NO STARTTLS

Why TLS is better without STARTTLS

A Security Analysis of STARTTLS in the Email Context

by Damian Poddebniak[1], Fabian Ising[1], Hanno Böck[2], and Sebastian Schinzel[1]

[1] Münster University of Applied Sciences [2] Independent Researcher

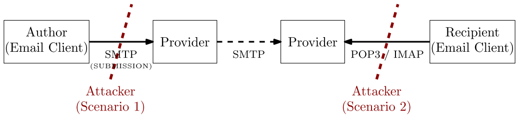
Full paper published at USENIX Security 21

Introduction

Connections between email clients and servers provide two ways to be protected with TLS: While implicit TLS encrypts the connection from the start and runs on a separate port, STARTTLS provides a mechanism to upgrade existing unencrypted connections to TLS.

Sometimes STARTTLS is seen as an opportunistic encryption mode that provides TLS protection only when available. This is trivially vulnerable to downgrade attacks. However, modern email clients usually have the expectation that STARTTLS is enforced, and when enabled, no unencrypted communication is possible.

Upgrading connections via STARTTLS is fragile and vulnerable to a number of security vulnerabilities and attacks. We found more than 40 vulnerabilities in STARTTLS implementations. We conclude that these vulnerabilities are so common that we recommend to avoid using STARTTLS when possible.



Attacks

We assume a Meddler-in-the-Middle (MitM) attacker who can modify connections established between an email client and the email server of an provider.

Stealing Login Credentials with SMTP and IMAP via Command Injection

In 2011 Postfix developer Wietse Venema described a bug in STARTTLS implementations that allowed injecting plaintext commands that a server interprets as if they were part of the encrypted connection. This works by sending additional commands with the STARTTLS command to the server in the same TCP segment.

We found that despite being known since 2011, this vulnerability is still very common. We found 15 vulnerable implementations, and in scans, 2% of all mail servers showed this vulnerability.

This command injection can be used to steal credentials with the SMTP and IMAP protocols.

Our attack requires a Meddler in the Middle (MitM) attacker that can modify network traffic and has login credentials for their own account on the same server. The attacker can inject commands that authenticate them and then start sending (SMTP) or storing (IMAP) an email. The login credentials sent by the victim will be stored in the email that the attacker can access.

The command injection can also be used for a cross-protocol attack to serve HTTPS content with the mail server's certificate. Detailed descriptions of these attacks can be found in our paper.

Mailbox content forgery via Response Injection

We discovered an attack similar to the command injection in email client applications. We call this a response injection. This bug affected many popular mail clients, including Apple Mail, Mozilla Thunderbird, Claws Mail, and Mutt.

By injecting additional content to the server message in response to the STARTTLS command before the TLS handshake, we can inject server commands that the client will process as if they were part of the encrypted connection. This can be used to forge mailbox content.

IMAP connection downgrade via PREAUTH and credential-stealing with REFERRAL

In the IMAP protocol, a server can signal the client in the first message that it is already authenticated via the PREAUTH command. The protocol forbids using the STARTTLS command in an authenticated state. Therefore if a client application accepts PREAUTH, it cannot enforce STARTTLS.

A Meddler in the Middle attacker can use this to prevent STARTTLS from upgrading the connection and force a client to an unencrypted connection.

This vulnerability was originally found in Trojitá in 2014. We discovered that multiple other email client applications were vulnerable to the same bug.

This bug is especially severe in combination with the IMAP features Login Referrals and Mailbox Referrals. These commands allow a server to instruct a client to log into another IMAP server. By using PREAUTH to prevent an encrypted connection, an attacker can use referrals to force a client to send credentials to an attacker-controlled server. Fortunately, the referral features are not supported by many clients. We found only one client - Alpine - vulnerable to this combination of PREAUTH and referrals.

Additional Attacks

We found additional attacks, whose security impact may vary for implementations. Please see our paper to learn more about these attacks.

Conclusion

All vulnerabilities described here rely on the transition of an insecure connection to a secure connection. Implicit TLS does not have such a transition and is therefore not vulnerable to any of these attacks. We therefore consider implicit TLS a more secure option than STARTTLS.

We also point out that STARTTLS always introduces at least one extra connection round trip. So implicit TLS generally provides better performance.

Impact

The demonstrated attacks require an active attacker and may be recognized when used against an email client that tries to enforce the transition to TLS. We have informed all popular email client and server vendors and most issues are already fixed. We think that the demonstrated attacks would be difficult to execute on a large scale and we primarily expect them to be used in targeted attacks. As a general recommendation you should always update your software and (to also profit from faster connections) reconfigure your email client to use implicit TLS only (see below).

Recommendations

For Email Client Users

If possible, we recommend that users check and configure their email clients to use SMTP, POP3 and IMAP with implicit TLS on dedicated ports, i.e., SMTP/Submission on port 465, POP3 on port 995, and IMAP on port 993. This is in line with already existing recommendations in RFC 8314 and was already recommended by security professionals before.

Some mail service providers, notably Microsoft and Apple, do not support implicit TLS for SMTP/Submission. We recommend that users ask their mail service providers to offer the more secure implicit TLS option.

For Application Developers

Both email server and client applications should offer implicit TLS by default. In the long term software developers may decide to not support STARTTLS at all and thus simplify both their code and configuration dialogs and files.

We recommend auditing all applications supporting STARTTLS - both on the server and the client side - for the bugs discovered. Most importantly, applications need to ensure that no unencrypted content gets processed as part of an encrypted connection. IMAP applications must make sure that they do not allow PREAUTH in combination with STARTTLS. We provide the EAST toolkit, which allows testing applications.

For Mail Server Administrators

Make sure your server supports implicit TLS for all supported protocols. If possible, consider disabling STARTTLS for IMAP, POP3 and SMTP submission.

If you really need to support STARTTLS, we recommend testing your server with our tool for the command injection vulnerability for all supported protocols. If your server software is vulnerable, you should ask your vendor for a security update.

FAQ

Isn't STARTTLS insecure anyway?

STARTTLS is used in two "modes", "opportunistic", and "enforced". Email clients must authenticate themselves with a username and password before submitting a new email or accessing existing emails. For these connections, the transition to TLS via STARTTLS **must** be strictly enforced because a downgrade would reveal the username and password and give an attacker full access to the email account.

How can I test if my software is vulnerable?

We provide the EAST toolkit that allows testing email clients and servers.

Testing an email server for the command injection is relatively easy with our Command Injection Tester. testssl.sh (dev version) and TLS-Attacker/TLS-Scanner (starttls branch) also check for the command injection.

Testing an email client is more complex, and we refer to EAST's Fake Mail Server component.

Are other protocols with support for STARTTLS or similar mechanisms affected?

We expect to see similar vulnerabilities in other protocols using STARTTLS, e.g., XMPP, FTP, IRC, or LDAP. Thus our recommendation to avoid STARTTLS and use implicit TLS when possible applies to those protocols as well. We encourage security researchers to look for such vulnerabilities in other protocols.

Some protocols only support STARTTLS and provide no implicit TLS mechanism. We recommend that standards bodies define implicit TLS modes for these protocols and that future protocols do this by default and avoid STARTTLS completely.

What about communication between email servers (MTA to MTA)?

Traditionally, STARTTLS between email servers only protects against passive attacks and is vulnerable to active attacks such as STARTTLS stripping. Thus STARTTLS vulnerabilities do not give an advantage to the attacker.

However, efforts like MTA-STS and DANE provide authentication mechanisms for MTA to MTA connections. Therefore server software should be investigated for STARTTLS vulnerabilities as well. Particularly relevant are the buffering bugs, which can happen both for the sending (response injection) and receiving (command injection) side of a mail server. We have found and reported some vulnerabilities in server software during our research.

Currently, there is no standardized way to use implicit TLS for MTA to MTA connections. Therefore, it is not possible to avoid STARTTLS without changes to the protocol specification.

How important is this?

It's not the most important thing you should worry about today.

How can I contact you?

You can reach us via mail or twitter:

- Damian Poddebniak, @dues__, poddebniak@fh-muenster.de
- Fabian Ising, @Murgi, F.Ising@fh-muenster.de
- Hanno Böck, @hanno, hanno@hboeck.de
- Sebastian Schinzel, @seecurity, schinzel@fh-muenster.de

Reported Vulnerabilities

The following is a list of all STARTTLS-related vulnerabilities we found during our research. None of the issues would be present if implicit TLS would have been used exclusively. Due to different kinds of reports, not all issues are publicly documented. All issues were reported more than than 90 days ago.

Email Clients

As an end user, make sure to use the newest version of your email software. The following list serves as a quick check if your client is still affected.

Response Injection (Buffering)

Product	Protocol	Status	Links
Apple Mail (macOS)	SMTP/POP3/IMAP	Fixed in macOS High	CVE-2020-9941, CVE-2021-30696

Product	Protocol	Status	Links
		Sierra 10.13.6/Big Sur 11.4	
Apple Mail (iOS/iPadOS)	SMTP/POP3/IMAP	Fixed in iOS/iPadOS 14.0	CVE-2020-9941
Mozilla Thunderbird	IMAP	Fixed in 78.7.0	CVE-2020-15685, Vendor advisory, Bug report
Claws Mail	SMTP/POP3/IMAP	Fixed in 3.17.6 for SMTP/POP3, See libEtPan for IMAP	CVE-2020-15917
Mutt	IMAP/SMTP/POP3	Fixed in 1.14.4	CVE-2020-14954
NeoMutt	IMAP/SMPT/POP3	Fixed in 2020-06-19	Commit/Patch, see also CVE-2020-14954
Evolution	SMTP/POP3	Fixed in 3.36.4 (evolution-data-server)	CVE-2020-14928
LibEtPan (Mail Framework for C Language)	IMAP/SMTP/POP3	Fixed in repository, unreleased	CVE-2020-15953
Exim (MTA sending)	SMTP	Unfixed (reported privately)	CVE-2021-38371
Gmail (iOS/iPadOS)	SMTP/IMAP	Unfixed (reported privately)	-
Mail.ru, MyMail	SMTP	Unfixed (reported privately, report closed as not applicable)	-
Yandex	SMTP/IMAP	Unfixed (reported privately)	-
PHP (stream_socket_enable_crypto)	SMTP/POP3/IMAP	Unfixed	Bug report (private)

Negotiation and Tampering bugs

Product	Description	Protocol	Status	Links
Gmail (Android)	Leak of emails	IMAP	Fixed (retested in 2021.07.11.387440246)	-
Gmail (Go)	Leak of emails	IMAP	Fixed (retested in 2020.10.15.341102866)	-
Samsung Email	Leak of emails	IMAP	Fixed (untested)	-
Alpine	Untagged responses accepted before STARTTLS	IMAP	fixed in 2.25	CVE-2021-38370, Release notes
Trojitá	Untagged responses accepted before STARTTLS	IMAP	Unknown	Bug report
Mozilla Thunderbird	Server responses prior to STARTTLS processed	IMAP	Fixed in 78.12	CVE-2021-29969, Vendor advisory
KMail	STARTTLS ignored when "Server requires	SMTP	Unfixed	Bug report

Product	Description	Protocol	Status	Links
	authentication" not checked			
Sylpheed	STARTTLS stripping	IMAP	Unknown	Bug report
OfflineIMAP	STARTTLS stripping	IMAP	Unknown	Bug report
GMX / Web.de Mail Collector	STARTTLS stripping	POP3/IMAP	Fixed	-
Mail.ru, MyMail, Email app for Gmail	STARTTLS Stripping	SMTP	Unfixed (report closed as not applicable)	-

Avoiding Encryption via IMAP PREAUTH

Product	Status	Links
Apple Mail (iOS/iPadOS)	Fixed in iOS 15.1	-
Mozilla Thunderbird	Fixed in 68.9.0	CVE-2020-12398
Alpine	Fixed in 2.23	CVE-2020-14929, Commit
Mutt	Fixed in 1.14.3	CVE-2020-14093
NeoMutt	Fixed in Release 2020-06-19	Commit/Patch, see also CVE-2020-14093
GMX / Web.de Mail Collector	Fixed	-

Certificate Validation

Product	Protocol	Description	Status	Links
OfflineIMAP	IMAP	Accepts untrusted certificates	Unknown	Bug report
GMX / Web.de Mail Collector	POP3/IMAP	Accepts untrusted certificates	Still allows self-signed	-
Yandex	SMTP/IMAP	Accepts untrusted certificates	Unknown (report closed as not eligible)	-
Mail.ru, MyMail	SMTP	Accepts untrusted certificates (SMTP, IMAP)	Unknown (report closed as duplicate)	-
Outlook (Android & iOS)	SMTP/IMAP	Certificate hostname not checked (SMTP, IMAP)	Unknown (report closed as low/medium severity)	-
Geary	SMTP/IMAP	Accepting an untrusted certificate creates a permanent trust exception for all certificates	Fixed in 3.36.3	CVE-2020-24661
Trojitá	SMTP	Accepts untrusted certificates	Fixed in repository (77ddd5d4) (no official releases)	CVE-2020-15047
Ruby Net::SMTP	SMTP	Only checks hostname, ignores certificate signature	Fixed in 2.7.2	Bug report

Crashes

Product	Protocol	Description	Status	Links
Alpine	IMAP	Crash when LIST or LSUB send before STARTTLS	Fixed in 2.25	CVE-2021-46853, Release notes
Balsa	IMAP	Nullptr dereference when TLS required and PREAUTH send	Fixed in 2.5.10	CVE-2020-16118

Product	Protocol	Description	Status	Links
Balsa	IMAP	Stack overflow due to repeated BAD answer to CAPABILITY command	Fixed in 2.6.3	Bug Report
Balsa	IMAP	Crash on untagged EXPUNGE response	Fixed in 2.6.3	Bug Report
Evolution	IMAP	Invalid free when no auth mechanisms in greeting	Fixed in >3.35.91	CVE-2020-16117

Miscellaneous

Product	Protocol	Description	Status	Links
KMail	POP3	Setup wizard in POP3 defaults to unencrypted connections	Unfixed	Bug Report
KMail	POP3	Config shows "encrypted", but it isn't	Unfixed	CVE-2020-15954
KMail	SMTP/IMAP	Dialog loop "forces" the user to accept invalid certificates	Unfixed	Bug Report
Mozilla Thunderbird	POP3	Infinite loop when POP3 server replies with -ERR to STLS command	Unknown	Bug Report
Trojitá	SMTP/IMAP	Hard to choose implicit TLS due to typo (German)	Fixed	Bug Report
Trojitá	SMTP	SMTP defaults to plaintext on port 587	Unknown	Bug Report

Email Servers

We found 320.000 vulnerable email servers in an Internet-wide scan and conducted a coordinated disclosure involving different CERTs. It is impracticable to inform and keep track of the update process of all mail service providers on the Internet, and thus we identified and prioritized popular mail service providers. We only list these in the following table.

Server issues are generally more severe than client issues. Unfortunately, no client configuration prevents server issues from being exploited, not even the usage of implicit TLS. Thus, you must ensure that your server (or your mail service provider) is not affected by STARTTLS issues.

You can use our command_injection_tester tool to verify that your server is not affected by the most severe issue.

Command Injection (Buffering)

Product	Protocol	Status	Links
Nemesis (used by GMX / Web.de, provider)	POP3/IMAP	Fixed (reported privately)	-
Interia.pl (provider)	SMTP/POP3/IMAP	Fixed (reported privately)	-
Yahoo (only MTA-to-MTA, provider)	SMTP	Unfixed (reported privately)	-
Yandex (provider)	SMTP/POP3/IMAP	Unfixed (reported privately)	-
s/qmail	SMTP	Fixed in 4.0.09	CVE-2020-15955
Coremail	SMTP/POP3/IMAP	Unfixed (reported via CERT)	-
Citadel	SMTP/POP3/IMAP	Unfixed	CVE-2020-29547, Bug report
Gordano GMS	POP3/IMAP	Unfixed	CVE-2021-37844
recvmail	SMTP	Fixed in 3.1.2 (reported	-

Product	Protocol	Status	Links
		privately)	
SmarterMail	POP3	Fixed in Build 7537	CVE-2020-29548
Burp Collaborator	SMTP	Fixed in 2020.9.2	Bug report, Vendor release notes
Dovecot	SMTP	Fixed in 2.3.14.1 and 2.3.15	CVE-2021-33515
Mercury/32	SMTP/POP3/IMAP	Fixed in 4.90	CVE-2021-33487
QMail Toaster (1.4.1)	SMTP	Project discontinued	-
Courier	POP3	Fixed in 1.1.5 (reported privately), known since 2013	Discussion from 2013, CVE-2021-38084, Fix
PHP (stream_socket_enable_crypto)	SMTP/POP3/IMAP	Unfixed	Bug report (private)

Session Fixation

Product	Protocol	Status	Links
Citadel	POP3/IMAP	Reported via forum, unfixed	Forum with report, CVE-2021-37845
IPswitch iMail	POP3/IMAP	Fixed in iMail 12.5.8	CVE-2021-37846, Changelog

Miscellaneous Issues

Product	Protocol	Description	Status	Links
Nemesis (used by GMX / Web.de, provider)	SMTP	Advertises authentication before STARTTLS even though it is disabled	Fixed (reported via Bugbounty)	-

Media reports

Golem.de: Sicherheitsrisiko STARTTLS
The Hacker News: Dozens of STARTTLS Related Flaws Found Affecting Popular Email Clients
The Record: STARTTLS implementations in email clients & servers plagued by 40+ vulnerabilities
SecurityLab.ru: Десятки уязвимостей в протоколе STARTTLS затрагивают популярные почтовые клиенты
LWN: STARTTLS considered harmful
Bulletproof TLS Newsletter: Vulnerabilities show fragility of STARTTLS
APNIC blog: Vulnerabilities show why STARTTLS should be avoided if possible

Presentations

USENIX Security '21 - Why TLS is better without STARTTLS
Driving IT Conference 2021: STARTTLS endangers your E-Mail passwords

Followup research

After we published our research some people found similar bugs in other protocols using STARTTLS.

StartTLS in LDAP
nbdkit: Reset structured replies on STARTTLS (CVE-2021-3716)
fetchmail: STARTTLS session encryption bypassing (CVE-2021-39272)
curl 7.79.0 fixes two STARTTLS vulnerabilities
CVE-2021-38542: Apache James vulnerable to STARTTLS command injection (IMAP and POP3)

First published: 2021-08-09, last changes: 2022-11-03