domain owners need to configure DNS records to publish MTA-STS policies and include the `"_mta-sts" TXT` record, but *they also need to run a web server to serve the policy file over HTTPS*, adding an additional service. To address this complexity, third-party operators have emerged to handle policy file hosting on behalf of domain owners, without necessarily being involved with mail transport; however, this setup requires domain owners to correctly configure `CNAME` records to allow these third-party operators to serve the policy using valid TLS certificates. Furthermore, the `mx` patterns specified in the policy file must match the actual `MX` records for the domain.

In this paper, we focus on the deployment and management challenges associated with MTA-STS. Our findings reveal that while MTA-STS offers a promising solution to improve email security, its practical implementation often leads to misconfigurations and inconsistencies due to the complex setup process often involving multiple parties. Consequently, such problems can lead to email delivery failures [36] and potential downgrade attacks when senders revert to opportunistic encryption due to validation failures [17]. Overall, our contributions are as follows:

- We conduct a large-scale longitudinal analysis of the MTA-STS ecosystem, revealing a significant proportion (29.6% of domains with MTA-STS records in our latest scan) to have faulty MTA-STS setups.
- We find that out of the 20,144 misconfigured domains, 640 (3.2%) domains will encounter email delivery failures from MTA-STS compliant senders.
- Our analysis shows that most individual errors arise from improper policy host configuration, particularly leading to TLS fallbacks, even with third-party policy host services. In our latest snapshot, 35% of self-managed and 3.9% of third-party policy servers fail to complete the TLS handshake.
- Our sender-side dataset reveals that 19.6% of domains perform MTA-STS validation when sending email to an MTA-STS enabled domain.
- We survey 117 email administrators to understand the practical landscape of MTA-STS and identified operational complexity as the primary bottleneck in MTA-STS deployment.

Our results highlight important points for improvement, and general mechanics that should be considered when developing security additions for established protocols. To facilitate further development and reproduction, we will publicly release all of our code, datasets and survey answers to the research community.

2 BACKGROUND

2.1 SMTP and STARTTLS

The Simple Mail Transfer Protocol (SMTP) is a fundamental protocol used for exchanging emails over the Internet. Sending an email usually starts with a sender drafting an email in their Mail User Agent (MUA). This email is then sent to the sender's Mail Transfer Agent (MTA) using SMTP or proprietary protocols over Hypertext Transfer Protocol (HTTP), e.g., when using webmail or Microsoft Exchange. The sender's MTA then looks up the recipient's MTA and its address using DNS, connects to it via TCP, and delivers the email using SMTP.

Unfortunately, SMTP lacks inherent security features like recipient authentication or end-to-end message encryption. To provide opportunistic transport encryption, the STARTTLS extension was introduced in 2002 [20].

Contrary to HTTP that uses a dedicated port for TLS, SMTP (at least for mail exchange between MTAs) uses TLS via `STARTTLS` on the same port used for plain-text communication. Regrettably, this makes STARTTLS susceptible to downgrade attacks, where a man-in-the-middle attacker can remove the `STARTTLS` command. Furthermore the adoption of Server Name Indication [2] has been slow for SMTP in comparison to HTTP, leading to many MTAs using non-matching or self-signed certificates [13, 16, 24].

2.2 MTA-STS

MTA-STS is a mechanism that allows a domain to require the use of Transport Layer Security (TLS) encryption with PKIX valid certificates for SMTP connections to specific `MX`es. It allows domain owners to specify a *policy* that lists one or more `mx` patterns matching the Mail Exchange (`MX`) hosts that support TLS for incoming email. Thus, *MTA-STS helps prevent downgrade attacks*. To deploy MTA-STS, a domain name owner, e.g., of `example.com`, needs to implement steps across three different services:

2.2.1 DNS. To signal MTA-STS support, an *MTA-STS policy record* for the zone needs to be published in DNS. The policy record is a `TXT` record with the label "_mta-sts" under the domain name[1]; for example, the domain name owner may publish the following `MTA-STS` record;

```
_mta-sts.example.com IN TXT "v=STSv1; id=20240431;"
```

The presence of this record indicates that the domain supports MTA-STS and directs email servers to fetch the policy file from the policy server. It contains two key-value pairs: `"v"`, which specifies the version of MTA-STS (currently, only `"STSv1"` is supported), and `"id"`, which uniquely identifies

[1] For brevity, we refer to this as an `MTA-STS` record in this paper.

the policy and should be updated whenever the policy file is modified. It also needs to fulfill the following conditions to be syntactically valid: 1) the record must begin with `"v=STSv1"`, 2) there cannot be more than one `TXT` records starting with `"v=STSv1"`, 3) there must be a `id` field present with an alphanumeric value, and 4) other key-value pairs can exist as extensions given that extension name and value satisfies ABNF rule in [26]. Additionally, the name for the policy server (see below), `mta-sts.example.com` must point to the right host address.

2.2.2 Policy Server. The policy server is a web server that hosts the MTA-STS policy file for a domain. The policy file is located at a defined '`.well-known`' URI:

```
https://mta-sts.example.com/.well-known/mta-sts.txt
```

where `example.com` is the policy domain that opts to support MTA-STS and the domain name for the policy server is `mta-sts.example.com`. The policy server must provide the file over HTTPS, requiring TLS validation using a PKIX valid certificate. Once the validation is successful, the policy file is fetched. This file specifies the list of `mx` patterns matching the allowed `MX` hosts for the policy domain, the `MTA-STS` version, the operating mode, and the duration for which the policy is valid and should be cached by SMTP clients, as detailed in §2.3.

2.2.3 `MX` Hosts. Every mail-receiving domain has one or more inbound MTAs, defined either explicitly through `MX` records or implicitly through the `A`/`AAAA` records at the domain apex.

The MTA-STS policy file specifies the list of `mx` patterns matching allowed `MX` hosts for which TLS encryption is required. When deploying the receiving side of MTA-STS support, a domain owner must ensure that all `MX` they intend to list, support TLS with a PKIX valid certificate [8].

2.3 MTA-STS Policies

The MTA-STS policy file is a crucial component of the MTA-STS mechanism, providing SMTP servers with the necessary information to determine TLS requirements for email delivery. The policy file consists of key-value pairs separated by `CRLF` characters, with the main components being:

- `v` (version), which specifies the version of the MTA-STS policy format. Currently, the only supported value is `"STSv1"`.
- `mode`, which indicates the expected behavior of a sending MTA in case of a policy validation failure (explained in §2.4). It can have one of three values:
  - `"enforce"`: In this mode, the sending MTA *must* enforce the MTA-STS policy. If the policy validation fails, the sending MTA *must* not deliver the email and should return an error to the sender.
  - `"testing"`: In the testing mode, the sending MTA should perform policy validation but may still deliver the email even if the validation fails. Combined with SMTP TLS Reporting [27], this mode is useful for testing and transitioning to MTA-STS without disrupting email delivery.
  - `"none"`: In this mode, the sending MTA should not perform any MTA-STS policy validation and should deliver the email as usual, regardless of the policy.
- `max_age`, which specifies the maximum lifetime of the policy in seconds; sending MTAs should cache the policy for up to this value and fetch the policy file at regular intervals.
- `mx`, which specifies allowed MX patterns; patterns can contain wildcard character. One or more of these patterns have to match with the selected `MX` host for policy validation, see §2.4.

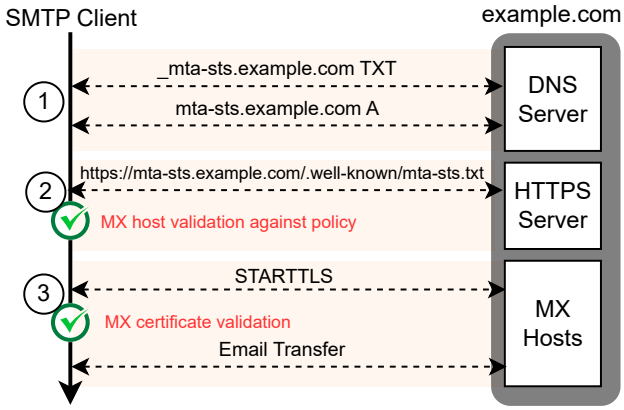

Figure 1: Flowchart of MTA-STS validation from an SMTP client: The client checks for MTA-STS support by looking up the `MTA-STS` DNS record and resolves the policy server's IP address ①; The client fetches the policy file through HTTPS, and matches the `MX` host against the `mx` patterns specified in the policy ②; The client connects to the `MX` host using STARTTLS, and validates the recipient MTA's certificate ③.

2.4 Policy Validation

When an email server wants to deliver an email to a domain with a valid `MTA-STS` DNS record present, it first fetches the policy file from the policy server over HTTPS. The downloaded policy is cached ("trusted") for up to `max_age` seconds, contingent upon a successful HTTPS connection, and is referenced for subsequent transmissions until it expires from cache [2] Next, the sender MTA confirms that the chosen `MX` host aligns with at least one of the `mx` patterns presented in

[2]This "trust on first use" (TOFU) approach can be vulnerable if the initial trust phase is compromised.

the policy file. If a match is found, the sending MTA initiates a TLS session with the corresponding mail server. If the MX host does not match any mx patterns, the MTA's behavior depends on the mode in the policy file: if enforce is specified, the MTA *must* refuse to deliver; otherwise, it can proceed.

Finally, during TLS negotiation, the MX host *must* present a valid PKIX [8] certificate, and the client must verify that the certificate accurately includes the MX host in its Common Name or Subject Alternative Names. Should the MX host lack TLS support or fail certificate validation checks, the sender either proceeds or refuses delivery depending on the mode set in the policy.

Figure 1 illustrates how an MTA perform MTA-STS validation while delivering an email.

2.5 Policy Delegation

Domain name owners can delegate MTA-STS policy hosting to a third-party service. This allows a third-party to publish and maintain the MTA-STS policy file on behalf of the domain owner, reducing the complexity for the domain owner.

To delegate the MTA-STS policy, the domain owner creates a CNAME record for the policy host ("mta-sts.example.com") pointing to the third-party provider's policy host (e.g., "mta-sts.provider.com"). This enables the provider to obtain a domain validated [7] TLS certificate and serve the policy file for the domain.

However, policy delegation can introduce complexity and potential for misconfiguration, especially *when the policy hosting provider and the email service provider are different*. In such cases, the domain owner must ensure that the policy hosted by the third-party accurately reflects the TLS capabilities of the email provider, both initially and whenever changes occur. Failure to maintain consistency between the policy and the email provider's configuration can lead to issues with email delivery and security.

2.6 Removing MTA-STS

When removing MTA-STS, domain owners must follow a proper process to avoid email delivery failures. Since sending email servers can cache both the MTA-STS DNS record and the MTA-STS policy file, abruptly removing either can cause issues.

To address this issue, RFC8461 [26] specifies the correct procedure for removing MTA-STS; (1) Publish a new policy with "none" mode and a small max_age (e.g., a day); (2) Publish a new MTA-STS record with a new id to trigger fetching the new policy; (3) Wait for the maximum amount of time specified by the previous policy's max_age and the new policy's max_age to ensure that all senders have refreshed their cached policies; (4) Remove the MTA-STS DNS record, the MTA-STS policy subdomain, and the policy file from the HTTPS endpoint.

TLD	Measurement Period	Domains with MX Records	
		Number	Percent with MTA-STS
.com	09/20/2021 – 29/09/2024	73,939,004	53,800 (0.07%)
.net		6,248,969	6,183 (0.09%)
.org		5,781,423	7,355 (0.13%)
.se		822,449	692 (0.08%)

Table 1: Overview of our datasets; the number of the domains with MTA-STS records are as-of September 29th, 2024.

3 MEASURING MTA-STS DEPLOYMENT

We begin by examining the deployment of MTA-STS, focusing specifically on domains that have implemented MTA-STS records.

3.1 Datasets

To span a wide array of registered domains, our methodology employs DNS scans across four TLDs: the .com, .org, and .net gTLDs, and the .se ccTLD. We choose these three gTLDs for their widespread adoption [40]. Meanwhile, the chosen ccTLD is notable for: (1) its proactive promotion of security protocols through registry-backed financial incentives [14], and (2) its openly available DNS zone files, providing a valuable dataset for research.

For each TLD, we acquire daily zone files from their registries (.com and .net from Verisign, .org from Public Internet Registry, .se from Internetstiftelsen). Subsequently, for every SLD, we retrieve the MTA-STS records. Similarly, we also collected the MX, and NS records of these domains. To mitigate the risk of overloading small DNS authoritative servers due to frequent DNS queries, we opt for weekly snapshots of the MTA-STS records for each domain and rate limit our queries. We collected data over 36 months, from September 9, 2021, to September 29, 2024, see Table 1.

3.2 MTA-STS Deployment

Overall we find a limited adoption of MTA-STS, while adoption slowly starts to accelerate from 2023 onward, see Figure 2. Initial adoption in 2021-10 ranged from 12,148 (0.02%) for .com to 1,916 (0.03%) domains for .org. As of 2024-09, we find adoption to have risen 3-4 times, with adoption ranging between 53,800 (0.07%) domains for .com and 7,355 (0.12%) domains for .org.

Although the deployment ratio is modest, it has been accelerating as popular email service providers have started to support MTA-STS. These providers include Google [22], Microsoft Outlook [15], Yahoo [43], and Mail.com [29]. Note that, unlike with DANE where a single operator adding TLSA

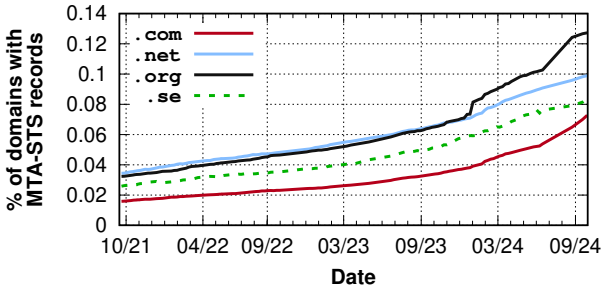

Figure 2: The deployment of `MTA-STS` records; 461 new domains related to the same organization under `.org` adopted MTA-STS in Jan 2, 2024 causing the notable spike.

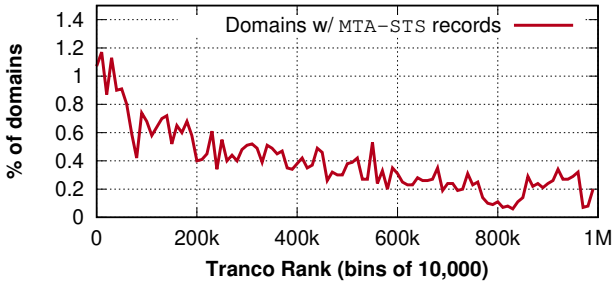

Figure 3: The percentages of domains publishing both `MX` and `MTA-STS` records as a function of website popularity based on Tranco 1M [33]; more popular websites are more inclined to deploy MTA-STS for their email services.

records enables DANE for all domains using that `MX`, MTA-STS requires *each individual domain owner* to take action. We attribute the more gradual adoption curve compared to DANE [24] to this mechanic.

Next, we examine whether popular domains are more likely to deploy MTA-STS. Figure 3 shows the percentage of the domains with `MX` records in the Tranco top 1M domains in `.com`, `.net`, `.org`, and `.se` that also publish `MTA-STS` records, as of November 1, 2024. We first observe that popular websites are more likely to have `MTA-STS` records, but the overall MTA-STS deployment remains low even among the most popular domains. For example, the average percentage of domains with `MTA-STS` records among the top 10,000 popular domains is 1.2%, while that of the bottom 10,000 popular domains is 0.4%, which suggests that although there is a positive correlation between website popularity and MTA-STS adoption, the deployment rate is still relatively low across all popularity ranges.

However, as discussed in the background, simply deploying an `MTA-STS` record does not mean correct deployment; we have to fetch the policy file from the HTTPS server, and the SMTP server also has to provide a PKIX-valid certificate. The presence of an `MTA-STS` record alone does not guarantee that MTA-STS is properly configured and operational.

4 MTA-STS MANAGEMENT

For an effective deployment and management of MTA-STS, a domain owner is required to: (1) publish an `MTA-STS` record on their name server, (2) issue a valid policy on their web server over HTTPS, and (3) provide a PKIX valid certificate from their `MX` hosts that match the patterns listed in the policy file. We now investigate whether domains with `MTA-STS` records actually fulfill these requirements.

4.1 Datasets

In this section, our aim is to delve into the *correct* deployment and functioning of MTA-STS among domains with `MX` and `MTA-STS` records. Therefore, in addition to the metrics of the dataset discussed in §4, we now also started to collect the following data:

(1) For each domain, the MTA-STS policy retrieved via HTTPS from the well-known policy domain URI (e.g., `mta-sts.example.com`).
(2) We connect to each `MX` for every domain using an instrumented SMTP client that:
   (a) Connects to the server using SMTP from a host with correctly configured forward-confirmed reverse DNS (FCrDNS) set up.
   (b) Issues an `EHLO`[3] with a name matching the reverse DNS to check for the `STARTTLS` capability[4]
   (c) Issues the `STARTTLS` command to transition the SMTP connection to TLS and retrieve the server's certificate.
   (d) Ends the connection *without* attempting to deliver an email.

This methodology was applied on a monthly basis from Nov 7, 2023 to Sep 29, 2024.

4.2 MTA-STS Validation

MTA-STS relies on several components to be implemented and in sync, recall §2.3. However, this also means that any error in any component may cause the entire MTA-STS validation to fail. Individual errors can occur in each component of MTA-STS including (1) `MTA-STS` records in DNS server (`_mta-sts.` and the delegation to the policy server on `mta-sts.`), (2) the policy file on the web server, or (3) the `MX`es themselve.

[3]Falls back to `HELO` if `EHLO` is unsupported.
[4]Some servers may not advertise `STARTTLS` due to greylisting [42] or CAPTCHAs; therefore, we focused on `MX`es that support at least some form of TLS for further analysis.

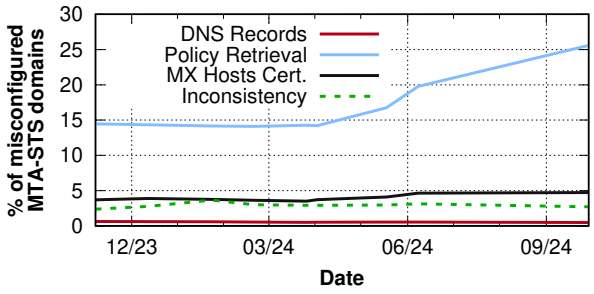

Figure 4: Percentage of MTA-STS enabled domains with errors in (1) `MTA-STS` records, (2) policy retrieval over HTTPS, (3) certificate of `MX` hosts, or (4) policy text where `mx` patterns of the policy file and `MX` records are inconsistent. Since August 2024, newly registered domains under `Porkbun LLC` [34] with invalid policy host certificates have increased policy server errors, affecting 7,237 domains in our latest snapshot.

Figure 4 shows the percentage of misconfigured MTA-STS enabled domains over time in four categories: (1) invalid `MTA-STS` records, (2) policy retrieval errors, and (3) PKIX-invalid `MX` hosts, which are individual, and (4) inconsistency errors where each component looks valid, but the `mx` patterns in the policy file do not match any `MX` records. Interestingly, we find a large portion of MTA-STS enabled domains having errors across all categories. For example, in our most recent snapshot, we find that among the 68,030 domains that have an `MTA-STS` record, 20,144 (29.6%) domains are incorrectly configured. Note that these errors are not exclusive, i.e., a domain may have multiple errors at the same time.

In the next section, we focus on each individual error and the inconsistency error to better understand why the errors occur and identify the most challenging aspects of correct MTA-STS deployment and management.

4.3 Misconfigurations: Individual Components

MTA-STS requires all components—the DNS authoritative server, `MX` hosts, and the policy server—to *correctly* adhere to their designated responsibilities. In this section, we explore misconfigurations in each of these components individually.

4.3.1 Self-hosted vs. Third-party Hosting. In the email ecosystem, both self-hosted solutions (e.g., Postfix [35]) and third-party hosting providers (e.g., Tutanota) are common. Management is simpler when the domain, `MX`es, and policy server are controlled by the same entity, as in fully self-hosted setups. In contrast, when using large email service providers, domain owners often *split* these components among multiple entities.

To investigate whether the DNS server, `MX` hosts, and policy server belong to the domain owner (i.e., self-managed) or a third party, we must rely on publicly available DNS records (`NS`, `MX`, `A`, `AAAA`). Drawing on the approach of prior work [23], we adopt a heuristic that leverages the popularity of hostnames and IPs.

(1) *Heuristic 1: Identifying Third-Party Hosting.* We label an entity as third-party if it operates mail or DNS infrastructure for many domains (≥ 50)[5] For example, `Google` hosts SMTP for over 11 million (5.8%) of domains in our dataset, and `DMARCReport` hosts policy services for 7,293 domains.
However, a complication arises when an administrator single-handedly manages multiple domains but does so using identical or nearby IP addresses. For example, 4,722 domains in our sample point their `MX` record to `mx.1.mxascen.com`, store policy files at 95.111.215.165 and 209.50.60.142 (both owned by the same operator), and share a uniform `A` record at `194.113.75.102`. Although these IPs appear "popular", they actually belong to a single administrator's self-hosted environment. To capture this nuance, we group domains by their `MX` records, `A` or `AAAA` entries, and policy-hosting IP addresses, labeling them as self-managed if they appear to be under a single administrator's control.
Another exception arises, when third-party providers assign unique hostnames to each customer yet point them all to same set of IPs. Therefore, we also consider `A` and `AAAA` records of `MX` entries to measure popularity for email hosting providers.

(2) *Heuristic 2: Identifying Self-Managed Hosting.* Classifying self-managed domains solely by low popularity can be misleading: a small footprint might also represent a new or minor third-party provider. Thus, we treat DNS servers as self-managed if they share the same second-level domain (SLD) as the queried domain's `NS` record. Similarly, an `MX` record sharing the same SLD indicates a self-managed mail server. For policy hosting, we designate any host that serves at most five[6] domains as self-managed.

Limitation: Despite our careful methodologies, there is a possibility that our approach misclassifies domains as self-managed or third-party hosted. For example, if an administrator independently manages all email and web servers for ≥ 50 domains, our methodology might incorrectly classify these domains as third-party hosted due to their prevalence

[5]We tallied effective SLDs for each `MX` and `NS` entry, then considered providers with 50 or more unique domains as third-party.

[6]Based on our dataset and operator insights, a single administrator commonly manages up to five domains in personal or small-scale setups.

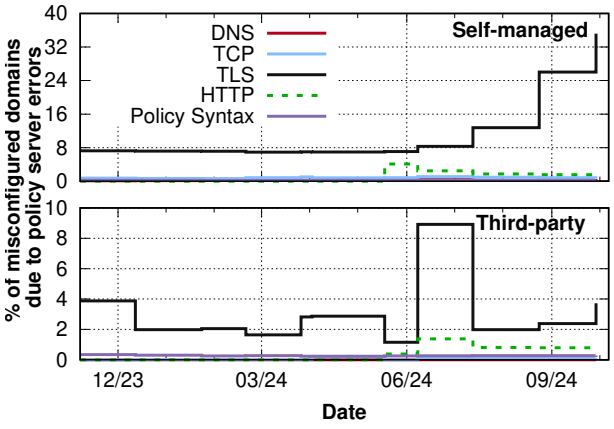

Figure 5: Percentage of MTA-STS enabled domains with misconfigured policy servers by error type and managing entity. In our latest snapshot, 9,588 (37.8%) self-managed and 1,393 (4.9%) third-party policy servers had misconfigurations. A spike on June 8, 2024, for third-party servers was due to a leading provider issuing self-signed certificates for 1,385 domains. Late spikes for self-managed domains were caused by `Porkbun` domains, as detailed in Figure 4.

in our dataset. Similarly, even if a domain owner uses a third-party DNS provider, they may still configure the DNS records themselves, retaining full control over their DNS settings.

4.3.2 Invalid `MTA-STS` records. The initial step for a domain name owner is to publish an `MTA-STS` record in their zone. The primary task for domain owners is to ensure the syntactical accuracy of the record and also make sure only one such record exists; failing to do so may result in MTA-STS being considered as not deployed [26].

During our observation period, we note that the vast majority of domains successfully publish a correct record, irrespective of who manages the zone. For example, in our most recent snapshot, 67,699 domains (99.5%, out of 68,030 domains) accurately publish `MTA-STS` records. Among 331 domains with errors, 65 (19.6%) domains have no `id` field, 203 (61%) have an invalid `id` which is not permitted by RFC8461[26] such as including `"-"`; the standard allows only alphanumeric characters [26]. 52 (15.7%) domains start with an invalid version prefix and 2 domains contain invalid extension fields (e.g., `"v=STSv1; id=1; mx: a.com; mode: testing;"`). In summary, we find MTA-STS policy records in the DNS to be generally well-managed.

4.3.3 MTA-STS Policy Retrieval Errors. Next, we focus on errors related to retrieving the MTA-STS policy via HTTPS. Previously, recall Figure 4, we found that the majority of misconfigurations in MTA-STS enabled domains occurs here. Although it may seem straightforward for a client to fetch a policy file from the web server, there are various operational aspects to consider:

(1) DNS: The domain name owner must have an `A` or `CNAME` record for the `mta-sts` subdomain pointing to the policy server.
(2) TCP: The host must have a webserver running and configured to listen on port TCP 443.
(3) TLS: The web server must present a PKIX-valid certificate; otherwise, an invalid certificate (e.g., expired) will cause the TLS handshake to fail, blocking policy file retrieval.
(4) HTTP: The web server must return an MTA-STS policy with an HTTP status code of 200.
(5) Policy Syntax: A successful response does not guarantee a *correct policy file*; the file must also be semantically valid per the standard [26].

We now analyze each error in relation to the managing entity responsible for the policy server. In our latest snapshot, among 68,030 domains with `MTA-STS` records, 53,935 (79.3%) were classified into those using third-party (28,591) and self-managed (25,344) policy servers. We also examine these errors from the domain owner's perspective, as setting up a policy server involves actions like adding `CNAME` records to enable the policy server to obtain a domain-validated TLS certificate.

In general, third-party managed policy servers are more likely to correctly deliver an MTA-STS policy compared to self-managed servers. In our latest snapshot, 1,393 (4.9%) third-party managed servers were misconfigured, compared to 9,588 (37.8%) self-managed servers, as shown in Figure 5. Below, we detail the individual failure cases encountered:

DNS errors: DNS errors (i.e., policy domain unresolvable) are rare in self-managed servers and non-existent in third-party policy servers. In our most recent snapshot, we could not resolve `A` or `AAAA` record for 42 domains that self-manage their policy file.

TCP errors: The majority of TCP errors result from closed ports or connection timeouts, indicating that these domains are not properly running their web servers. In our latest snapshot, 193 (0.7%) self-managed policy servers and only 34 (0.1%) third-party managed servers faced this issue.

TLS errors: Interestingly, TLS is the primary cause of policy server errors across all categories. In our latest snapshot, 8,871 self-managed and 1,113 third-party policy servers failed to complete the TLS handshake due to errors. For self-managed domains, the majority of TLS errors (8,385, 94.5%) stem from `Common Name` or `Subject Alternative Name` mismatches, indicating that the presented certificate does not include the correct `mta-sts.` subdomain.

However, for domains managed by third-party providers, 463 (43.6%) fail due to missing certificates installed for the domain. This is unexpected, as prior research on DANE MX hosts [23] found third-party managed systems to be *less* error-prone. However, in our findings, one third-party provider, `DMARCReport`, accounts for 354 (76.5%) of the domains with SSL alert errors.

We found that all these domains have `CNAME` records delegating their policy servers to `DMARCReport`. However, upon contacting their support, we learned that these domains have *never* been hosted on their service. This indicates the issue stems not from third-party mismanagement but from a misunderstanding of MTA-STS policy delegation by email administrators. Some administrators may have mistakenly pointed their `CNAME` records to third-party policy servers, believing they could enforce MTA-STS without proper authorization or aligning the `mx` patterns with the actual `MX` records. This misconfiguration highlights the complexities of MTA-STS implementation, particularly in policy delegation. We explore this issue further in §5.

HTTP errors: HTTP errors are relatively rare, occurring in 1,336 domains. Among these, 377 (1.5%) are self-managed domains, and 215 (0.8%) are domains outsourcing policy management. As expected, the majority (387 (65.3%)) of these HTTP errors are due to 404 error codes.

Policy Syntax errors: Syntax errors in supplied policies are rare. Only 55 self-managed domains and 81 domains using third-party policy servers provide syntactically incorrect policies. Of these, 87 (64%) errors result from domain owners misunderstanding the standard, leading to invalid `mx` patterns such as using email addresses, trailing dots, or even empty patterns.

Interestingly, we found 5 domains using third-party hosting providers serving completely empty policy files despite presenting valid TLS certificates. All 5 domains are managed by `DMARCReport`, which handles policies for approximately 7,000 domains but returns empty policy files only for these 5. We will explore this issue further in §5.

4.3.4 PKIX-invalid MX hosts. In this section, we shift our focus to the MXes, which are *expected* to provide PKIX-valid certificates. Although senders may accept invalid certificates when the `mode` in the policy file is set to either `"testing"` or `"none"`, we validate the certificates regardless of the `mode` to assess their readiness for MTA-STS and identify any misconfigurations or weaknesses in the certificate setup. When the `mode` is set to `"enforce"`, senders must not send emails if the host provides PKIX-invalid certificates [26].

Among the 68,030 domains with `MTA-STS` records in our most recent snapshot, we are able to classify 64,195 (94.4%) domains into 40,683 (59.8%) domains using third-party MXes, with the remaining 23,512 (34.6%) domains self-managing the MXes.

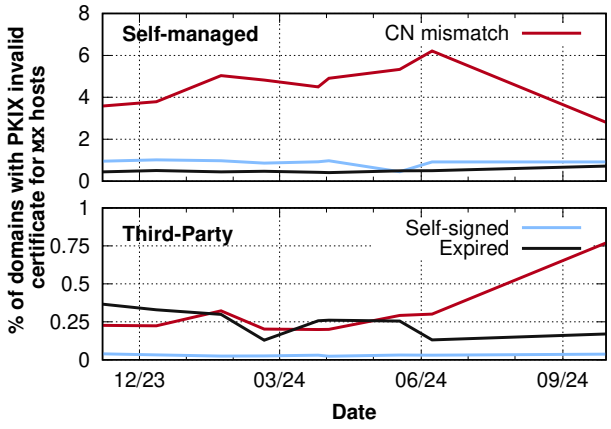

Figure 6: Percentage of MTA-STS enabled domains with PKIX invalid certificate for `MX` hosts; in our latest snapshot, 1,046 (4.4%) domains that self-manage their MX hosts and 397 (1%) domains that use email hosting services present PKIX-invalid certificates. 270 domains with self-hosted `MX` host fixed their `Common Name` mismatch error in our latest snapshot.

Figure 6 presents our finding. First, we note that third-party providers generally manage their MX hosts well as expected; only 397 (1%) domains using third-party email provider have at least one MX presenting an invalid PKIX certificate[7], compared to 1,046 (4.4%) for domains that self-manage their MXes.

Since a domain can have multiple MX hosts and may have partially invalid settings, we further divided the invalid domains into two categories as shown in Figure 7: 1) all invalid, 2) partially invalid. In our latest snapshot, 1,326 (1.9%) domains (149 (0.3%) third-party and 993 (4.2%) self-managed) are unable to present a valid TLS certificate on *all* MXes.

4.4 Misconfigurations: Inconsistency Errors

In the previous sections, we focused on individual errors that can occur in DNS records, policy servers, and `MX` hosts. However, even if each component of MTA-STS *appears* to be correctly configured, there is still a possibility that the sender may not be able to validate the recipient domain's policy when the `MX` records do not match the `mx` patterns listed in the policy file. *Especially if the* `mode` *in the policy file is* `"enforce"` *while the connected MX host does not match any of the patterns in the policy file, MTA-STS compliant senders must not deliver the message to that host.* Hence, we now

[7]Upon further investigation, we find that this issue is attributed to one large provider (`mxrouting.net`) responsible for 122 (39%, out of 313) domains

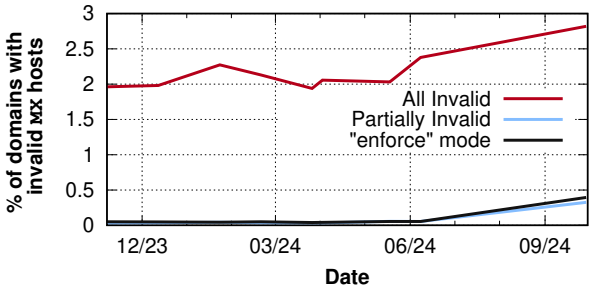

Figure 7: Percentage of MTA-STS enabled domains with `all` invalid, and `partially` invalid `MX` hosts. `"enforce"` mode presents domains with stricter policy and at least 1 invalid `MX` host; these are subject to potential email delivery failure from MTA-STS compliant senders and in our latest snapshot, we found 269 such domains.

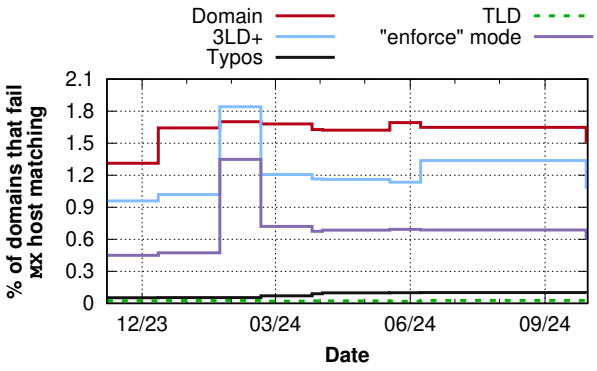

Figure 8: Percentage of MTA-STS enabled domains with mismatches between `mx` patterns listed in the policy file and actual `MX` records due to mismatches in TLD, SLD, or extra subdomains, or typos. `"enforce"` mode presents mismatched domains with stricter policy; these are subject to email delivery failure from MTA-STS compliant senders and in our latest snapshot, we found 406 such domains.

focus on domains with an inconsistency between their `MX` record and the `mx` pattern in their policy.

When inconsistency occurs, we group the possible causes as follows:

- TLD mismatch: The top-level domain of the `MX` record differs from what is in the policy file.
- Complete domain mismatch: The domain name in the `MX` record is entirely different from the `mx` patterns, lacking any meaningful overlap.
- Partial domain mismatch (3LD+): The SLD portion aligns, but further labels diverge.
- Typographical errors: Minor typos in the `mx` pattern (e.g., edit distance ≤ 3 [25]) prevent a match (note that TLD mismatches do not qualify as typos).

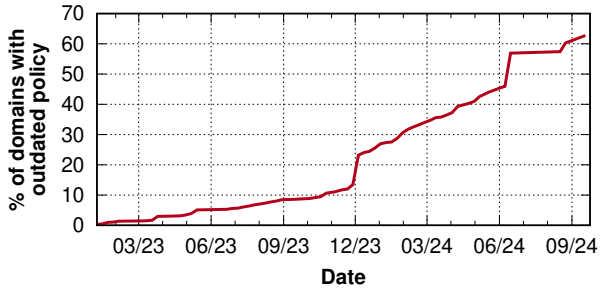

Figure 9: The percentage of domains with mismatched `mx` patterns in their MTA-STS policy file that can be correctly matched with historical `MX` records.

Figure 8 shows the results. Interestingly, we find a small but notable set of domains (63 in our latest snapshot) that have inconsistent `MX` patterns with an edit distance of ≤ 3 to at least one `MX` of the domain. Such typos are often caused by manual process [11], e.g., when the policy hosting provider asks the domain owner to manually enter their `MX` records without fetching them through DNS;[8]

Furthermore, our latest snapshot reveals 730 domains whose `mx` patterns match the effective SLD of the corresponding `MX` host but include extra labels, creating mismatches from the third label onward (`"3LD+"` in Figure 8). Upon closer inspection, we discover that among these 730 domains, 597 (81.8%) have the `mta-sts` subdomain in the policy file. We believe this is due to a misunderstanding of RFC8461 [26].

Interestingly, this issue is not limited to users who manage their own MX host or policy server. We observe a spike in `3LD+` mismatches on January 23rd, 2024. This was caused by one provider, `lucidgrow.com`, which assigns a unique `MX` host to each domain but outsources policy server management to a third-party service, `DMARCReport`. On January 23rd, for all 246 of these domains, none of their `MX` record matched with `mx` pattern in their policy file. To make matters worse, they also had the `mode` in the policy file set to `"enforce"`, which means they might have encountered email delivery failures from MTA-STS compliant senders during this period. Even though the issue was quickly resolved, it demonstrates that mismatches can occur even when domain name owners outsource their MX host and policy server management *to different entities*.

Finally, 1,023 domains have `mx` patterns that are completely different from their actual `MX`es, suggesting that the mismatch might not be related to the `MX` records captured *in*

[8] A similar problem has been observed in the context of DNSSEC, where a previous study [10] pointed out that the chain of trust can be easily broken when a DNS registrar asks the domain owner to generate and upload `DS` records, rather than fetching the `DNSKEY`s and generating the records themselves. This manual process introduces the risk of misconfigurations and inconsistencies.

the same snapshot. We investigate whether these currently mismatched MX hosts can be correctly matched with any of the domain's historical `MX` records. This situation may arise when administrators forget to update `MX` records or `mx` patterns in the policy file after migrating mail servers.

Thus, to test our hypothesis, we first consider the MTA-STS enabled domains that have domain name mismatches in our latest snapshot. Then, for each historical snapshot, we check if we can find `MX` records that match with the `mx` host in the policy, which is shown in Figure 9; interestingly, we observe an increasing trend: a majority (644, 63%) of these unmatched `mx` patterns stem from obsolete `MX` records where domain owners did not update their policy file following a change in MXes. This highlights another challenge in correctly managing MTA-STS, as it requires consistent synchronization between the policy hosts and `MX` records, which suggests that *this problem can be exacerbated if domain name owners use different third party entities for their policy host and MX servers.*

4.5 Inconsistency From Multiple Third Parties

We have observed that `mx` mismatches are amplified when policy hosting and `MX` operations are outsourced to different third parties. In MTA-STS, ensuring consistency is the domain owner's responsibility, which may be overlooked when multiple third-party services are utilized. Therefore, we now examine how inconsistencies manifest when both email and policy hosting are outsourced to different providers.

4.5.1 Methodology. For domains relying on third-party services for both `MX` hosts and policy servers, we determine whether the same provider manages both. This is straightforward, as policy delegation typically uses `CNAME` records; by comparing the SLDs or second labels of the policy host `CNAME` and `MX` records, we can infer if they share the same management. For example, Tutanota customers have `MX` records set to `mail.tutanota.de` and `CNAME` records for policy delegation set to `mta-sts.tutanota.com`, where the shared label `tutanota` indicates the same provider.

4.5.2 Results. Initially, we find 26,414 domains that use third-party services for both their `MX` host and policy server. Of those, we can further classify 18,922 domains where their hosting services are managed by different providers and 7,492 domains where the same provider manages both services in our latest snapshot. We now examine how the consistency differs depending on whether the two entities are managed by the same provider or not. As Figure 10 shows, we observe that inconsistency is almost non-existent when both entities are managed by the same provider; only 1 domain `laura-norman.com` has inconsistency error due to a typo

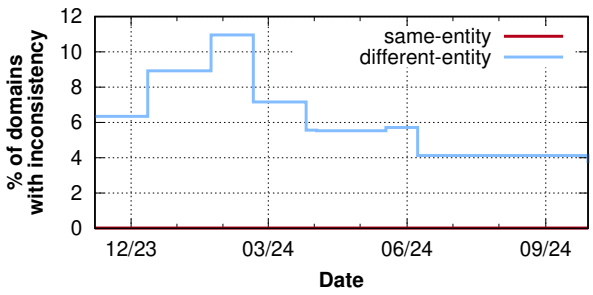

Figure 10: Inconsistency in domains outsourcing both policy servers and email hosting, based on whether the same provider manages both.

throughout all the snapshots. On the contrary, 640 (3.4%) domains have this issue when management of these components is split over to different third parties.[9]

Collectively, these observations again underscore the importance and challenge of coordination and communication between providers even when a domain outsources its MTA-STS management.

4.6 Key Takeaways

Our analysis of MTA-STS management highlights the following key insights:

(1) Policy server misconfigurations are the most common individual error in MTA-STS deployment. Even with third-party services, many domains face issues. Across all snapshots, 70-85% of errors are related to misconfigured policy servers.
(2) Self-managed email servers struggle more with maintaining PKIX-valid certificates. In our latest snapshot, 1,046 (4.4%) self-managed domains had broken PKIX configurations, compared to just 397 (1%) domains using email hosting providers.
(3) Inconsistencies between policy `mx` patterns and `MX` records persist, particularly when policy and email management are outsourced to different entities. In our latest snapshot, only 1 domain had this issue with the same provider for both, compared to 640 (3.4%) domains using different providers.

These findings underscore the challenges of MTA-STS management and emphasize the need for improved coordination and communication between domain owners and third-party providers.

[9]The spike in observed in January is also due to an email provider, `lucidgrow.com`

Providers	CNAME Patterns in TXT or A Records	# of Domains	Email Hosting Support	Behavior After Opt-out		
				Returning NXDomain	Reissuing Cert	Policy File Update
Tutanota	_mta-sts.tutanota.de	7,614	✓	✗	✗	✗
DMARCReport	a-com.mta-sts.dmarcinput.com	7,293	✗	✗	✓	↦ Empty File
PowerDMARC	a-com._mta.mta-sts.tech	3,753	✗	✓	✗	mode ↦ none
EasyDMARC	a_com__mta_sts.easydmarc.pro	2,222	✗	✗	✓	✗
Mailhardener	a.com._mta-sts.mailhardener.com	1,558	✗	✓	✗	mode ↦ none
URIports	a-com._mta-sts.uriports.com	1,100	✗	✓	✗	✗
Sendmarc	a.com._mta-sts.sdmarc.net	805	✗	✗	✓	✗
OnDMARC	_mta-sts.a.com._mta-sts.smart.ondmarc.com	451	✗	✗	✓	✗

Table 2: The list includes the top eight policy hosting providers, along with their hosting policies for customers who have opted out. The number of domains is based on our most recent data snapshot from September 29, 2024.

4.7 Responsible Disclosure

For misconfigured domains based on our latest snapshot, we sent notification emails to the `postmaster` address of 20,144 misconfigured domains from the latest snapshot between 22 Oct, 2024 and Nov 6, 2024. Unfortunately, more than 5,000 emails bounced for various reasons and as expected in prior work [38]. Given the comparably low impact and high visibility—likely accelerating mitigation—of the observed issues, and based on feedback from operators who *did* receive these mails, we decided to not attempt alternate notification channels. After the end of the notification campaign, we observed that 2,064 (10%) misconfigured domains had their issue(s) resolved, even though this may have been independent of our notifications.

5 MTA-STS POLICY DELEGATION

In the previous section, we explored how the contents of delegated MTA-STS policies can lead to validation errors. However, errors may also arise from the delegation of MTA-STS policy hosting itself. To better understand this issue, we now examine popular policy hosting providers and how they implement policy delegation for domain owners.

Domain owners who delegate policy hosting must set up `CNAME` records pointing to their policy hosting provider. By analyzing these `CNAME` records for the MTA-STS policy label, we identify the providers responsible for hosting the policies. Table 2 lists the eight most popular policy hosting providers from our latest snapshot, along with their `CNAME` patterns and the number of domains using their services.

Despite these providers serving a significant customer base, we were surprised to find that some domains relying on them still experience issues (see §4.3.3); for example, some policy hosts returned expired certificates or empty policy files, causing errors. Interestingly, these issues affect only subsets of each provider's customers; for instance, only five domains using `DMARCReport` had empty policies.

Our qualitative analysis suggests that these cases often involve customers who discontinued the policy hosting service but left MTA-STS configurations pointing to the provider. To understand this phenomenon further, we contacted all providers via their support systems to inquire about their procedures for handling incomplete customer opt-outs. We observed four different approaches:

(1) **NXDOMAIN Response:** Three policy hosting services (`MailHardener`, `URIports`, and `PowerDMARC`) return `NXDOMAIN` responses to the canonical name a `CNAME` points to, which makes senders unable to resolve the policy domain.
(2) **Continued Certificate Issuance:** Four providers (`DMARCReport`, `EasyDmarc`, `Sendmarc`, and `OnDMARC`) continue issuing certificates for the `mta-sts` subdomain via ACME domain validation, even for inactive customers. However, `EasyDmarc`, `OnDMARC`, and `Sendmarc` do not update the policy file when a customer's `MX` records change. As a result, if the `mode` is set to `"enforce"` and the `mx` patterns are not updated to match new `MX` records, the domain may fail to receive emails from MTA-STS-compliant senders
(3) **Empty Policy File:** One operator, `DMARCReport`, changes the policy to an empty file. This will cause a parsing failure according to the MTA-STS standard, making senders treat this error equivalent to a `"none"` `mode`.
(4) **Mail Service Discontinued:** `Tutanota` rejects incoming emails if unsubscribed customers continue pointing their `MX` record to `Tutanota`, but the policy file remains unchanged. This can cause delivery issues if customers update their `MX` records without adjusting their `MTA-STS` settings. We received no response on whether `Tutanota` renews certificates for inactive customers. However, in our latest snapshot of 7,614 domains served by `Tutanota`, 10 domains with policy server errors still point their `MX` record and `mta-sts` IP to `Tutanota`, with 8 having expired SSL certificates.

In summary, none of the providers follow the best practices outlined in §2.6. Three providers return `NXDOMAIN` responses, effectively shutting down the policy file resolution, while

four others serve stale or invalid policies, risking email delivery failures for inactive customers.

These findings underscore the need for policy hosting providers to adopt a standardized and graceful deprovisioning process when customers opt out or become inactive.

6 SENDER-SIDE MTA-STS VALIDATION

While our main analysis has concentrated on how domains configure MTA-STS for incoming email, it is equally important to understand whether sending MTAs actually validate these policies when delivering outbound mail. In other words, even if a domain publishes a valid policy, its security benefit hinges on sender-side adoption and enforcement. For instance, if major providers neglect MTA-STS validation, recipients gain little protection from the protocol's deployment. Consequently, we complement our recipient-focused perspective by briefly exploring how senders implement MTA-STS, the potential obstacles they encounter, and how these practices affect the overall ecosystem.

6.1 Sender Side Dataset

To assess whether senders implement MTA-STS, we utilized an aggregate dataset shared with us by email-security-scans.org, which examines MTA sending behavior by recruiting participants to send emails to their platform [17]; this includes domains that implement MTA-STS with varying configurations and policies. The dataset spans over 3,806 individual deliverability tests spanning 2,394 unique sender domains. Data was collected between February 2023 and November 2024. For our analysis, we consider the most recent test per sender domain.

Limitations: It is worth noting that major mail operators heavily influence the data provided by email-security-scans.org; of 11,564 recorded `MX` interactions, 3,043 (26.31%) `ehlo` responses are attribute to outlook.com, and a further 2,663 (23.03%) are from google.com. In total, the top 10 operators account for 7,019 (60.7%) of all recorded interactions.

6.2 MTA-STS Validation

Our findings reveal that 2,264 (94.6%) domains support TLS when delivering email. The vast majority, 2,232 (93.2%) domains, perform opportunistic TLS (i.e., accepting any TLS certificate). Only 31 (1.3%) domains always require PKIX-valid certificates, regardless of MTA-STS or DANE being in use.

Regarding MTA-STS validation, we find that a total of 469 (19.6%) domains perform MTA-STS validation when sending to an MTA-STS-enabled domain; this is encouraging compared to the domains configured with `MTA-STS` records (i.e., approximately 0.1%) measured in §3. Interestingly, 714 (29.8%) sender domains perform DANE validation; of these, 203 (8.5%) domains validate both MTA-STS and DANE. However, 62 (2.6%) of them prefer MTA-STS over DANE[10], which is not recommended by RFC 8461 [26].[11]

7 SURVEY

To contextualize our network measurements and contrast our findings with operational practice, we conducted a survey in April 2024 on MTA-STS use among operators running email services. The survey responses provide a qualitative perspective on the real-world challenges and practices associated with MTA-STS deployment. The complete set of survey questions is available in Appendix C. A discussion of survey ethics is included in the ethics section in Appendix A.

7.1 Survey Methodology

Recruitment. We recruited survey participants from mailing lists, including the Mail Operators' List (MailOP) [1], the North American Network Operators' Group (NANOG) [31], and a mailinglist for Email Security Standards in the European Union (MESSEU) [30]; these mailing lists are populated by email administrators who demonstrate a high level of familiarity with email security protocols and standards. Out of 120 initial respondents, 117 engaged with at least one question in the survey.

Demographics. The participant demographics, as shown in Figure 11, cover a broad spectrum of mail-setup sizes. These range from 22 operators managing fewer than 10 accounts to 36 operators overseeing more than 500 accounts. This diversity indicates that our survey provides a comprehensive overview of the mail operator landscape.

Questionnaire Design. When designing the questionnaire, we avoided leading questions, and utilized a Likert scale for opinion questions. We generally followed survey methodology best practices as outlined by Redmiles et al. [37]. We provide the full questionnaire in §9.

Limitation. Although 117 survey responses yielded valuable qualitative insights, they cannot be taken as broadly representative. We recruited participants through mailing lists, which likely attracted more security-conscious email administrators than average, as noted in 7.2. Because the survey was unsupervised and had no pre-filtering, self-selection bias may be present, and common issues like self-reporting or social desirability biases could further influence results. Nevertheless, our goal was to *augment our technical observations* with qualitative perspectives; despite these limitations, the survey's findings offer useful context for understanding real-world email operations.

[10]This is done by presenting a PKIX-valid certificate while our `TLSA` records for DANE did not match the certificate.

[11]This is a known bug in a common Postfix MTA-STS milter [28].

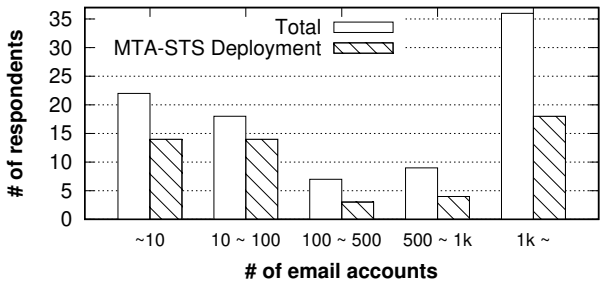

Figure 11: The distribution of the number of email accounts managed by each of the 92 respondents who answered the question regarding MTA-STS deployment.

7.2 Survey Findings

Deployment. Out of the 94 participants who responded to the question about their familiarity with MTA-STS, 89 (94.7%) had heard of the standard. Furthermore, of 88 respondents replying to the question whether they deployed MTA-STS for their primary domain, 50 (56.8%) confirmed its deployment. These high familiarity and deployment rates are likely related to our sample recruited from email expert groups, see §7.1.

Motivation for Deployment. The primary reason for deploying MTA-STS, identified by 34 out of 42 respondents (80.9%), was to prevent downgrade or interception attacks caused by STARTTLS stripping or DNS poisoning. Interestingly, 9 participants expressed greater trust in the web PKI than in DANE. Similarly, 10 participants favored MTA-STS over DANE, citing DANE's complexity in managing DNSSEC requirements; this complexity was also cited as a key motivation for the development of MTA-STS.

From a requirements standpoint, 13 out of 41 (31.7%) participants stated that customer demand drove their adoption of MTA-STS, while 14 (34.1%) participants indicated that regulatory compliance mandated its implementation. Furthermore, 5 participants responded that MTA-STS had a better reputation among large email providers, which contributed to their decision to adopt it.

Challenges in Deployment. Among the 43 respondents who had deployed MTA-STS, 21 (48.8%) identified operational complexity as the primary bottleneck. Additionally, 17 (39.5%) believed that DANE is fundamentally more secure, while 5 (11.6%) indicated that they do not require email encryption.

As for the 33 respondents who have not deployed MTA-STS, 15 (45.4%) stated they use DANE instead. Another 9 (27.2%) found MTA-STS too complicated to manage.[12]

Management. Of the 41 respondents who commented on the setup and management of MTA-STS, 8 (19.5%) found configuring the HTTPS policy file challenging, while 11 (26.8%) reported difficulties in managing policy updates. Moreover, regarding the policy update sequence, among the 42 respondents, 15 (35.7%) indicated that they never updated their policy. Notably, 10 (23.8%) responded updating the TXT record first, a practice not recommended due to potential temporary disruptions in mail delivery, as outlined in the standard [26].

MTA-STS vs. DANE. Out of 79 respondents to the related question, 78 (98.7%) answered that they are familiar with DANE. Among these, 26 (33.3%) stated that they do not serve any `TLSA` record. Additionally, 10 respondents responded that their DNS authoritative server and/or registrar lacked DNSSSEC support, which shows the challenge of DANE deployment.

While both MTA-STS and DANE are designed to enhance email security, a majority of administrators (51, 72.8%) argued that DANE is the superior solution. One administrator highlighted the potential drawbacks of MTA-STS, writing: "*I think MTA-STS is actively harmful in the sense that it undermines the incentive to deploy DNSSEC and DANE because it offers a false alternative. Note that the TOFU principle of MTA-STS is objectively less secure than DANE which does not rely on TOFU.*"

7.2.1 Survey Summary. While MTA-STS has limited general prominence, it is well known within sector-specific groups. Both MTA-STS and DANE are familiar to operators in our sample, with concerns focused on their respective complexities. For DANE, the primary challenge is its reliance on DNSSEC, despite its superior security. Although MTA-STS was envisioned as an alternative in environments where DNSSEC is impractical, operators cited other reasons for its adoption, including confidence in the established web PKI ecosystem and support from major providers such as Google. At the same time, they identified policy server setup and ongoing maintenance as persistent challenges, echoing issues we highlighted in §4.

8 RELATED WORK

SMTP Encryption: Several studies [13, 16, 21, 41] focused on `STARTTLS` deployment and have reported its widespread adoption. However, Poddebniak et al. [32] demonstrated that `STARTTLS` is vulnerable to attacks such as stripping, command injection, and mailbox spoofing. To mitigate these vulnerabilities, DANE and MTA-STS are being used. Lee et al. [23, 24] showed a rise in DANE deployment in 2020 and

[12]One administrator wrote "*We now have to manage a webserver cluster for redudency [sic], a new website, and dns records, and sync them all, [...] , since mta-sts is a file on a webserver and not in DNS, the list of *valid* entries could be inaccurate.*"

2022, albeit with numerous misconfigured domains, particularly among self-managed SMTP servers; They identified DNSSEC dependency and key rollover challenges as the primary barriers to DANE adoption. *On the contrary, in our study, we show that maintaining a separate HTTP server for policy hosting and synchronizing the policy with* `MX` *records are the major challenges in MTA-STS adoption.*

MTA-STS Ecosystem: Due to its early stage of development, only a few studies have analyzed the adoption of MTA-STS. In 2019, Tatang et al. [39] reported only 221 domains with an `MTA-STS` record out of 1.7M domains. In 2022, Holzbauer et al. [17] developed a crowdsourced platform to measure email delivery, DNSSEC validation, and TLS configuration, initially excluding MTA-STS. Later, they extended it to measure MTA-STS validation by senders, which we leveraged in this paper (§6.1). In 2023, Blechschmidt et al. [6] analyzed inbound and outbound MTA-STS support on a smaller sample compared to our study; among the DomCop top 10M domains, they identified 6.9K domains with an MTA-STS record, but only 569 had a policy file. They examined 47 email providers and found only six fetched the MTA-STS record, with four attempting validation.

While previous studies explored sender-side MTA-STS support and issues, *none of them examined recipient-side misconfigurations in depth.* Our study investigates `why` such misconfigurations occur across managing entities and validates findings through surveys with policy providers and email administrators. Additionally, our larger sender-side validation dataset offers a broader view of MTA-STS validation in the wild.

9 CONCLUSION

In this paper, we conducted a comprehensive, longitudinal study of the MTA-STS ecosystem, examining its deployment, management, and practical challenges. We found that MTA-STS deployment is gradually rising but numerous misconfigurations and inconsistencies exist in MTA-STS setups. Improperly configured policy servers are the primary culprit, affecting 17,184 (85%) of the misconfigured domains. Even when a domain owner outsources both email and policy service to third-party entities, misconfigurations are prevalent due to a lack of synchronization.

We also conducted a survey among email operators to understand the MTA-STS ecosystem in practice. While awareness of MTA-STS was high (94.7%), many cited operational complexity (48.8%) as an underlying factor for low adoption rate. Respondents also reported difficulties in managing policy updates (26.8%) and maintaining policy servers for multiple domains.

ACKNOWLEDGMENTS

We thank anonymous reviewers and our shepherd, Oliver Hohlfeld for their helpful comments. This research was supported in part by NSF grant CNS-2339378, and Commonwealth Cyber Initiative.

REFERENCES

[1] Mail Operators' List. https://www.mailop.org/.
[2] D. E. 3rd. Transport Layer Security (TLS) Extensions: Extension Definitions. RFC 6066, IETF, 2011.
[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, IETF, 2005. http://www.ietf.org/rfc/rfc4033.txt.
[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, IETF, 2005. http://www.ietf.org/rfc/rfc4035.txt.
[5] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, IETF, 2005. http://www.ietf.org/rfc/rfc4034.txt.
[6] B. Blechschmidt and B. Stock. Extended Hell(o): A Comprehensive Large-Scale Study on Email Confidentiality and Integrity Mechanisms in the Wild. *USENIX Security*, 2023.
[7] R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten. Automatic Certificate Management Environment (ACME). RFC 8555, IETF, 2019.
[8] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, IETF, 2008. http://www.ietf.org/rfc/rfc5280.txt.
[9] J. Chen, V. Paxson, and J. Jiang. Composition kills: a case study of email sender authentication. *USENIX Security*, 2020.
[10] 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.
[11] C. Dietrich, K. Krombholz, K. Borgolte, and T. Fiebig. Investigating system operators' perspective on security misconfigurations. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 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] DNSSEC deployment in Sweden. https://archive.icann.org/meetings/london2014/en/schedule/wed-dnssec/presentation-dnssec-deployment-sweden-25jun14-en.pdf.
[15] Enhancing mail flow with MTA-STS. https://learn.microsoft.com/en-us/purview/enhancing-mail-flow-with-mta-sts.
[16] 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.
[17] F. Holzbauer, J. Ullrich, M. Lindorfer, and T. Fiebig. Not that Simple: Email Delivery in the 21st Century. *USENIX ATC*, 2022.
[18] G. Huston. Measuring the use of DNSSEC. 2023. https://blog.apnic.net/2023/09/18/measuring-the-use-of-dnssec/.

[19] H. Hu and G. Wang. End-to-End Measurements of Email Spoofing Attacks. *USENIX Security*, 2018.
[20] P. Hoffman. SMTP Service Extension for Secure SMTP over Transport Layer Security. IETF RFC 3207, IEFT, 2002.
[21] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar. TLS in the wild: an Internet-wide analysis of TLS-based protocols for electronic communication. *NDSS*, 2015.
[22] Increase email security with MTA-STS and TLS reporting About MTA-STS and TLS reporting. https://support.google.com/a/answer/9261504?hl=en.
[23] 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.
[24] 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.
[25] Levenshtein distance. https://xlinux.nist.gov/dads/HTML/Levenshtein.html.
[26] D. Margolis, M. Risher, B. Ramakrishnan, A. Brotman, and a. J. Jones. SMTP MTA Strict Transport Security (MTA-STS). RFC 8461, IETF, 2018.
[27] D. Margolis, A. Brotman, B. Ramakrishnan, J. Jones, and M. Risher. SMTP TLS Reporting. RFC 8460, 8460, RFC Editor, 2018.
[28] MTA-STS Overrides DANE. https://github.com/Snawoot/postfix-mta-sts-resolver/issues/67.
[29] Mail.com MTA-STS Policy. https://mta-sts.mail.com/.well-known/mta-sts.txt.
[30] Modern Email Security Standards for EU (MESSEU). messeu@sys4.de.
[31] North American Network Operators' Group. https://www.nanog.org/.
[32] D. Poddebniak, F. Ising, H. Böck, and S. Schinzel. Why TLS is better without STARTTLS: A Security Analysis of STARTTLS in the Email Context. *USENIX Security*, 2021.
[33] 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.
[34] Porkbun LLC. https://porkbun.com/.
[35] Postfix. http://www.postfix.org/.
[36] Potential Risks with MTA-STS Usage? https://support.dmarcreport.com/support/solutions/articles/5000885320-potential-risks-with-mta-sts-usage-.
[37] E. Redmiles, Y. Acar, S. Fahl, and M. Mazurek. A summary of survey methodology best practices for security and privacy researchers. 2017. https://drum.lib.umd.edu/items/683d78b0-a0e3-4fae-9c93-b75aae4ad11b.
[38] B. Stock, G. Pellegrino, F. Li, M. Backes, and C. Rossow. Didn't You Hear Me? — Towards More Successful Web Vulnerability Notifications. *NDSS*, 2018.
[39] D. Tatang, F. Zettl, and T. Holz. A First Large-Scale Analysis on Usage of MTA-STS. *DIMVA*, 2021.
[40] TLD Distribution. https://domainnamestat.com/statistics/tld/others.
[41] The current state of SMTP STARTTLS deployment. https://www.facebook.com/notes/protect-the-graph/the-current-state-of-smtp-starttls-deployment/1453015901605223/.
[42] What is greylisting and how does it work? https://www.mail.com/blog/posts/what-is-greylisting/33/.
[43] Yahoo! Mail MTA-STS. https://mta-sts.yahoo.com/.well-known/mta-sts.txt.

APPENDIX

A ETHICS

A.1 Data Collection

For this study, we scanned DNS records, MTA-STS policy files, and STARTTLS certificate for `MX` hosts. All of these scans were done at low rate limits to ensure that we do not adversely affect any users' ability to obtain these data in a timely manner. For DNS scans, we opted to use public resolvers to avoid the risk of overloading small recursive resolvers. None of the measurement data we obtained throughout our study involved human subjects or any personally identifiable information.

A.2 Disclosure Emails

We initially did not notify misconfigured domain owners, as such notifications can result in administrators becoming overwhelmed with alerts, often treating them as spam [38]. However, based on feedback from the scientific community, we decided to notify the affected domains via email. We also included a simple feedback mechanism in our email to understand whether our initial conjecture is valid or not; Of 497 feedback responses, 341 considered our message to be helpful. We also received 45 acknowledgments thanking us for the notification.

A.3 Survey

Our study centers on organizations rather than individuals, gathering data on system deployments. The Institutional Review Board (IRB) has reviewed our methodology and determined that it does not constitute human subject research, thereby exempting it from the protocols typically required for such studies. Despite this exemption, we maintained a commitment to best practices comparable to those used in human subject research, ensuring that participants were informed about their rights regarding data access and the option to withdraw at any time.

B SMTP TLS REPORTING

SMTP TLS reporting[27] is a mechanism that allows the senders to inform the recipient MTA of any problems with its TLS negotiations or MTA-STS/DANE policy validation. This enables administrators to address misconfigurations and potential security vulnerabilities. When a sending MTA encounters issues related to Transport Layer Security (TLS) while attempting to deliver emails to a domain that has a TLS Reporting (TLSRPT) policy, it compiles a report detailing these issues and sends it to the designated reporting address specified by the receiving domain. A domain's TLSRPT policy

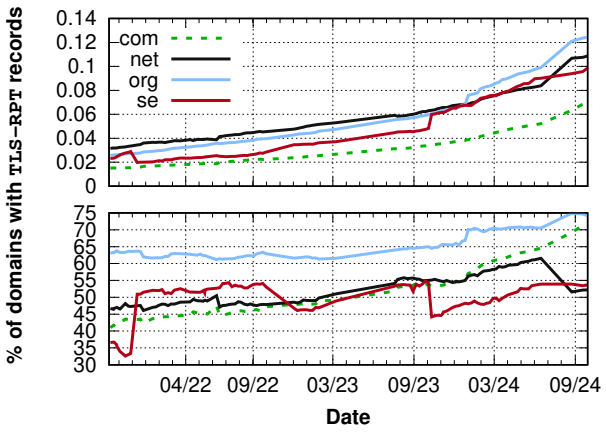

Figure 12: The deployment of TLSRPT records; top graph shows the % of domains with MX records having TLSRPT records; bottom graph shows the % of domains with MTA-STS records having TLSRPT records. At December 21, 82 .se domains revoked their TLSRPT records causing the downspike in the top graph. Additionally, from Jun' 24 to Aug '24, 1,411 .net domains added TLSRPT records; only 198 of them had MTA-STS records which explains the downward spike around that time in the bottom graph.

is found at the TXT record of _smtp._tls subdomain which is referred to as TLSRPT record. Along with MTA-STS, MX, NS, and A, we also collected TLSRPT record of all the SLDs in our dataset from §3.

While unfortunately not many domains currently support report sending[13] (only 20 in §6.1), number of domains that support TLSRPT records are far greater as shown in Figure 12; this implies domain owners are open to receive TLS reports to debug their potential misconfigurations. Initial adoption in 2021-10 ranged from 11,531 (0.02%) for .com to 1,527 (0.03%) domains for .org. As of 2024-09, we find adoption to have risen 3-4 times, with adoption ranging between 52,641 (0.07%) domains for .com and 7,192 (0.12%) domains for .org.

Although the adoption rate is still relatively low, we can see a high percentage of domains supporting MTA-STS has SMTP TLS reporting enabled. In our disclosure emails (§4.7), we recommended domain owners to adopt TLS Reporting if they have not already enabled it.

C SURVEY QUESTIONNAIRE

All the questions except in Page 1 are optional. In questions where we had Other (please specify) as the last option, a textbox was there for the participants to specify their answer. SCQ denotes single choice question, MCQ denotes multiple choice question, and YN denotes yes no question. TB denotes open-ended response questions with a textbox. GS denotes grid-style questions with a matrix where respondents rate multiple items or statements (rows) against a set of consistent response options (columns). LS denotes Likert scale questions.

Page 1: Consent Form. Participants are presented with the following two mandatory consent questions:

- I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.
- I understand that information I provide will be used for scientific reports and publications.

If the participant answered No to any of the above questions, the survey would end with no further input.

Page 2: Basic Info.

TB Would you be willing to provide the name of the organization whose e-mail service you manage? (If you don't want to, please enter No.)

TB Would you mind sharing the name of the domain whose e-mail service you manage? We'd like the main domain name, not the domain name of your mail server. For example, example.com, not mx.example.com. (If you don't want to give the name, please enter No). If you enter the domain name, we may fetch MX/NS/TLSA/MTA-STS records.

SCQ How many email accounts exist under your operated infrastructure? Options are < 10, 10 50, 50 100, 100 500, > 500.

Page 3: MTA-STS check 1.

YN Have you heard about MTA-STS (SMTP MTA Strict Transport Security)?

If the participant answered No to this question, the survey would end with no further input.

Page 4: MTA-STS check 2.

YN Does your domain support MTA-STS? In other words, do you have MTA-STS TXT record in your domain? If you have multiple domains under your infrastructure, please fill this for your most used domain.

If the participant answered No to this question, the survey would jump to Page 10.

Page 5: Deployment for inbound emails.

GS Select the best option for each of the following statements for your most used domain.
Statements are: 1. Have a valid MTA-STS TXT record, 2. Have a valid MTA-STS policy with proper HTTPS setup, 3. DNS MX records are consistent with mx values in HTTPS policy body, 4. Email server (i.e. MX record of the domain) supports STARTTLS, 5. Have a

[13] Only 2 major providers send TLS reports currently: Google and Microsoft

PKIX valid certificate for some of my SMTP servers, 6. Have a PKIX valid certificate for all my SMTP servers. Column options are: `Yes`, `No`, `Not sure`.

LS Why did you choose to adopt MTA-STS for your domain?
Statements are: 1. Prevents downgrade or interception attack by STARTTLS stripping or DNS poisoning, 2. Dependency on web PKI sounds more trustworthy, 3. Provides optional testing only mode, 4. Other alternative DANE requires DNSSEC and is harder to manage. Columns are scaled from `most important` to `not important`. Here the participants also had the option to provide open-ended comments in a textbox.

LS Why do you think operators roll out MTA-STS?
Statements are: 1. Because customers asked us to, 2. Because we are required by regulation (e.g. DNSSEC regulation is the US/NL), 3. Because we wanted to play with it, 4. Because we believe it will make google accept our mails more, 5. Because we always want to be on the pulse of tech-dev.
Columns are scaled from `most important` to `not important`.

LS What is the largest bottleneck for MTA-STS deployment in your opinion?
Listed assumptions are: 1. Operational complexity, 2. Better alternative in DANE, 3. Do not need email encryption.
Columns are scaled from `most important` to `not important`.

Page 6: Misconfigurations.

SCQ Is the MTA-STS setting of your domain valid? Options are `yes`, `no`, and `don't know`.

LS What is the most difficult thing you found in setting up and managing MTA-STS for your domain?
Statements are: 1. Setting up associated DNS records, 2. Configuring HTTPS policy file, 3. Configuring SMTP server with a PKI valid certificate, 4. Managing policy update, 5. Opting out of MTA-STS.
Columns are scaled from `most difficult` to `not difficult`. Participants also had the option to provide open-ended comments in a textbox.

LS What do you think is the main reason behind the prevalent invalid MTA-STS configurations?
Row options are: 1. Dependency error between policy and DNS (e.g. mismatch between `mx` pattern in policy file and DNS `MX` record), 2. SMTP server error (e.g. lacking PKIX-valid certificate), 3. HTTPS policy server error (e.g. TLS certificate failure), 4. DNS error (e.g. wrongly configured DNS setting).
Columns are scaled from `matters most` to `does not matter`.

SCQ While updating your policy, which sequence do you maintain? Options are: 1. Update MTA-STS TXT record first, and then update HTTPS policy body, 2. Update HTTPS policy body first, and then update MTA-STS TXT record, 3. Never updated, Don't Know (my policy management is automated/ outsourced/ i am not sure about the order).

Page 7: Policy Host Management.

SCQ How do you manage your MTA-STS policy host? Options are: 1. outsourced to a 3rd-party policy hosting provider, and 2. self-managed.

If the participant selected `self-managed` in this question, the survey would jump to Page 11.

Page 8: Management 1.

SCQ Which 3rd-party policy host service do you use? Options are: Tutanota, URIPorts, Mailhardener, PowerDMARC, EasyDMARC, OnDMARC, DMARCReport, Other (please specify).

LS To what extent do you agree with the following statements regarding hosted MTA-STS services?
Statements are: 1. Using hosted MTA-STS service reduces operational complexity to manage MTA-STS policy, 2. Using hosted MTA-STS service reduces error rate due to misconfigurations in HTTPS policy.
Columns were scaled from `strongly agree` to `strongly disagree`.

SCQ How do you manage your incoming SMTP server? Options are: 1. outsourced to a external email hosting provider, and 2. self-managed.

If the participant selected `self-managed` in this question, the survey would jump to Page 11.

Page 9: Both outsourced.

YN Does your email hosting provider manage your MTA-STS policy?

Page 10: MTA-STS not supported.

SCQ Why do you NOT deploy MTA-STS for your domain? (Please add options if there are any other reasons.) Options are: 1. I do not understand how it works, 2. I understand how it works, but I don't think I need it, 3. I understand how it works but it is too complicated to deploy and manage, 4. I use DANE, 5. Other (Please specify).

YN Have you ever used MTA-STS?

Page 11: DANE check 1.

YN Have you heard about DANE (DNS-based Authentication of Named Entities)?

If the participant selected `No` in this question, the survey would jump to Page 13.

Page 12: Comparison w/ DANE.

GS Does your email server support DANE for inbound emails?
Statements are: 1. My mail server domain (i.e. MX address of the domain) has a valid TLSA record, 2. My mail server supports STARTTLS, 3. My DNS authoritative server and registrar both have support for DNSSEC, 4. My SMTP server has a TLS certificate and TLSA record is consistent with the SMTP server certificate.
Column options are: `Yes`, `No`, `Don't know`.

LS In your opinion, which protocol is better in design for mandating email encryption? MTA-STS or DANE?
Options are: 1. Which protocol is easier to deploy, 2. Which protocol has less requirements, 3. Which protocol is easier to maintain/keep functional, 4. Which protocol has the higher security benefit, 5. Which protocol has the higher general benefit, 6. Which protocol incurs lower total cost.
Column scales are: 1. Definitely MTA-STS, 2. More MTA-STS, 3. Balanced, 4. More DANE, 5. Definitely DANE.

TB Are there any other implementation considerations around MTA-STS and DANE that you would like to share?

Page 13: MTA-STS check 3.

YS Does your email server(s) validate MTA-STS for outbound connections? Options are: `Yes`, `No`, `Don't Know`.

If the participant selected `No` in this question, the survey would end with no further input.

Page 14: Validation.

SCQ Which tool do you use to validate MTA-STS for outbound connections? Options are: postfix-mta-sts-resolver, mox, proprietary, other (please specify).

Page 15: Validation.

LS What do you think is the major bottleneck behind lack of MTA-STS validation support?
Row options are: 1. Lack of incentive from the sending side, 2. Difficulty in policy cache maintenance, 3. Low deployment rate among domains, 4. Lack of awareness of its benefits. Column options scaled from `most important` to `not important`.