Email

Michalis Polychronakis

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, POP3, local, …

Typical flow:
MUA → MSA →
MTA → … → MTA →
MDA → MUA

©2010 OnlyMyEmail Inc (www.OnlyMyEmail.com) with many thanks to the Gnome project (www.gnome.org) for the images
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
  Spear-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
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

Displayed messages

Screendump, reflections, shoulder surfing, …
Securing Email Transit

These days encryption is mandatory for email transmission and retrieval

MUA $\rightarrow$ MSA: STARTTLS (port 587/25), SMTPS (port 465)

MDA $\rightarrow$ MUA: POP3S (port 995), IMAPS (port 993)

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

MTA $\rightarrow$ MTA

Another story…
STARTTLS: Opportunistic Encryption

Many MTAs still do not support TLS

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 this happened
   Just assumes that the receiving MTA does not support TLS

Eavesdropping is still possible
   Better than nothing: bulk passive eavesdropping not possible
Figure 1 shows the overall results of STARTTLS behavior. From the 'All Email' bar on the left we can see that nearly 60% of all emails are sent via an encrypted connection, but only about 30% pass strict validation. 60% is an encouragingly high percentage, but this number is potentially skewed since the bulk of email volume is sent to a small number of large mailbox providers. We need to aggregate the data in a few different ways in order to compensate for this and get a clearer picture of STARTTLS behavior across all email.
Massive Growth in SMTP STARTTLS Deployment

August 19, 2014 at 10:01am

When we posted in May about the state of STARTTLS deployment, we had no idea that we would see such significant changes to email encryption across the industry in just a few short months. We previously reported that only 28.6% of our outbound notification emails were successfully encrypted and passed strict certificate validation (58% if you count opportunistic encryption). Since STARTTLS encryption requires both sides to deploy it, we encouraged others to take the next step. As a result of recent changes by major providers, most notably Microsoft and Yahoo, 95% of our notification emails are now successfully encrypted with both Perfect Forward Secrecy and strict certificate validation.
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**
- 79%
  - Messages from Gmail to other providers.

**Inbound**
- 56%
  - Messages from other providers to Gmail.

Download data
Google, Yahoo SMTP email servers hit in Thailand

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 2007:f8b0:4001:c02:1a...
Connected to aspmx.1.google.com.
Escape character is '^]'.
220 250-ENHANCEDSTATUSCODES
EHLO example.com
250-mx.google.com at your service,
250-SIZE 35882577
250-8BITMIME
250-XX
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 ESMT 13si4081429qct.164
EHLO stomp.colorado.edu
250-mx.google.com at your service, 1
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

PGP (Phil Zimmermann) -> OpenPGP (RFC 4880)

Gnu Privacy Guard (GPG): GPL implementation

Offers authentication and confidentiality

Sign plaintext, then encrypt

sender’s identity remains hidden: only recipient can verify signature

Encrypt, then sign ciphertext

Verify signature without decryption (e.g., at a gateway)

Anyone can sign a message even if they can’t decrypt it: include sender/recipient identities in plaintext 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 signatues 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+oTwFcsAaarXHFeRAQEJgsJgP/X3noON57U/6XVygOFjSY51TpvuAduPZ8M
aIFalUkCNULLGxmtsbwRaDiWLTcEwG3k+7zXDFx4YxUcOgJn0Qat1k8b3nxADL0
O/EIVc/k8z6aGaPLB7rTlizMGot5n6/08rPwwVrb03tmT8UNMAUCgoM02d6HX
rKvnc2aBPFI=
=muAH
-----END PGP SIGNATURE-----

http://www.slideshare.net/rvenkatesh25/network-security-primer
PGP Additional Features

Compression

Sign -> Compress -> Encrypt
Compression after encryption is pointless (no redundancy)
Signature does not depend on the compression algorithm

Email Compatibility

Ciphertext contains arbitrary 8-bit octects
Some email systems may interpret some of them as control commands
Solution: base64 encoding (33% overhead)

Segmentation

Transparent message segmentation and reassembly for very large messages
Segments mailed separately
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?
Hey, I just got home from the party. The one with the IRC folks?

Yeah.

Hey, how was it?

Got too drunk. I screwed up, bad.

What happened?

There was a girl. No idea who she was. Don't even know her name. I was too drunk to care.

And what, you slept with her?

No. I signed her public key.

Shit, man.
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"

MIIQwYJKoZIhvcNAQcCoIiQDCCDEDACAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3DQEHAaCCDnwwggnGMIIJL6ADAaECAhBQR9a+DX0FHxQOVHQhPMA0GCSqGSIb3DQEBBAUAM8ixETAPBgNVBAcTCEudGVybmV0MRcwFQYDVQQKEw5WZXJpU21nbiwgSW5jLjE0MDIgA1UECxMrVybANpZ24gQ2xh
   c3MgM2BDQSAUeIlu2G12aWR1YygU3Vic2NyawJ1cjeAvFw05NzAaMjcwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDaw

http://www.slideshare.net/rvenkatesh25/network-security-primer
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:
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

mikepo has an invitation available
If you know mikepo, you can ask them for an invitation to Keybase.
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)
WARNING: Enigmail 1.7 *completely**broken*

**Forum:** Enigmail Support  
**Creator:** cleca  
**Created:** 2014-08-12

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 CATASTROPHE.

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):

Anders Nilsson @nilssonanders
Sweden's biggest newspaper #Aftonbladet includes their private key in guide to PGP mail them (via @_zulln) bit.ly/1FfHAOI
End-to-End vs. Cloud-to-Cloud

IMAP: one of the oldest “cloud” services!
- Keep messages at 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
Encrypted Webmail?

Several recent efforts to transparently combine the convenience of webmail with PGP encryption

Is this really possible in a secure way?

JavaScript crypto is not a good idea

Secure JS code delivery?
Secure key storage?
Secure runtime (it’s a web browser!)?

Google end-to-end: implement crypto functionality within a browser extension

More control
Still not trivial
My Fellow Users,

I have been forced to make a difficult decision: to become complicit in crimes against the American people or walk away from nearly ten years of hard work by shutting down Lavabit. After significant soul searching, I have decided to suspend operations. I wish that I could legally share with you the events that led to my decision. I cannot. I feel you deserve to know what’s going on—the first amendment is supposed to guarantee me the freedom to speak out in situations like this. Unfortunately, Congress has passed laws that say otherwise. As things currently stand, I cannot share my experiences over the last six weeks, even though I have twice made the appropriate requests.

What’s going to happen now? We’ve already started preparing the paperwork needed to continue to fight for the Constitution in the Fourth Circuit Court of Appeals. A favorable decision would allow me resurrect Lavabit as an American company.

This experience has taught me one very important lesson: without congressional action or a strong judicial precedent, I would _strongly_ recommend against anyone trusting their private data to a company with physical ties to the United States.

Sincerely,
Ladar Levison
Owner and Operator, Lavabit LLC

Defending the constitution is expensive! Help us by donating to the Lavabit Legal Defense Fund [here].
Lavabit: “so secure that even our administrators can’t read your e-mail”

But they could, if they wanted to…

“Basically we generate public and private keys for the user and then encrypt the private key using a derivative of the plain text password. We then encrypt user messages using their public key before writing them to disk.”

“Because we need the plain text password to decrypt a user’s private key, we don’t support secure password authentication. We decided to support SSL instead (which encrypts everything; not just the password).”

http://highscalability.com/blog/2013/8/13/in-memoriam-lavabit-architecture-creating-a-scalable-email-s.html
I don’t like SPAM!
Spam lifecycle

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

Message content
   Evade anti-spam filters: content obfuscation
   V1agra, Via'gra, Vi@graa, vi*gra, Viagra

Spam email delivery
   Webmail accounts (sweatshops, stolen)
   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
- DNSBLs: domains of known spammers, open relays, zombie machines, hosts that shouldn’t be sending emails (e.g., ISPs DHCP pools), …

Honeypots

Outbound filtering (block port 25)

Email authentication
SPF: Origin Authentication

SMTP allows anyone to send an email with an arbitrary “From” address

Sender Policy Framework

DNS TXT record with hosts that are allowed to send email from the domain

Receiving SMTP servers compare IP address that attempts to send an email with allowed addresses of the domain(s) provided in the HELO and MAIL FROM commands

Helps to block spam at it source

mikepo@styx:~> dig google.com TXT
;; ANSWER SECTION:
google.com.  3600 IN TXT "v=spf1 include:_spf.google.com ip4:216.73.93.70/31 ip4:216.73.93.72/31 ~all"
DKIM: Email Validation

DomainKeys Identified Mail: digitally sign some email headers and message body

Allows the recipient to verify that
  The email is sent from the domain it claims to be sent from
  It has not been tampered with

Domain’s public key is stored in a DNS TXT record

X-Google-DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=1e100.net; s=20130820; h=x-gm-message-state:mime-version:date:message-id:subject:from:to :content-type; bh=0BSnrwLTQ7KblIwINxoPJN40a/K5PZCIV8atL6a1Dvg=; b=Nch9yEorgibAjkh90ukDL6SU0FYn70qP6AMsWFpLO+W3iroMoVdKIjKk8Cv6Gc1TW ...

mikepo@styx:~> dig 20130820._domainkey.1e100.net TXT
;; ANSWER SECTION:
20130820._domainkey.1e100.net. 86400 IN TXT "k=rsa\;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnOv6+Txyz+SEc7mT719QQtOj6g2MjpErYUGVrRGGc7f5rmE1cRP1lhwx8PVoHOiuRzyok7IqvJvAub9kk9fBoE9u ...

**SPF + DKIM = DMARC**

Domain-based Message Authentication, Reporting & Conformance (DMARC)

Standardizes how email receivers perform email authentication using SPF and DKIM

Tells receivers what to do if neither of those authentication methods passes – such as junk or reject the message

DMARC policies are published as DNS TXT records

```
mikepo@styx:~> dig _dmarc.google.com TXT
;; ANSWER SECTION:
_dmarc.google.com. 600 IN TXT "v=DMARC1\; p=quarantine\; rua=mailto:mailauth-reports@google.com"
```
DMARC Email Authentication Process

http://dmarc.org/overview/
TorrentLocker spam has DMARC enabled

Use of email authentication technique unlikely to bring any advantage.

Last week, Trend Micro researcher Jon Oliver (who presented a paper on Twitter abuse at VB2014) wrote an interesting blog post about a spam campaign that was spreading the 'TorrentLocker' ransomware and which, unusually, was using DMARC.

TorrentLocker is one of the most prominent families of encryption ransomware — a worryingly successful kind of malware that first appeared two years ago. The malware initially implemented its cryptography rather poorly, but has since become one of the most successful of its kind.

DMARC is an email technology that builds on both SPF and DKIM. Both these technologies allow a domain owner to take some responsibility for the emails sent from their domain: SPF by listing those IP addresses used to send email; DKIM by digitally signing the emails.

DMARC adds to SPF and DKIM a mechanism that allows a domain owner to advise senders what to do about...
SPF, DKIM, DMARC

SPF validates MAIL FROM vs. its source server
   “Envelope” information

DKIM validates the “From:” message header
   Plus other message headers and mail body

Not effective against spammers who
   Use their own domains
   Use legitimate email services, such as webmail
   Pretend to be another user on the same domain

Good for whitelisting and verifying email from trusted sources (.gov, banks, …)

   Besides spam, we also care about phishing…
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”

Deception

   From: info@paypa1.com
   http://www.bankofamerica.com.attacker.net/
The Root of the Problem...

Subject: Important! You must change your XXXXXXX password
Date: XXXXXXXXXX
From: XXXXXXXXXXXXX

[This is not a spam mail, this email is from me, XXXXXXXXXXXXX]

Member of XXXXXXXXXX Department,

PLEASE CHANGE YOUR XXXXXX PASSWORD!

We just upgraded the security of XXXXXX. Your current password is no longer working. You must change your password if you want to log into XXXXXX. [...] 

To change your XXXXX password: 
http://XXXXXXXXX.XXX -> forgot your password -> follow the instructions
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 asking for password

Anti-phishing filters, tools, …
Spear Phishing

Well-prepared, personalized, convincing messages targeted to particular individuals
  Seemingly coming from trusted colleagues
  May contain *personal* information about their target

Highly effective, used extensively in targeted attacks
  Document attachments exploiting 0day vulnerabilities

Many recent incidents
Maybe rethink email altogether?

Recent secure messaging apps offer further benefits

EFF’s Secure Messaging Scorecard

- Encrypted in transit?
- Encrypted so the provider can’t read it?
- Can you verify contacts’ identities?
- Are past communications secure if your keys are stolen?
- Is the code open to independent review?
- Is security design properly documented?
- Has there been any recent code audit?

Many encouraging efforts

OTR, TextSecure, Pond, …

https://www.eff.org/secure-messaging-scorecard