All of Government

Secure Government Email MTA-STS Deployment Guide

Issued by

Digital Services branch

Internal Affairs
Te Tari Taiwhenua

New Zealand Government

Contents

1 Document Control

Service Name	Secure Government Email
Author	Matt Oliver, Operations Security Specialist, Department of Internal Affairs
Title	Secure Government Email MTA-STS Deployment Guide
Date and Version	5 November 2025, v1.1
Index Number	AoGSD.SGE.Guidance.2025_256

2 Introduction

The Secure Government Email (SGE) Common Implementation Framework was confirmed late in 2024 with a goal to improve email security standards across the New Zealand Government. This deployment guide compliments the Framework through providing examples on how to deploy Mail Transfer Agent Strict Transport Security (MTA-STS) in line with both the framework and to meet the requirements of the New Zealand Information Security Manual (NZISM).
It provides configuration examples covering commonly used platforms and configurations but is not an exhaustive guide. It is recommended Agencies without experience in the deployment of email security tools engage with a provider with expertise in this area.
Services are available for Agencies via the Secure Email Management and Administration service in Marketplace Marketplace | Pae Hokohoko — Infrastructure Managed Services catalogue.

3 NZISM and SGE Framework requirements for MTA-STS

Chapter 15.2 (Email Infrastructure) of the NZISM directs that Agencies SHOULD enable MTA-STS to prevent the unencrypted transfer of emails between complying servers, and when doing so that a minimum of TLS1.2* must be used.

The SGE Framework maps back to the NZISM but extends it further requiring “MTA-STS needs to be enabled and set to ‘enforce’ on all email enabled domains.”

*TLS version enforcement is not part MTA-STS. Refer to the Implicit TLS Configuration Guide for details on TLS1.2 or greater enforcement.

4 MTA-STS Configuration

Enabling MTA-STS in enforce mode can have an impact on incoming email to your organisation. This is described in more detail in Appendix 1 – MTA-STS Description.

TLS Reporting needs to be in place prior to configuring MTA-STS, otherwise you may not be able to see any TLS failures. Refer to the TLS Reporting Configuration guide for details.

MTA-STS requires the configuration of a DNS TXT record and a policy file. The configuration steps are:

1. Create the MTA-STS DNS TXT record.
2. Create and host your MTA-STS policy file in testing mode.
3. Test your record.
4. Monitor MTA-STS though TLS Reporting.
5. Edit your policy file to 'enforce.'
6. Continue to monitor for failures.

It is assumed you have the required access and knowledge to configure the required services.

The following is based on using the GovtDNS portal. It will be similar for other DNS platforms.

Step 1: Create the MTA-STS DNS TXT record.

1. Log in to the DNS portal and browse to your domain.
2. Click on the Current tab under DNS Zones.
3. Click on Edit.
4. Click on the + symbol to add a new row.
5. Set the name to _mta-sts
6. Set the TTL to 86400 (24 Hours, recommended).
7. Change the type to TXT.
8. Set the data to `v=STSv1; id=<timestamp>` EG: 20251027T000000

   Note: The ID field can be anything you like. It is only used to identify when changes have been made to the policy file so they take effect prior to the max age in the policy file. Using a timestamp as in the example above is one approach.

   The v= (version number) must be `v=STSv1` exactly and is followed by a semi-colon before the id.
9. Add a comment or change number if needed for your environment.
10. Click on Publish.
11. Repeat this for other domains if required.

    Any sending server which supports MTA-STS will already be looking for the existence of this record. Now it resolves, the sending server will attempt to connect to a hosted file at `https://mta-sts.<yourdomain>/.well-known/mta-sts.txt`

    The fact this file may not yet exist will not impact mail flow.

Step 2: Create and host your MTA-STS policy file in test mode.

1. Create a new text file called mta-sts.txt. This file needs to contain the MTA-STS version, mode, MX destinations and maximum age:

   ```
   version: STSv1
   mode: testing
   mx: minties-govt-nz.mail.protection.outlook.com
   mx: mx1.minties.govt.nz
   mx: mx2.minties.govt.nz
   max_age: 604800
   ```

   The version is always STSv1. The mode should be set to testing. The MX entries must match all MX servers configured for your domain, including any alternate servers. The minties.govt.nz domain used is primarily hosted on M365, making the syntax `<domain>.mail.protection.outlook.com`. It also has two alternate MX servers. The max_age is the maximum allowable time, in seconds, the record can be cached for, before a future DNS lookup should check for changes. 604800 seconds equals 7 days, and is a good default entry to work to.

   Note: You must not use * wildcards in the mx records in your mta-sts.txt file.

2. This file must be named mta-sts.txt and hosted using https at `https://mta-sts.<yourdomain>/.well-known/mta-sts.txt`. It must be stored in the /.well-known/ directory. The TLS certificate for the HTTPS endpoint hosting your MTA-STS policy must match the domain in the policy URL. For example, if the policy is served at https://mta-sts.minties.govt.nz/.well-known/mta-sts.txt, the certificate must include either minties.govt.nz or a wildcard entry such as *.minties.govt.nz in its CN or SAN to ensure secure policy retrieval. The site location must have a valid TLS certificate issued by a public Certificate Authority (CA) trusted by major mail providers.

   The mta-sts.txt file should be accessible in a highly available location.

Step 3: Test your MTA-STS record and file are operational.

Once DNS records are published, and you have the file hosted in the right location, use an external service to check the configuration is valid. Using an MTA-STS record check on the MXToolbox SuperTool provides a useful test. (https://mxtoolbox.com/SuperTool.aspx#)

Enter your domain name and select the MTA-STS Record Lookup. This will return the DNS Record, followed by the mta-sts.txt file. It will confirm the MX record match and finally will show the associated CA certificates. If there are any issues they will be highlighted.

Your MTA-STS record should now be operational in testing mode. At this stage it is not impacting any mail flows but will be producing reports via your TLS Reporting service.

Step 4: Monitor MTA-STS through TLS Reporting.

Monitor MTA-STS through TLS Reporting. Refer to the TLS Reporting configuration guide for more details. You are primarily looking for:

- Frequent failures from specific senders
- STARTTLS not offered
- Certificate issues
- Downgrade attack indicators

This helps you assess whether enforcing MTA-STS would block legitimate mail. Adding MTA-STS in testing mode on top of TLS-RPT provides you with richer metadata in TLS reports.

It will not show what version of TLS is negotiated on connections, however versions 1.0 and 1.1 are effectively deprecated through RFC 8996, so will not be used by most senders already.

Identify and resolve any issues with your mail server configuration, such as:

- TLS certificate mismatches
- DNS misconfigurations
- Policy file errors

Ensure that legitimate email traffic is not being disrupted before enforcing strict security policies.

Step 5: Edit your policy file to 'enforce.'

Once you are satisfied any TLS issues are resolved, edit your hosted mta-sts.txt file; change the mode from testing to enforce. Save the file and test it externally through browsing to it, then update the ID number field in your DNS TXT record.

One option prior to moving from testing to enforce mode is to reduce both the max_age and TXT record TTL to 3600 (1 hour). This is a defensive approach allowing faster recovery should you need to roll back the change.

Step 6: Continue to monitor for any issues.

As per step 4, continue to monitor your TLS Reports for any issues with MTA-STS.

Appendix 1 – MTA-STS Description

What is TLS-RPT?

MTA-STS stands for Mail Transfer Agent Strict Transport Security. It's a security protocol designed to improve the security of email delivery over the internet through enforcing the use of TLS (Transport Layer Security) when sending emails between mail servers.

When MTA-STS is configured in enforce mode, and the sending server supports TLS, MTA-STS enforces the use of TLS encryption. If TLS fails to negotiate, the session will not connect and will not revert to sending in the clear. In that case the sending server will queue the email till such time as the error is resolved or it times out and the message fails.

If the sending MX service does not support TLS it will not be impacted by MTA-STS as it will never perform a lookup. In that case the sending service will attempt to send in the clear, however it is important to note, the SGE Framework also requires the use of implicit TLS meaning sending in the clear is not permitted. This is covered in the Implicit TLS deployment guide. For the purposes of this guide, it is not MTA-STS which will block the transmission of emails from MX services which do not support TLS.

How it works:

When one mail server sends an email to another, it typically uses SMTP. SMTP can use TLS to encrypt the connection, but this is opportunistic, meaning if TLS isn't available or fails, the email may be sent in plain text. This may open the door to man-in-the-middle attacks.

MTA-STS addresses this by allowing domain owners to publish a policy that:

1. Requires TLS for incoming email.
2. Specifies the correct MX (Mail Exchange) servers to prevent impersonation.
3. Is delivered securely via HTTPS, making it harder for attackers to tamper with.

The Mail Transfer Agent on the sending server:

1. Performs a DNS lookup for a TXT record at _mta-sts.<domain> on the destination domain. If the policy exists and is valid it indicates the recipient domain uses MTA-STS and it records the policy ID for caching purposes.
2. Performs an HTTPS fetch of the recipient domains policy file from: `https://mta-sts.<domain>/.well-known/mta-sts.txt`. This site location must have a valid TLS certificate issued by a public Certificate Authority (CA) trusted by major mail providers. The certificate must match the MX hostname listed in your MTA-STS policy file. The match can be either as a CN or SAN and wildcards are acceptable.
3. Checks the MX records in DNS match those listed in the policy.
4. Checks the enforcement level in the policy (none, testing, enforce).
5. Checks the HTTPS certificate for the destination is valid.
6. Records the max_age for the caching timeout.
7. Attempts to establish a STARTTLS connection to the destination, verifying the MX servers TLS certificate, which must be both valid and signed by trusted CA.
8. Finally, if all checks are passed it will proceed and deliver the email securely over TLS. If any checks fail and enforce mode is in place delivery is aborted.

Key components of MTA-STS:

The key components of MTA-STS are:

- Policy file: Hosted at https://mta-sts.<yourdomain>/.well-known/mta-sts.txt, it defines the domain's mail security requirements.
- DNS record: A TXT record at _mta-sts.<yourdomain> signals the domain supports MTA-STS and provides the policy version. This can be achieved through using a CNAME record to point the DNS lookup to an alternate location, such as to a managed DMARC service provider who may manage and host the MTA-STS file on your behalf.
- Enforcement mode: Can be none, testing, or enforce. Only enforce mode mandates TLS and validates MX records. Using none is of no value, it is more sensible to go straight from not having any record to testing. Testing mode will not impact mail flows.

Benefits of MTA-STS in enforce mode:

Using MTA-STS in enforce mode provides several critical security benefits for email communications, especially for organizations that prioritize confidentiality and integrity:

Guaranteed TLS Encryption for SMTP

- Enforce mode ensures that email is only delivered if a secure TLS connection can be established. (Refer to the FAQ page for what happens when servers do not support TLS)
- Prevents downgrade attacks, where an attacker forces the connection to fall back to plain text SMTP.

Protection Against MX Record Spoofing

- The MTA-STS policy includes a list of authorized MX hosts.
- Sending servers validate that the destination MX matches the policy, preventing attackers from redirecting mail to rogue servers via DNS spoofing or cache poisoning.

Reduced Risk of Adversary-in-the-Middle (AitM) Attacks

- Opportunistic TLS is vulnerable to AitM attacks during the STARTTLS negotiation.
- MTA-STS in enforce mode requires a valid certificate and a secure connection, making AitM attacks significantly harder.

Improved Email Delivery Assurance

- While enforce mode may cause delivery failures if TLS can't be established, it ensures that emails are only delivered securely.
- This is especially important for sensitive communications (e.g., legal, financial, healthcare).

Visibility and Monitoring

- Domains can start with testing mode to monitor how many sending servers respect MTA-STS.
- Once confident, switching to enforce mode provides strong guarantees about secure delivery.

Complementary to Other Email Security Protocols

- MTA-STS works alongside SPF, DKIM, and DMARC, which protect against spoofing and phishing.
- It adds a layer of transport security, which those protocols do not address.

Trust and Reputation

- Domains that implement MTA-STS in enforce mode signal a strong security posture.
- This can improve trust with partners and customers, especially in regulated industries.

Appendix 2 – Frequently Asked Questions

What happens if the sending server does not support MTA-STS?

If the sending server does not support MTA-STS it almost certainly also does not support TLS. It will attempt to send in the clear. Under the SGE Framework such a connection will be denied because of the Implicit TLS requirement. The connection will be denied, but not because of MTA-STS.

Why is MTA-STS important?

- It protects against TLS downgrade attacks, where the threat actor attempts to force a connection to drop back to sending in the clear so they can capture the data unencrypted.
- It protects against MX record spoofing which could redirect email to malicious servers.
- It protects against adversary-in-the-middle attacks during email transmission.

What happens if a sending server can't establish a secure connection in enforce mode?

The email will not be delivered. It will be queued on the sending server until a secure connection can be established or until it times out based on the sending servers policy. This prevents the email from being sent over an insecure connection.

How does the caching work?

When the sending server performs its DNS lookup and HTTPS fetch it records the ID number and max_age. From then on it will only do a DNS lookup to check the ID number. If that number has not changed it will not do another HTTPS fetch till the max_age timer has expired.

In practical terms with a max_age of 86400 (7 days) this means if you change the policy but do not change the ID number it will take up to 7 days for your policy change to take effect. To have the changes apply faster, update the ID number.

What are the risks of using enforce mode?

Enabling enforce mode without adequate testing may result in email delivery failures. As you are the recipient these may not be immediately visible. The deployment methodology in this document recommends you start off in testing mode, then work through any issues prior to moving to enforce mode.

Delivery failures can occur if:

- TLS is misconfigured
- MX records do not match the policy
- The HTTPS policy file is unreachable.

Mitigating these risks is achieved through testing and monitoring before switching to enforce mode.

Can I use DANE instead of MTA-STS?

The use of DANE is not recommended at this stage.

DANE (DNS-based Authentication of Named Entities) and MTA-STS aim to solve similar problems in email security, but they use very different mechanisms.

Dane requires DNSSEC to operate. The GovtDNS platform is only able to provide DNSSEC to domains hosted on the GovtDNS infrastructure which is why MTA-STS has been chosen as the preferred secure delivery method for email across the New Zealand Government.

MTA-STS is very widely supported including by all major email providers. At the time of writing Google do not support DANE.

Where should I host my mta-sts policy file?

The mta-sts.txt file should be hosted in a high security, highly available location. If the file becomes unreachable sending servers will not know what policy is in place. In this case the sending server may revert back to opportunistic TLS potentially opening an attack vector.

The hosting location could be a public cloud service, managed service provider, or an Agency web server. The key is ensuring it is secure, and highly available.

If the file were to be edited by a threat actor, they could manipulate how sending servers interact with your domain, potentially leading to email delivery disruption, interception, or denial of service.

Can I use a CNAME to direct my DNS queries to a managed service provider?

Yes, you can use a CNAME record on your domain to point mta-sts queries to an alternate host location. This is useful if you wish to use a managed service provider for hosting your mta-sts record. An example of this is given in Appendix 3.

Does MTA-STS impact outbound email?

No. MTA-STS is only relevant to inbound email to your domain. Outbound email is subject to the recipient domains MTA-STS policy if they have one.

Appendix 3 – Configuration example

The following configuration example is of a lab set up with MTA-STS in enforce mode. For this example the domain minties.govt.nz is used, with its DNS is hosted on Cloudflare, and the policy file hosted on a GitHub repository. This is not provided as a 'recommended' solution, but to highlight this possible configuration. There are several possible configuration and hosting possibilities, we are not recommending any specific solution.

The domain is minties.govt.nz. The GovtDNS platform is configured to send all DNS requests to Cloudflare which is being used as the DNS host for the domain. Cloudflare has a CNAME record pointing mta-sts.minties.govt.nz to a GitHub instance where the policy file is hosted.
GitHub automatically hosts the CA certificate for the domain.

Cloudflare DNS configuration:

MTA-STS Record:	_mta-sts.minties.govt.nz
Record Type:	TXT
Name:	_mta-sts
Content:	"v=STSv1; id=20250725T000000"
TTL:	auto

MTA-STS Policy file:	mta-sts.minties.govt.nz
Record Type:	CNAME
Name:	mta-sts
Content:	minties-govt-nz.github.io
TTL:	auto

GitHub Configuration on minties-govt-nz.github.io:

CNAME Record:	mta-sts.minties.govt.nz
Folder location:	.well-known
File name:	mta-sts.txt
File content:	version: STSv1 mode: enforce mx: minties-govt-nz.mail.protection.outlook.com mx: mx1.minties.govt.nz mx: mx2.minties.govt.nz max_age: 604800
_config.yml entry:	include: [".well-known"]

Note: the _config.yml entry as required on GitHub as their backend software is a package called Jekyll, which treats folders dot-prefixed folders as hidden. This entry overrides that block.

The policy file mta-sts is stored in the .well-known folder.

Appendix 4 – List of Acronyms

Acronym	Definition
AitM	Adversary in the Middle
CA	Certificate Authority
CN	Common Name
DANE	DNS-based Authentication of Named Entities
DKIM	Domain Keys Identified Mail
DMARC	Domain-based Message Authentication, Reporting, and Conformance
DNS	Domain Name System
JSON	JavaScript Object Notation
M365	Microsoft 365
MTA-STS	Mail Transfer Agent Strict Transport Security
MX	Mail Exchange
NZISM	New Zealand Information Services Manual
RUA	Reporting URI for Aggregate data
SAN	Subject Alternate Name
SGE	Secure Government Email (Framework)
SMTP	Simple Mail Transfer Protocol
SPF	Sender Policy Framework
STARTTLS	Start Transport Layer Security
TLS	Transport Layer Security
TLS-RPT	Transport Layer Security Reporting
TTL	Time To Live