CSE331       Computer Security Fundamentals

11/16/2017         Email

Michalis Polychronakos
Stony Brook University
Email Overview

**MUA:** Mail User Agent
Thunderbird, webmail, Pine, ...

**MSA:** Mail Submission Agent
SMTP (port 587)
Often same as initial MTA

**MTA:** Mail Transfer Agent
SMTP (port 25)

**MDA:** Mail Delivery Agent
IMAP (port 143), POP3 (port 110), local, ...

Typical flow: MUA → MSA → MTA → … → MTA → MDA → MUA
SMTP Transport Example

S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.org>
C: To: "Alice Example" <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 January 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
Email/Messaging Security and Privacy Goals

Protect message content
Verify communicating parties’ identities
Fight spam
Fight phishing
Hide communication patterns

(subject of future lecture)
Who can read my email?

Adversaries with local or remote access to my devices

Intruders, spouse, administrator, ...

Malware, stolen credentials, physical access, ...
Who can read my email?

Adversaries with local or remote access to MTAs and other intermediary servers

Intruders, administrators, other insiders, LEAs, …
Who can read my email?

Adversaries with access to any intermediate network

Intruders, administrators, other insiders, LEAs, …

Passive eavesdropping, MitM, DNS poisoning, …
Confidentiality Threats Recap:

Stored messages

- Compromised system (either local or remote)
  Malware, intruder, insider, stolen/lost device, …

- Compromised authentication
  Password theft, brute-force phone pin, …

Messages in transit

- Eavesdropping and interception

Displayed messages

- Screendump, reflections, shoulder surfing, …
Securing Email Transit

These days encryption is *mandatory* for client-to-server email transmission and retrieval.

MUA ➔ MSA: STARTTLS (port 587/25), SMTPS (port 465)
MDA ➔ MUA: POP3S (port 995), IMAPS (port 993)

```bash
mikepo@capcom:~> nc smtp.gmail.com 25
220 mx.google.com ESMTP i185sm2356739qhc.49 - gsmtp
HELO foo.example.com
250 mx.google.com at your service
MAIL FROM:<mikepo@example.com>
530 5.7.0 Must issue a STARTTLS command first.
```

MTA ➔ MTA relaying: *Another story…*
STARTTLS: Opportunistic Encryption

Many legacy MTAs still do not support TLS
   Fail-open design is necessary

MTAs do their best to deliver messages
   A recipient MTA might present a self-signed certificate (common in antispam and email AV systems)
   There is no PKI for email...

MitM is trivially easy
   STARTTLS command is sent over a plaintext channel (!)
   Analogous to SSL stripping, but in this case the client has no indication that downgrade has happened
   Just assumes that the receiving MTA does not support TLS

Message interception is still possible
   Better than nothing: bulk passive eavesdropping not possible
I Want to STARTTLS

mikepo@capcom:~> nc aspmx.1.google.com 25
220 mx.google.com ESMTP h126si17458667qhh.29 - gssmtp
EHLO foo.example.com
250-mx.google.com at your service, [128.59.23.41]
250-SIZE 157286400
250-8BITMIME
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-CHUNKING
250 SMTPUTF8
STARTTLS
220 2.0.0 Ready to start TLS
<TLS Handshake>
mikepo@capcom:~> nc aspmx.1.google.com 25
220 mx.google.com ESMTP h126si17458667qhh.29 - gssmtp
EHLO foo.example.com
250-mx.google.com at your service, [128.59.23.41]
250-SIZE 157286400
250-8BITMIME
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-CHUNKING
250 SMTPUTF8
STARTTLS
220 2.0.0 Ready to start TLS
<TLS Handshake>

I want to STARTTLS

Can be stripped off by a MitM attacker
How much email was encrypted in transit?

Generally speaking, use of encryption in transit increases over time, as more providers enable and maintain their support. Factors such as varying volumes of email may explain other fluctuations.

Outbound

84%
Messages from Gmail to other providers.

Jan 2016

Inbound

73%
Messages from other providers to Gmail.
A tiny GUI change prompted many networks to deploy STARTTLS
Inbound email encryption: 88%

START 01/01/2013  END 11/12/2017
Google, Yahoo SMTP email servers hit in Thailand

Staff writer | September 12, 2014 | telecomasia.net

Internet users in Thailand have been hit by a massive man-in-the-middle attack aimed at grabbing email login credentials from fake SMTP servers.

The attack has been verified on Google’s and Yahoo’s email servers and on two of the country’s largest fixed-line ISPs, though preliminary analysis suggest that all SMTP servers are targeted.

The STRIPTLS attack as it has become known works by inserting a man-in-the-middle at the ISPs. This is done via a transparent proxy.
On SMTP, STARTTLS and the Cisco ASA

During the course of trying to increase the security of my e-mail while in transit, I was working on enabling TLS in Postfix to opportunistically encrypt connections to SMTP servers. While verifying my configuration, I ran into an interesting issue.

In order to test my configuration out I was sending e-mails to a Gmail address via Postfix, unfortunately I wasn’t seeing any logging in Postfix indicating that TLS was being used. So I attempted to investigate whether STARTTLS was actually being advertised by manually connecting to Google’s SMTP servers using telnet:

telnet aspmx.1.google.com 25
Trying 2607:f8b0:4001:c02::1a...
Connected to aspmx.1.google.com.
Escape character is ‘]’.
220 ********************
EHLO example.com
250-mx.google.com at your service,
250-SIZE 35882577
250-8BITMIME
250-XXXXX
250 ENHANCEDSTATUSCODES

Every server I connected to in Google’s MX record was not advertising STARTTLS. On a whim, I attempted to connect to Google’s SMTP servers from an entirely different network:

telnet 173.194.68.26 25
Trying 173.194.68.26...
Connected to qa-in-f26.1e100.net (173.194.68.26).
Escape character is ‘]’.
220 mx.google.com ESMTP l3si4081429qct.164
EHLO stomp.colorado.edu
250-mx.google.com at your service,
250-SIZE 35882577
250-8BITMIME
250-STARTTLS
250 ENHANCEDSTATUSCODES
End-to-End Email Encryption

Two major standards: **PGP** and **S/MIME**

- Similar, but incompatible
- Both rely on public key cryptography
- Both support signing and/or encryption

**Main difference: how certificates are signed**

Typical workflow

- Encrypt message with a random symmetric key
- Encrypt symmetric key with the public key(s) of recipient(s)
- Digitally sign a hash of the message

**Metadata still in the clear!**

- Email headers
- Appended "Received:" records
- Subject line
Pretty Good Privacy

De fact standard for secure email

PGP (Phil Zimmermann) ➔ OpenPGP (RFC 4880)
  Gnu Privacy Guard (GPG): GPL implementation

Authentication
  Senders attach their digital signature to the message
  Receivers verify the signature using public-key cryptography

Confidentiality
  Symmetric key encryption
  Random session key generated for each message
  Session key is encrypted with recipient’s public key

Both are typically used on the same message
PGP Encryption

Use a different keypair for signing and encryption

http://www.slideshare.net/rvenkatesh25/network-security-primer
PGP Signed Message Example

From: alice@wonderland.com
Date: Mon, 16 Nov 1998 19:03:30 -0600
Subject: Message signed with PGP
MIME-Version: 1.0
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit
Content-Description: "cc:Mail Note Part"

-----BEGIN PGP SIGNED MESSAGE-----

Bob,

This is a message signed with PGP, so you can see how much overhead PGP signatures introduce. Compare this with a similar message signed with S/MIME.

Alice

-----BEGIN PGP SIGNATURE-----
Version: PGP for Personal Privacy 5.0
Charset: noconv

iQCVAwUBM+oTwFcsAarXHFeRAQEsJgP/X3noON57U/6XVygoFjSY5lTpAduPZ8M
a1Fa1UKu1LLGxmtsbwRiDWLtCeWG3k+7zXDfx4YxuUcofGJn0QaTlk8b3nxADL0
O/EIvC/k8zJ6aGaPLB7rTIizamG0t5n6/08rPwwVkB03tmT8UNMAUCgoM02d6HX
rKvnc2aBPFI=
=muAh
-----END PGP SIGNATURE-----

http://www.slideshare.net/rvenkatesh25/network-security-primer
Encrypted Email: Two Main Challenges

Public key authenticity

Assurance that a public key is correct and belongs to the person or entity claimed

Has not been tampered with or replaced by an attacker

Public key discovery

How can we find the public key of a person/entity?

Especially the very first time we contact them
PGP: Web of Trust

Decentralized trust model
- In contrast to the centralized hierarchical model of PKI
- Users create their own certificates

Users validate other users’ certificates, forming a “web of trust”
- No trusted authorities: trust is established through friends
- Adjustable “skepticism” parameters: # fully and # partially trusted endorsers required to trust a new certificate (1 and 3 for GnuPG)
- Key signing parties

Main problems
- Privacy issues: social graph metadata
- Bootstrapping: new users are not readily trusted by others
- When opinions vary, “stronger set” wins: impersonation through collusion/compromised keys
- Scalability: WoT for the whole world?
S/MIME

Based on standard X.509 certificates
  Analogous operation to SSL: trusted CA sign certificates
  Traditional PKI

Uses multipart MIME to include cryptographic information in the message

Widely supported by most email readers (e.g., iOS)

Works well within corporations
  Certificate distribution through Active Directory infrastructure
S/MIME Signed Message Example

From: alice@wonderland.com
Date: Mon, 16 Nov 1998 19:03:08 -0600
Subject: Message signed with S/MIME
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="simple boundary"

--simple boundary
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit
Content-Description: "cc:Mail Note Part"

Bob,

This is a message signed with S/MIME, so you can see how much overhead S/MIME signatures introduce. Compare this with a similar message signed with PGP.

Alice

--simple boundary
Content-Type: application/octet-stream; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"

MIIQQwYJKoZIhvcNAQcCoIiQDCCEDACACQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3DQEHAaCCDnwwgggGMIJl6ADAgECAlBQQRR9a+a+DX0FHxQOQVHQhPMA0GCSqGSIb3DQEBAUAMGIxETAPBgNVBAcTCEludGVybmV0MRcwFQYDVQQKEw5WZXJpU2lnbiwgSW5jLjE0MDIGA1UECmxMrVyaVNpZ24gQ2xhc3MgMSSBDQSAatIEluZGl2aWR1YmWwU3Vic2NyaWJlZjAeFw05NzA0MjcwMDAwMDBaFw05ODA0MjcyMzU5NTIaMIIBFzERMA8GA1UEBxMISW50ZXJuZXQxFzAVBgNVBAoTD1ZlcmlTaWduLCBCJbmMuMTQwMgYDVQQLEytWZXJpU2lnbiBDbGFzcyAwIENBIC0gSW5kaXZpZHVhbCBTdWJzY3JpYmVYwRAYD
Finding Public Keys

Public PGP key servers
  pgp.mit.edu
  keyserver.pgp.com

Cache certificates from received emails

Integration with user management (LDAP)

Ad-hoc approaches
  List public key on home page
  Print on business card
  Exchange through another medium on a case by case basis

Association with social profiles/identities
  keybase.io
MIT PGP Public Key Server

Help: Extracting keys / Submitting keys / Email interface / About this server / FAQ
Related Info: Information about PGP /

Extract a key

Search String:  
Do the search!

Index:  
Verbose Index:  

- Show PGP fingerprints for keys
- Only return exact matches

Submit a key

Enter ASCII- armored PGP key here:
Michalis Polychronakis

keybase.io/mikepo

8E8D 8F30 8899 8AFF

twitter: polychronakis • tweet
twitter: polychronakis • gist

mikepo has an invitation available
If you know mikepo, you can ask them for an invitation to Keybase.

mikepo from the command line

# first
keybase join  # if you're new, or
keybase login  # if you're not.

# then
keybase push  # if you already have a public key, or
keybase gen  # if this is all new to you

Tracking (6)
- hargikas
- mstamat
- gianluca_string

Trackers (6)
- hargikas
- kontaxis
- mstamat
Biggest Issue: Usability

Non-trivial setup
  S/MIME: complex certificate enrollment process
  PGP: user is responsible for everything

Key management

Key revocation

Public key fingerprints

Poor mail client integration
  Can lead to catastrophic failures: e.g., Enigmail+Thunderbird silent encryption failure

(Let alone key discovery and trustworthiness issues)
Enigmail 1.7 is completely broken for my purposes.

Steps to reproduce the problem:

1) Write an email in TB.
2) Ensure "Force encryption" in Enigmail.
3) Ensure "Force signing" in Enigmail.
4) Recheck encryption and signing settings... OK.
5) Send the email.
6) Look at the received email. OOPS. It is NOT signed and NOT encrypted.

Sorry to say this so directly, but an encryption system, which CONFIRMS to the user in it's graphical user interface on two different places that it will encrypt AND THEN SENDS THE EMAIL WITHOUT ANY ENCRYPTION IN PLAIN TEXT ... is just the BIGGEST IMAGINABLE CATASTROPHIE.

Sorry for my profane language but there is simply no excuse for such...
Swedish media org @Aftonbladet publishes its GPG private key for a second time (first time was in 2012):

@nilssonanders
Sweden's biggest newspaper #Aftonbladet includes their private key in guide to PGP mail them (via @_zulln ) bit.ly/1FfHAO1
PSIRT PGP Key (0x33E9E596)

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: Mailvelope v1.8.0
Comment: https://www.mailvelope.com

xsFNBfM/2KMBEADbwToJM3BCVE10eC22HgVEqNEDppXzuD2dgfKuy0M4tx2LDe7GkPjo6AOsw4yi8bakLiiddpw5B0J/AR1VtIjIDEmS0F9MRZIcV0UKyA5qVc9Baf2nAicY7nezkIJUmyLcIVMC60pqSHzo0Ewy2PZjzxcI4vDGhHmcgfV5XR+uYld3LtVI+A/5jv326LBb16bCNts/tohW2T0LraMPOtCdH848t4tCyp33s8/dZ2C+EOmd4iX1kIymZ1kqEf2Nvcs1aRUXy27sL01VhCymI6UNWcEeHoU2yJxMiBniozBkZUwcR6ysg97nnq633dN9mf7V30PS3zAjlH0Vmzg3B/Nfoqzy2daEU/JDUBhiAo+xr9VF3POoC8JySORgyUm/2t3TTBaH+DnfsUBiqo5U2T0n8x2RlFWxyZNYCTku5J0vPqRBft13SYdJD7LLDPs62nqphaVb34elpwukqIk0TMRu9mB4Ecq+cNFR3ZpN1AKj+H0b/TUJwCJpVju2/3g0wdqHh+OqlvCNm8vIGnQZQW30QWgh/UPoh3RPj+WqnDq88NmBqBI4aAV4u8MqOOb/ztRXVXkAWYHbIZLo925NjFyPuxwhWlCotKen18dZef8bA81RjYuINmCJ0GQus+JG8TJyeBsdK/q8HD5h1kCRSzMHDL+Ra3z/1+FfIwARQAABzR1BZG9iZSBQ01SVCAC8sHNPcRaYXyRyKMvUuY29tsPLBewQQACgLwUCtbYrWuUjAeeEzgAYLcQgHAAwIJEIbAD8Kvh3YWBBUAGqDFgIBAhkBAhsDAb4BAADk2/t/6+FPZw4VmLIPzsTZPoQ/1/KZ7R1lYbQosHveFwyW0WX3uL1sEeD5Qo7HQt6NNMAOw51JswFvFOWIa9u6SHRoU1kGTSESReOq5HnX4DcBubsKmoMS68Pui888wYOIM4Up9V9PuaueO0Uo4sRYhHg5QBoqr6vow5Oc4uTwnfnj7n40H0++291OPJ68B6+kMuQyG4smwxsZlhlqGMOHcs0/CuI3W+n5w+xLM7N5jJCTjNXR+tGmstdmRPSoLWOSos+ZfWfNW0CLKjYuhp3p6H9x8R13wrp2re0GhqKRTg3D4UcAqPs

-----END PGP PUBLIC KEY BLOCK-----
End-to-End vs. Cloud-to-Cloud

IMAP: one of the oldest “cloud” services!
   Keep messages on the server
   Conveniently access them from multiple devices

Useful cloud-based email features
   Powerful search, collaborative SPAM filtering, …
   Need access to the plaintext! Gmail cannot index encrypted messages

Tradeoff: privacy vs. convenience
   Active research on searchable encryption
I don’t like SPAM!
Spam Sources

Commercial entities
   Legitimate or “gray” businesses, advertisers, …

Spammers’ own hosts or open relays ➔ easily blocked

Botnets
   Abuse of ISPs and webmail providers
   Abuse of legitimate user email accounts
   Address harvesting from users’ address books

Beyond email
   Fraudulent messages: Facebook, Twitter, Yelp, Amazon, online comments, forum messages, …
   Fraudulent activities: likes, retweets, clicks, app store rankings, fake reviews, …
Spam lifecycle

Gathering addresses
Valid, active addresses are precious
Stolen address books, web crawling, black market, …

Message content
Advertising, 419 scams, fraud, phishing, malware, …
Anti-spam filter evasion: content obfuscation

Spam email delivery
Valid accounts: newly created (sweatshops), hijacked ones, …
Fake social media accounts “primed” over time
Open relays/proxies (not common anymore)
Malware: most spam comes from infected machines-botnets
Fighting Spam

Content-based filtering
- False positives vs. false negatives
- Local vs. cloud-based

Blacklisting
- IPs/domains of known spammers, open relays, zombie machines, hosts that shouldn’t be sending emails (e.g., ISP DHCP pools), …

Honeypots
- Relays, proxies, spamtraps (fake email addresses)

Outbound filtering (block port 25)
- SMTP authentication is now mandatory by most ISPs

Email authentication
- SPF, DKIM, DMARC, …
Phishing

Spoofed emails pointing to spoofed webpages
  Financial institutions, could services, and other targets

Asking for credentials, credit card numbers, and other sensitive information
  “Your Fedex package information”
  “Your account has been suspended”
  “Your credit card statement”

Spear phishing

*Enticing* messages that appear to come from well-known individuals or businesses
Address Obfuscation

Misspelled/similar domain names
  From: info@paypa1.com  http://www.citybank.com

Misleading <A> tags

Seemingly legitimate/complex/long URLs
  http://www.bankofamerica.com.attacker.net/
  http://www.visa.com:UserSession=2f6q988316484495&usersoption=SecurityUpdate&From@61.252.126.191/verified_by_visa.html

Homographs, internationalized domain names (IDN), punycode
  Most browsers display IDNs only for the system’s configured language
  Punycode if a non-default language or mixed languages are used

Dot-less addresses and other URL encoding tricks
  www.cs.stonybrook.edu ➔ http://130.245.27.2 ➔ http://2197101314

URL shorteners and redirection chains
  Hide the actual destination URL
Recent phishing message targeting SBU users

From: SBU Team <ebrahle2@kent.edu>
Date: Tue, Feb 2, 2016 at 8:42 PM
Subject: cyber security
To: XXXXXXXXXXXX

We've detected spam-like activity in your webmail account, which is against our Acceptable Use Policy (AUP).

Kindly click on the link below to verify that you're the owner of the account and not a spammer.

http://is.gd/stonybrooksecure

We apologize for any inconvenience this may have cause you.

Thanks,
SBU Team
More training of users to click on things...
Phishing Countermeasures

Stop confusing users
  Institutions shouldn’t include links in emails

User education
  Don’t trust links in emails – type the address in your browser
  (analogous to: don’t trust phone calls that ask for your info – always call the number at the back of your card)

Augmenting password logins
  Two-step login: show user-specific information before prompting for the password
  Probably too inconvenient

Anti-phishing filters, tools, …

U2F tokens!
Spear Phishing

Well-prepared, personalized, convincing messages targeted to particular individuals

Seemingly coming from trusted colleagues

Personalized for their target: real names, personal and business information, recent activity (e.g., real purchases), …

Highly effective, used extensively in targeted attacks

Document attachments exploiting 0day vulnerabilities
Links to fake login pages for credentials stealing

Many recent incidents
Maybe rethink email altogether?

Recent secure messaging apps offer many benefits

- True end-to-end encryption: the provider shouldn’t be able to read message contents
- User-friendly verification of contacts’ identities
- Forward security: ensure past communications will be secure even if private keys are stolen
- Open-source design and implementation, code audits

**No spam!** Only approved contacts can send messages

Many encouraging efforts

- Signal, OTR, Pond, …
- Proprietary, but better than nothing: WhatsApp, iMessage

Metadata is still there!