Measuring SMTP STARTTLS Deployment Quality

By Binu Ramakrishnan, Security Engineer, Yahoo Mail

Summary

At Yahoo, our users send and receive billions of emails everyday. We work to make Yahoo Mail easy to use, personalized, and secure for our hundreds of millions of users around the world. In line with our efforts to protect our users' data, our security team recently conducted a study to measure the deployment quality of SMTP STARTTLS deployments. We found that while the use of STARTTLS is common and widespread, the growth has slowed in recent years. Providers with good/valid certificates have better TLS settings compared to others, and we believe there is an important need to improve the quality of

STARTTLS deployments to protect messages – and therefore, users – from active network attacks.

The Modern Mail Ecosystem

Simple Mail Transfer Protocol (SMTP) is the underlying protocol used for email transmission, especially when sending or receiving email between different providers. The SMTP protocol does not require encryption by default, and mail providers like Yahoo depend on the STARTTLS extension to encrypt messages in transit. Unfortunately, not all providers support STARTTLS when they send or receive emails, potentially exposing them to network eavesdropping.

The diagram below offers a simplified view of a modern mail ecosystem. Communication between service providers are over the SMTP protocol, and the providers use MTAs to send and receive messages to/from other providers. MTAs speak the SMTP protocol and use STARTTLS to encrypt the messages in transit. To send a message, the sender (MTA outbound) resolves a mail exchanger record (MX) for the recipient's domain from DNS. The MX record contains the recipient's (MTA inbound) server name. Once the recipient's server name is resolved, the sender connects to that server and transmits messages.

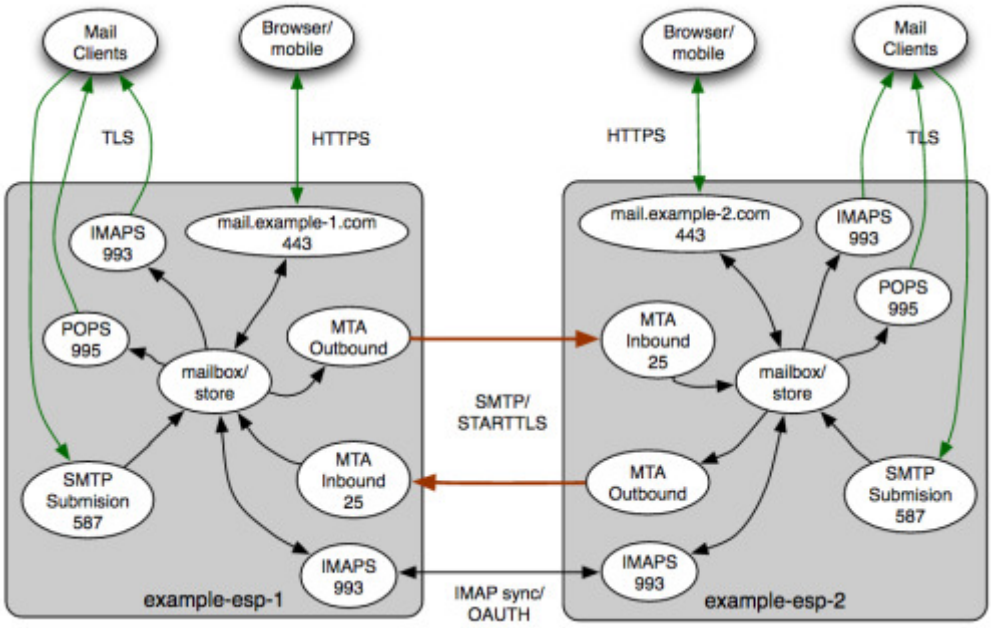

Figure 1: A high level overview of a mail ecosystem

STARTTLS has received a lot of attention in recent years. Around half a dozen studies were published and presented in 2015 (see Appendix), all of which underscore the importance of securing mail delivery infrastructure against mass surveillance and network eavesdropping. Since mail is an open system, a collective industry wide effort is critical to secure our email communication.

What is STARTTLS ?

STARTTLS is an extension that enables opportunistic upgrades of plaintext communication to encrypted communication between STARTTLS aware client and server. The diagram below shows an SMTP session between a client and a server. When the server desires to receive emails over TLS, it returns 250 STARTTLS back to client in response to EHLO from client. If the client supports TLS, it may initiate a TLS handshake and once the TLS session is established, messages will be sent over an encrypted channel.

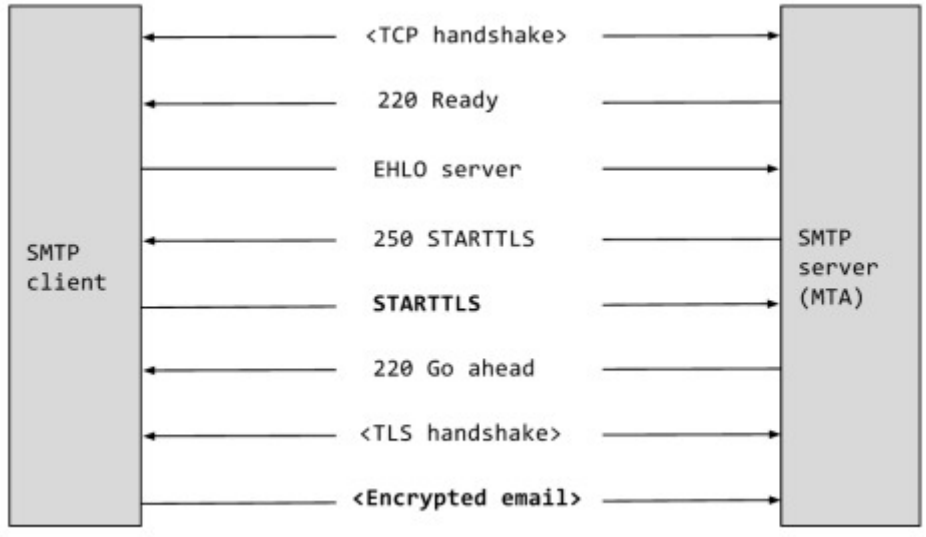

Figure 2: SMTP STARTTLS session between a client and a server

STARTTLS provides protection against passive attacks and, in fact, the opportunistic nature of STARTTLS drove widespread adoption of TLS in SMTP. At the same time, 'opportunistic' encryption also means that STARTTLS is not effective against MITM (active) attacks because of: (1) **STARTTLS downgrade attacks** - by stripping STARTTLS from an active SMTP session that forces messages to send over cleartext, and (2) the possibility of **DNS MX spoof attacks** in which a compromised name server returns a spoofed MX target host or IP address and diverts the traffic through the attacker's mail server.

Methodology

For this study, we collected 12M unique domains from a 30 day period in January 2016 of mail outbound logs. Of the 12M domains we scanned, we gathered stats for 9M domains with 3.7M unique MX hosts and ~1M unique IP addresses. The data collected is aggregated and presented in multiple buckets – unique Domain, MX, IP etc. This data is also compared with a previous study (slides) we did in May 2015 (presented at M3AAWG 34th General Meeting in Dublin, Ireland). We scanned the domains with a fast TLS scanner written in Go and used Unix tools to analyze the data.

Caveats

The Go TLS implementation has limited cipher support: specifically, it does not support deprecated/insecure ciphers. It also does not have SSLv3 client side support. This study is based on domains we collected from Yahoo, and we considered only those domains with at least three or more emails sent during that period.

Findings

Our findings are grouped and presented in buckets based on:

- Domains - Unique domains (9M)
- MX - Unique MX hosts (3.7M)
- IP - Unique IP addresses (1M)
- Valid Cert - Unique MX with valid CA signed certificate (1.8M)
- Strict validation - Valid cert with a matching host name (peer verify) (626K)

Note that these 9M domains are hosted by 3.7M MX hosts which in turn map to 1M unique IP addresses. Many domains share the same MX and many MXs share the same IP.

STARTTLS Adoption

Around 80% of MXs we scanned support STARTTLS. When compared to a similar study we conducted last year, STARTTLS adoption rate was flat with no significant growth expected in the near future. Adoption rate in the case of the unique IP bucket is lower than the other two buckets.

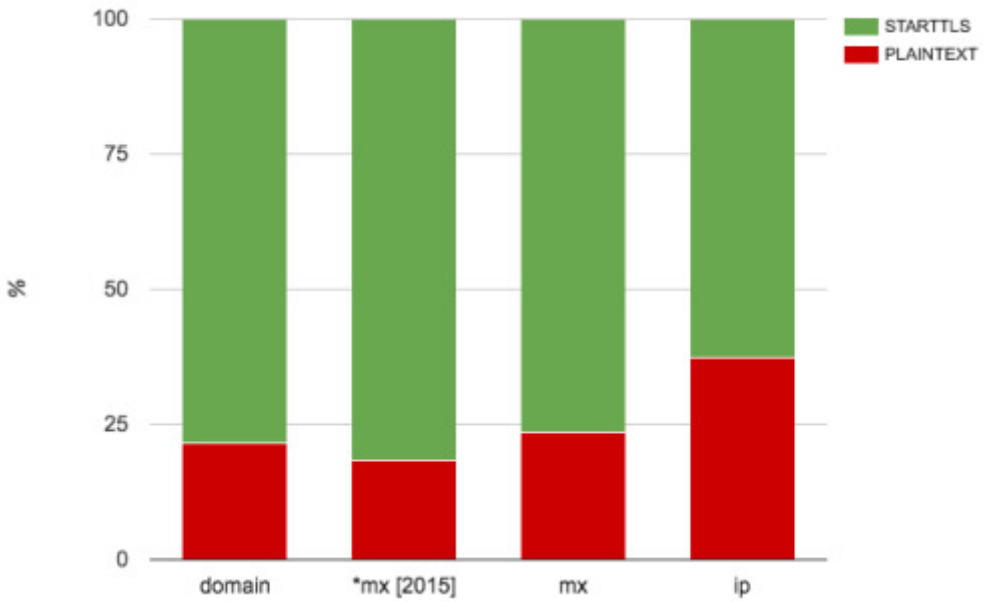

Figure 3: STARTTLS adoption (**data from 2015)*

TLS X.509 Certificates

Public Key Size

Public key size is the length of the RSA (or ECDSA) key used by the server. An RSA key size less than 2048 bits is considered weak, but we found that around 14% of MXs are still using weak 1024 bit RSA public keys. Interestingly, key sizes in the last two buckets were found to be more compliant than other buckets, which is expected considering that those hosts have valid CA signed certificates. We also observed five valid ECDSA certificates.

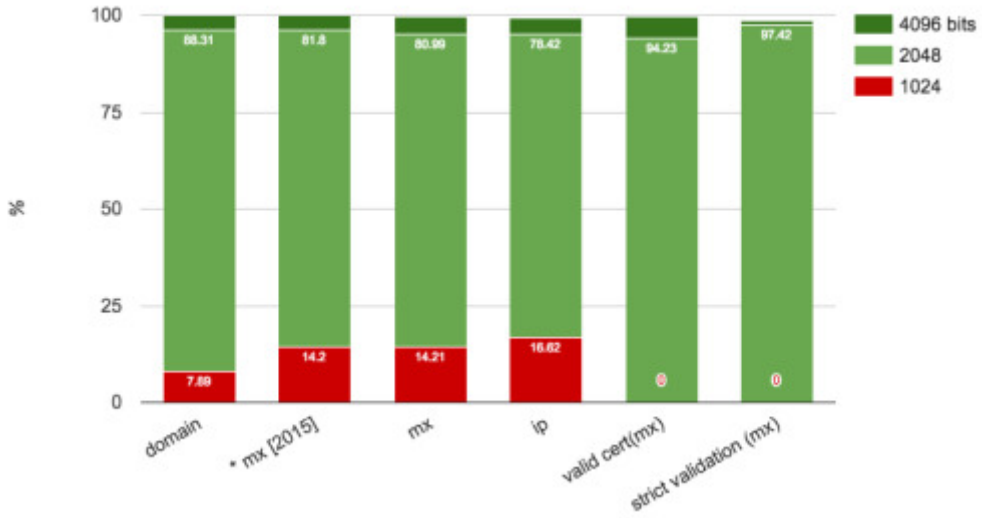

Figure 4: Public key size distribution chart

Signature Algorithm

Signature algorithm is the cryptographic hash algorithm used by certificate authorities to sign TLS certificates. SHA1 based certificates are deprecated and currently being phased out. We have observed a few RSA-SHA1 based certificates issued in 2015 but found no RSA-SHA1 certificates issued in 2016 (as of January 31, 2016). However, a significant number of these SHA1 certificates remain valid well beyond 2016, which is a concern. Almost all browser vendors (in the HTTPS world) decided to mark SHA1 signed certificates as ‘untrusted’ if they encounter them after January 1, 2017. When compared with data from 2015, we find a significant increase in SHA256-based certificates which is expected. You may also notice a small percentage of MD5 based certificates, especially in Domain, MX and IP buckets. Note that almost all are either expired or self-signed.

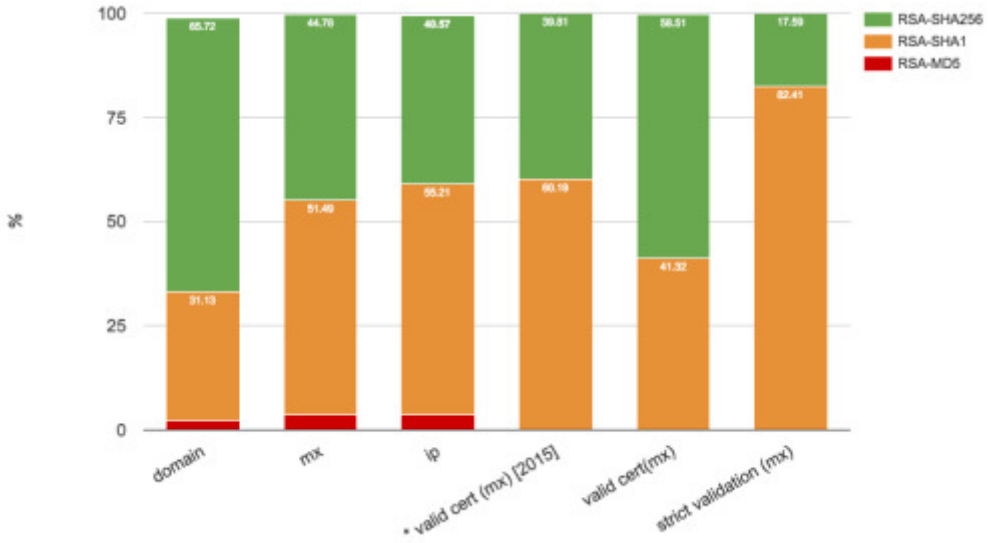

Figure 5: Signature algorithm distribution chart

Certificate Validation

This chart presents the certificates distribution in three groups: (1) Untrusted, (2) ValidCert, and (3) StrictValidCert. The ValidCert group represents certificates that chain to a trusted root CA and the StrictValidCert is the grouping of valid certificates with peer verified. Note that peer verification is against the MX hostname, not to the email domain. The unique domain bucket has more valid and strict-valid certificates than the other two buckets with more than 50% certificates that are peer-verified. This was largely because the large mail service providers that host millions of third party domains mostly use valid certificates for STARTTLS. In the case of unique IP category, we find a large percentage of untrusted certificates.

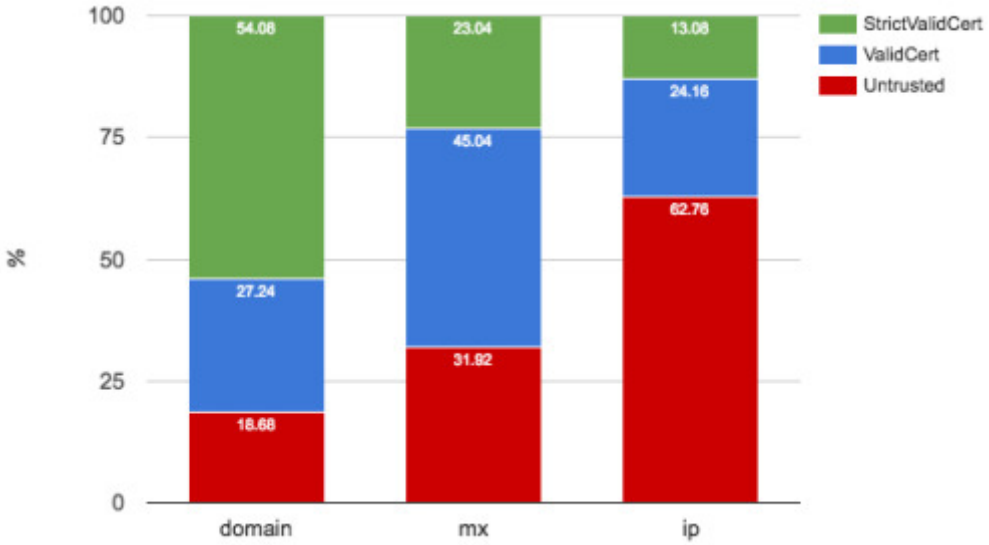

Figure 6: Certificate validation

Certificate Validation - Error-type Distribution

This chart shows the distribution of certificate validation error types. Hostname mismatch (PeerVerifyFailed) is more prevalent than self-signed/expired certificates in the domain and MX buckets. This was largely because the large hosted email providers prefer to use CA signed certificates over self-signed certificates. Interestingly, even the large mail providers grapple with hostname mismatch. Self-signed and expired certificates are more prevalent within the IP bucket.

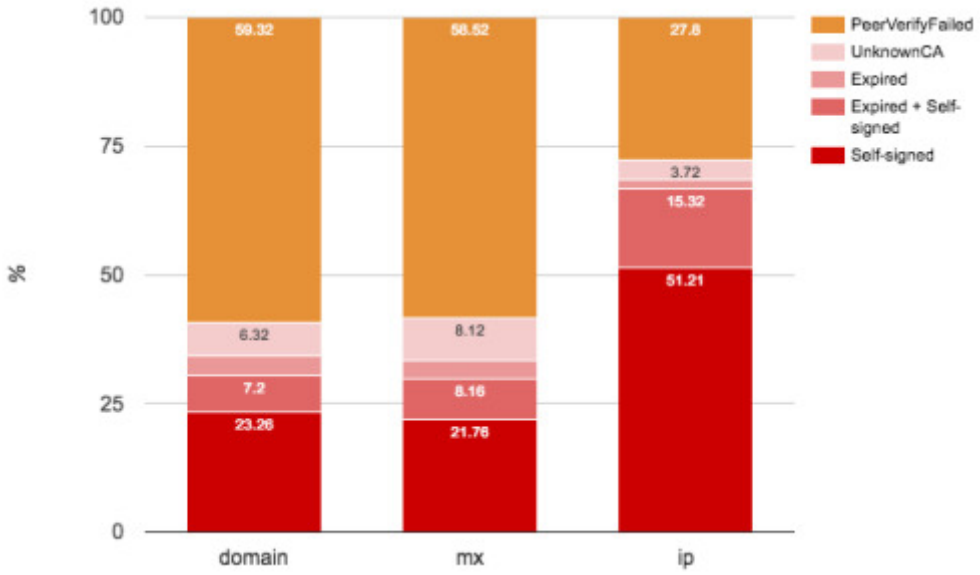

Figure 7: Certificate validation error-type distribution

Certificate Chain Depth

Chain depth of zero mainly represents self-signed certificates (in red) and is more prevalent in the first three buckets. However, for valid and strict-certs buckets, the chain depth is either two or three, which is expected.

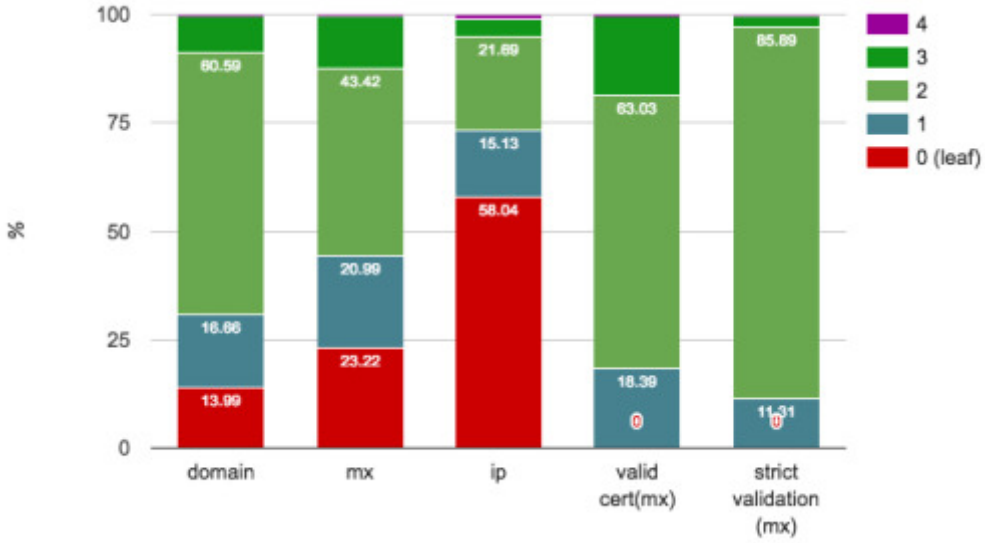

Figure 8: X509 certificate chain depth distribution

TLS Session

TLS Protocol Version

TLS version 1.2 usage increased since last year. The usage is higher in verified and strict-certs buckets. TLS1.1 usage is not statistically significant.

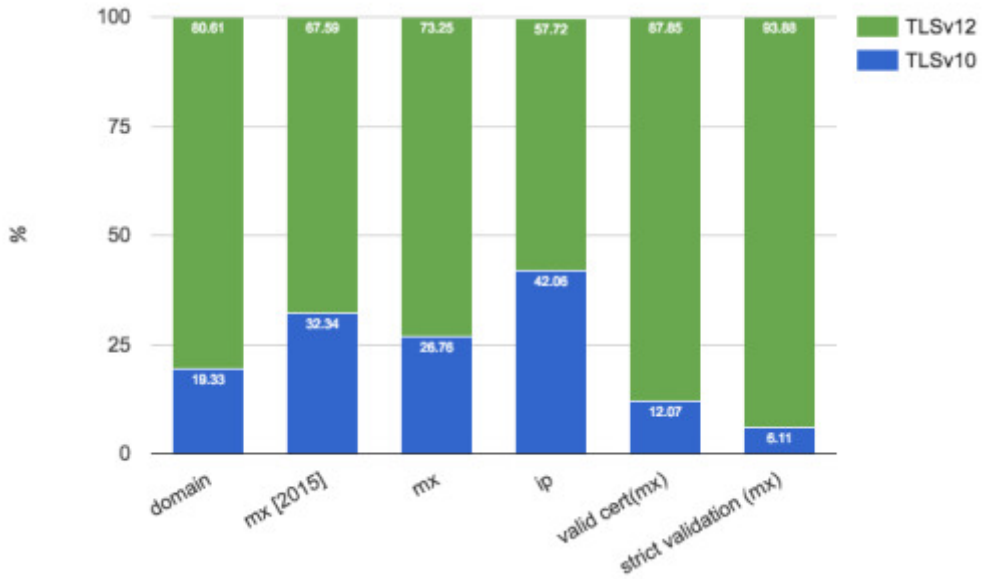

Figure 9: TLS protocol version

Negotiated Ciphers

The data presented in this chart may not be 100% accurate, as our scanner is written in Go and the Go TLS implementation has limited cipher support. In particular, the Go TLS implementation does not support deprecated/insecure ciphers and DHE cipher suites, nor does it have SSLv3 client side support.

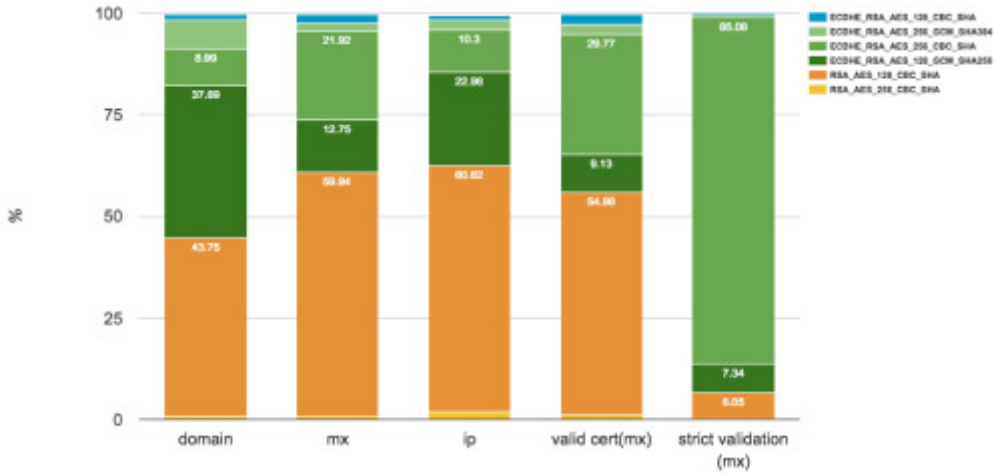

Figure 10: TLS session cipher distribution

Deployment Quality - Focus areas for email service providers

Though STARTTLS protects against passive network eavesdropping, it is not effective against active MITM attacks in its current form. An industry-wide effort is underway to strengthen the mail delivery infrastructure and the end goal is to protect against active MITM attacks, thereby upholding users' privacy. Below are a few recommendations that can greatly improve STARTTLS deployment quality. While these steps alone cannot protect against active attacks, by implementing these changes, mail providers can meet the baseline requirements to fight against pervasive monitoring attacks and increase the difficulty of active attacks.

Server side

- **Eliminate self-signed and expired certificates**. There are a few certificate authorities that provide certificates free of cost, including Let's Encrypt. Let's Encrypt is a new certificate authority that provides free TLS certificates with the ability to automate certificate refresh, which solves the cert expiration issue. DNS-based Authentication of Named Entities (DANE) is an alternate way to authenticate STARTTLS server entities without a certificate authority. DANE relies on Domain Name System Security Extensions (DNSSEC) for security, but the challenge is that DNSSEC is not widely deployed and its adoption rate remains low. DANE does not require certificates issued by certificate authorities.
- **Upgrade valid certificates to conform to strict validation (peer verify)**. Operators must make sure their certificates are not only valid, but also match their hostname. We observed a large number of valid certificates with hostname mismatches, some of which were from large mail providers.
- **Replace SHA1 based certificates with SHA256 based certificates**. The SHA1 cryptographic hash algorithm is considered weak and the industry recommendation is to transition from SHA1 signed certificates to SHA256 signed certificates as early as possible.
- **Leverage strong ciphers and TLS protocol versions**
    - Disable SSLv2 and SSLv3 protocol versions
    - Enable Perfect Forward Secrecy (PFS) algorithms (ECDHE, DHE (dhparam > 2048))
    - Track vulnerabilities and patch TLS library (e.g., OpenSSL) as applicable

Refer to Mozilla's server side TLS configuration and https://cipherli.st/ for good TLS config examples

Client/Sender

- **Strict certificate validation**. Validate MX certificates and verify them by matching the hostname of the server with the name in the certificate presented by the server. A soft validation is recommended initially, which is useful for Log and monitoring (see below).
- **Log & monitoring**. Data related to validation failures when connecting to a recipient server help to detect active network attacks. Log events such as STARTTLS=false, MX mismatches, and cert validation failures for this purpose.
- **Keep up to date with root CA certificates bundle**. SMTP clients, unlike browsers, have no standard mechanism to update CA bundles. In recent years, Microsoft and Mozilla pruned their CA bundle and removed many old root certificates. Our recommendation is to keep your root CA bundles up to date, irrespective of which root CA bundle you trust.
- **Certificate revocation support (CRL, OCSP, OCSP stapling)**. Considering the opportunistic nature of current SMTP deployments, until now there was no compelling reason to check whether the certificates presented by servers are revoked or not. But this feature may become more important in coming years.

Recommendations

The use of STARTTLS is common and widespread; however, its growth has slowed in recent years. Through our study, we found that providers with good/valid certificates have better TLS settings compared to others. There is an important and fundamental need to improve the quality of STARTTLS deployment in order to protect messages – and therefore, users – from active network attacks. As a baseline requirement, email providers should work to eliminate self-signed, expired certificates and use good ciphers with PFS on SMTP servers. Senders should validate the certificates and log validation failures, as the failure logs can provide valuable insights and use it for reporting.

Appendix

TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication http://arxiv.org/pdf/1511.00341v2.pdf

No Need for Black Chambers: Testing TLS in the E-mail Ecosystem at Large http://arxiv.org/pdf/1510.08646v2.pdf

Neither Snow Nor Rain Nor MITM... An Empirical Analysis of Email Delivery Security http://conferences2.sigcomm.org/imc/2015/papers/p27.pdf

Analysis of TLS in SMTP World http://www.slideshare.net/BinuRamakrishnan/analysis-of-tls-in-smtp-world

The Current State of SMTP STARTTLS Deployment (2014) https://www.facebook.com/notes/protect-the-graph/the-current-state-of-smtp-starttls-deployment/1453015901605223/

Yep! We Poked all your mail daemons http://www.slideshare.net/SBAResearch/yep-we-poked-all-your-mail-daemons

STARTTLS Everywhere (2014) https://github.com/EFForg/starttls-everywhere

Acknowledgments: We want to thank Mike Shema, Elizabeth Zwicky, Suzanne Philion, and colleagues from Yahoo Mail Delivery and Paranoids teams for their support and contribution to this work.

security *smtp* *starttls* *email*

Find Me On