A Multifaceted Study on the Use of TLS and Auto-detect in Email Ecosystems

Ka Fun Tang Che Wei Tu Sui Ling Angela Mak Sze Yiu Chau

Department of Information Engineering

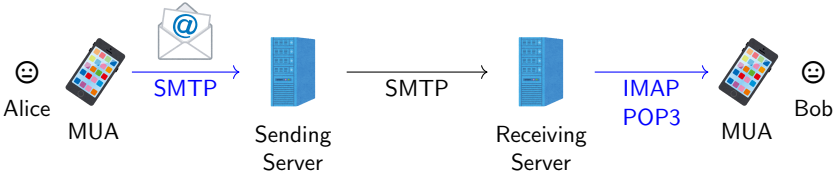
The Chinese University of Hong Kong

February 26, 2025

香港中文大學
The Chinese University of Hong Kong

A Brief Introduction to Email Ecosystems

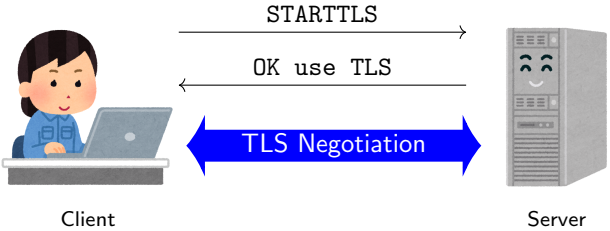
- Email ecosystems use IMAP, POP3 and SMTP protocols to connect Mail User Agent (MUA) and Mail Transfer Agent (MTA) together



Connection Mechanisms Used in Email Protocols

Two types of security mechanisms are commonly used in email protocols:

1. Implicit TLS: TLS channel is established when client connects to server
2. STARTTLS: Both client and server have to negotiate in plaintext phase before upgrading to TLS

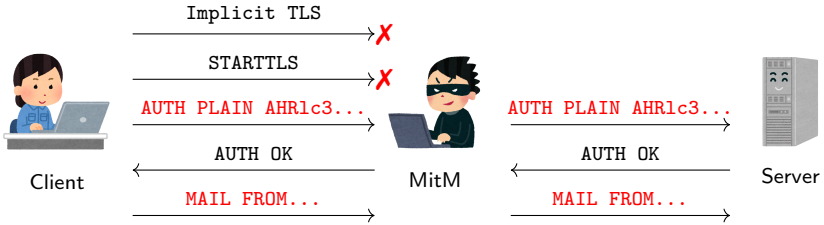


Adversary Model

- A man-in-the-middle adversary model
- Can observe, modify and block the traffic
- e.g. Internet Service Provider, Wi-Fi Access Point. . .

Problems with Auto-detect Mechanism

- For better usability, some email clients will try out different TLS options with the server
- An active MitM can intercept the connection and **opportunistically downgrade** the connection if opportunistic TLS is used



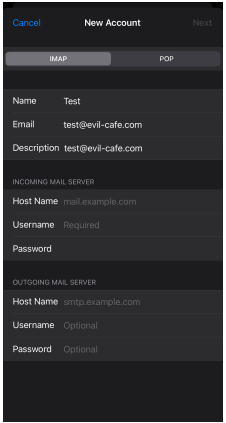
Problems of STARTTLS Implementation

- If both implicit TLS and STARTTLS connections cannot be made, there will be two possible outcomes:
  1. Client proceeds with unencrypted connection
  2. Client terminates connection if TLS cannot be established
- The relative security of the TLS usage options can be ordered as:

Implicit TLS  > STARTTLS  > no-TLS 

Email Client with Auto-detect

If you have used iOS, you may have seen this...



Methodology

1. Designed 4 test cases targeting on their security mechanism and certificate validation on 49 email clients
2. Gathered 1899 university email setup guides and manually inspected 810 custom email server setup guides to understand how IT admins instruct users to setup email clients
3. Conducted server-side evaluation on 798 certificate chains
4. Tested the server-side partial countermeasures on the server domains collected from setup guides

Experimental Setup

- Created 4 test cases to investigate clients' behaviour when STARTTLS or implicit TLS cannot be established
- Modified *mitmproxy* to intercept the network traffic
- To mimic real-world scenarios, mail server is deployed on a purchased domain using *dovecot* and *postfix*
- For test cases construction, please refer to the paper

Findings

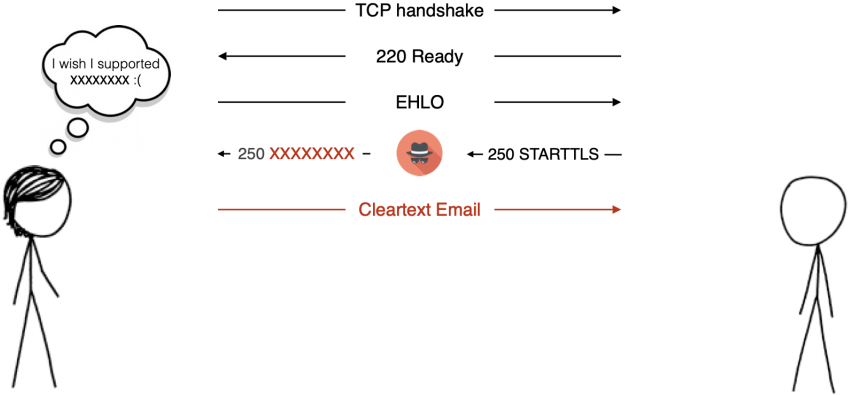
Out of 49 clients, 19 clients may **silently** downgrade to no-TLS,

- 10 clients will downgrade to no-TLS through auto-detect mechanism
- 9 clients will downgrade to no-TLS even specifying the use of STARTTLS
- Using classic + **new variants** of TLS stripping attack

For these email clients, MitM can obtain the user credentials.

Implicit TLS  > STARTTLS  > no-TLS 

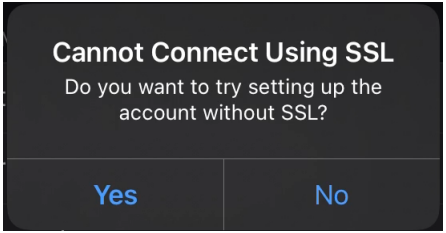
STARTTLS Stripping (1)



https://zakird.com/slides/cccmail.pdf

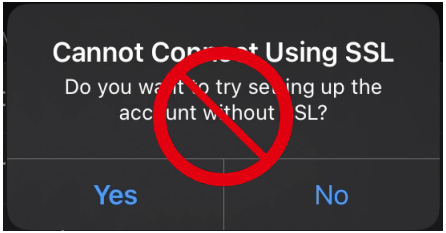
The Case of iOS Apple Mail

When client fallbacks to no-TLS connection, normally a warning prompt will pop up:



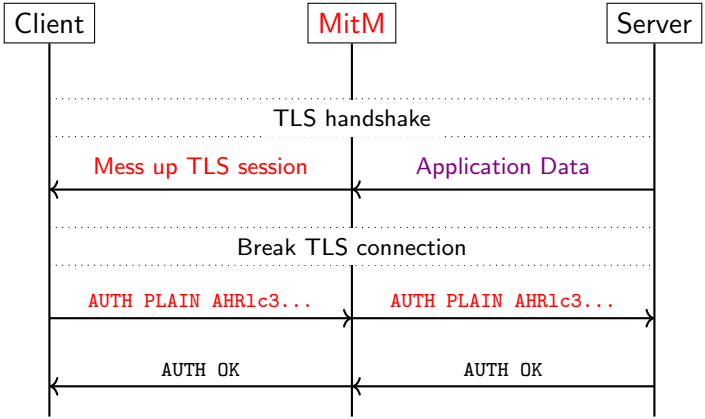
The Case of iOS Apple Mail (Cont'd)

BUT the prompt does not appear...



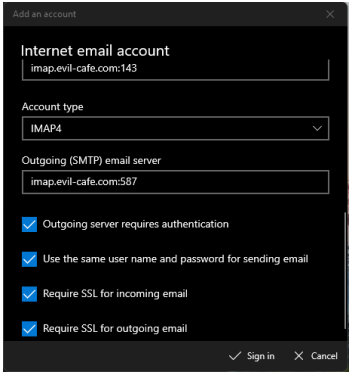
The Case of iOS Apple Mail (Cont'd)

A novel variant of TLS stripping attack, without triggering the prompt!

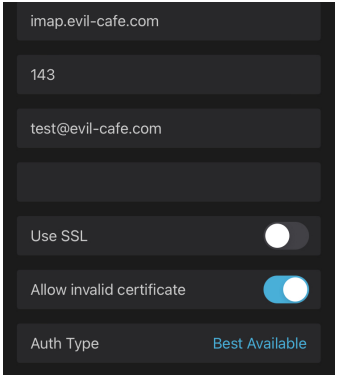


Findings (Cont'd)

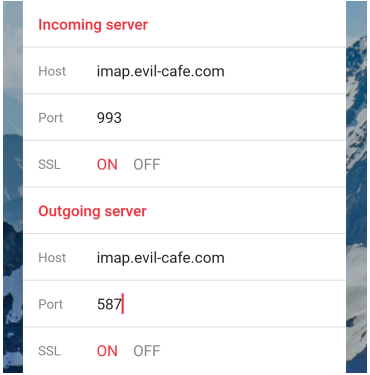
- Some clients may use different wordings regarding the use of TLS



(a) ☑: STARTTLS
☐: no-TLS



(b) ☑: Implicit TLS
☐: STARTTLS



(c) ☑: Implicit TLS
☐: no-TLS

Experimental Setup

- To investigate how clients handle certificate validation in the following scenarios:
  - Self-signed certificate 
  - Expired certificate 
  - Invalid certificate 

  - Mismatch prefix certificate[1] 
- Replace server certificate on the test mail server to see behaviours of the clients

[1] An extra domain is purchased to generate a valid certificate

Findings

- Out of 49 clients, 19 clients at least miss a critical check on certificate validation
  - 1 client accepts expired certificate
  - 18 clients only validate the certificate chain but not the hostname

    → An attacker can impersonate the server by using a valid certificate from a different domain
- 1 client will prompt user to use no-TLS if the certificate is invalid

  → A passive attacker can observe the traffic if user chooses to proceed with no-TLS

Implicit TLS > STARTTLS > no-TLS

Real-world Deployments

To study how the IT admins instruct users to avoid potential security risks,

- Used Google Custom Search API to get top 10 results from 7045 university domains
- Identifed and read the setup guides manually
- Gathered 810 university setup guides related to custom mail server connection setup
- To further categorise the setup guides:
  - Generic: guides provided minimal information about mail server
  - Specific: guides tailored to a specific email client

Findings

Out of 810 setup guides, 310 (38.27%) are generic guides,

- 90 of them abstractly state the use of auto-detect mechanism

Even for 500 (61.73%) specific setup guides,

- 227 of them instruct users to use auto-detect
- 42 of them mention the use of "Android System Client", which varies from device vendors

Among all the setup guides, none of the setup guides instruct what the user should do if there is a warning prompt in auto-detect

→ Up to users to decide whether to proceed or not

Server-side Evaluation

- A list of mail server domains were curated from the setup guides and collected their respective certificate chains
- Certificate chains are verified using pyOpenSSL and default CA bundle on Ubuntu 22.04
- For analysis related to public key information and lifespan in certificates, please refer to the paper

Findings

- Out of 798 certificate chains, 21 of them failed in chain validation
  - 3 of them are self-signed certificates
  - 4 of them are expired certificates
  - 6 of them missing some issuer certificates
  - The remaining of them do not include `subjectKeyIdentifier` in root certificate
- Out of 414 unique leaf certificates, 13 of them failed in hostname validation
  - 11 of them do not match hostname in both `commonName` and `subjectAltName`
  - 2 of them do not include `subjectAltName`
- Clients should tighten the validation

Disclosure

- The vulnerabilites were responsibly disclosed to the vendors and setup guide issues to the universities
- Apple has confirmed our findings and scheduled a fix in Spring 2025
- 2 universities thanked us for our reports and promised us to reevaluate their setup guides

Conclusion

- We studied the following aspects in the email ecosystems:
  - client-side implementations
  - setup guides offered by IT admins
  - server-side deployments
- Some email clients show improper handling of security downgrade and certificate validation, especially on the auto-detect mechanism
- IT admins should instruct users more explicitly on the use of TLS and how to handle warning prompts
- Specification should not leave for vendors to interpret what to implement

Thank you!

Questions?

Problems with RFC

- In RFC3207 (SMTP), if a server responds with `454 TLS not available due to temporary reason`, client can choose to continue with the connection or not
- In RFC9051 (IMAP), if no explicit user configuration is set, client can connect to the server simultaneously with STARTTLS and implicit TLS connection
  $\rightarrow$ may create a race condition

Findings

- RFC9051 states that IMAP clients can try both implicit TLS and STARTTLS concurrently
- Only 3 clients implemented this feature
  - 1 client implemented it on SMTP
  - 3 connections are observed on port 465, 587 and 25
- An on-path attacker maybe able to block TLS traffic to force the connection to use no-TLS connection

Server-side Countermeasures

To force clients to use TLS connection,

- IMAP servers may deploy `LOGINDISABLED` capability
- SMTP servers may reply with `530 must issue a STARTTLS command first` if client attempts to login without TLS

We probed the server domains to see if they have deployed such mechanisms.

Findings

- 100 (62.5%) IMAP servers and 202 (69.6%) SMTP servers have supports on the aforementioned mechanisms
- For IMAP servers, instead of listing `LOGINDISABLED` capability, 10% of IMAP servers will hide all the authentication methods from the list of capability, and leave STARTTLS as the only option for the client

→ Mail servers already deployed countermeasures that can prevent cleartext transmission of user credentials