How Mail Uses the DNS

John Levine | ICANN 86 Sevilla | 

Mail is very old

- RFC 822 predates the DNS
  - Names came from `HOSTS.TXT`
- RFC 974 used the DNS to find mail hosts
- All traffic unencrypted, just like telnet, FTP, …
  - Added a lot of security stuff since then
- Updates kept backward compatibility, often pretty ugly
- New features added as options to existing spec

Store and forward and relaying

- Mail is sent in the background, no user involvement
- Store and forward: mail usually goes through multiple hops
- Each hop allows diversion and inspection
  - Often a feature, e.g., force outgoing mail through an ISP gateway or spam filter
- Added STARTTLS
  - But clients don’t normally check the cert so diversion still possible
  - Between hops message in the clear on each relay host

Your basic mail flow

Your basic mail flow

SRV
A/AAAA
(TLSA)

Submission

Mail Server

MX
A/AAAA
(TLSA)
MTA-STS A/AAAA

SMTP

DNSBL A/TXT
SPF TXT
DKIM TXT
DMARC TXT

Mail Server

POP/IMAP

A/AAAA
(TLSA)

What the lookups are for

- Sending mail client
  - Find and connect to the sending server
- Sending server:
  - Find next server (MX/A/AAAA)
  - Check that it's the server you expect (TLSA, MTA-STS)
- Recipient server:
  - Is sender malicious? (DNSBLs)
  - Is sender authorized to send for this domain? (SPF)
  - Has sender signed the message? (DKIM)
  - Adequately authenticated? (DMARC)
  - Message contains bad stuff? (other DNSBLs)
- Recipient mail client
  - Find server, retrieve mail

Set up MUA

Submission

Mail Server

- Too many ways to set up the MUA
  - Manual
  - A/AAAA
    `https://autodiscover.domain/...`
  - SRV per service
- SRV lookups for various options
  - `_submission._tcp.domain SRV`
  - `_submissions._tcp.domain SRV`
  - `_imap._tcp.domain SRV`
  - `_imaps._tcp.domain SRV`
  - `_pop3._tcp.domain SRV`
  - `_pop3s._tcp.domain SRV`

Submission mail flow

A/AAAA lookup for host

TLSA lookup for cert

```
… TLS negotiation …
S: 220 server.example ESMTP
C: EHLO somehost
S: 250-server.example
   250-8BITMIME
   250 AUTH LOGIN PLAIN
C: AUTH PLAIN AHByaW50ZXIAZnV6enk=
S: 235 2.7.0 Auth succeeded.
```

```
C: MAIL FROM:<test@examp1e.com>
S: 250 2.1.0 Sender accepted.
C: RCPT TO:<bob@example.net>
S: 250 2.1.5 Recipient accepted.
C: DATA
S: 354 Send your message
C: … message headers and body …
C: .
S: 250 2.6.0 Accepted message
C: QUIT
S: 221 2.0.0 Good bye.
```

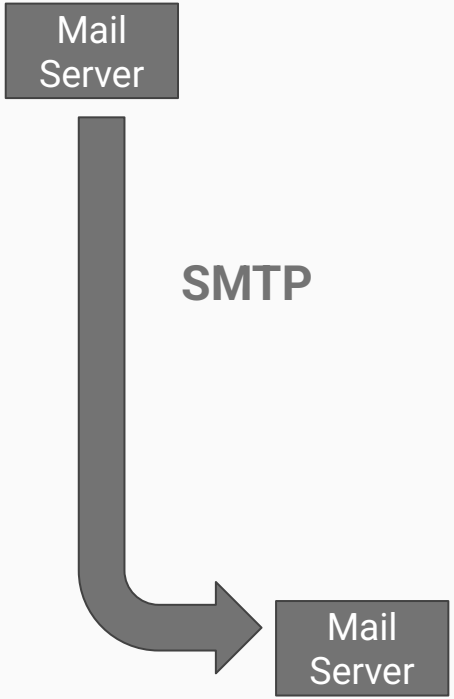
MUA to sending server

Submission

Mail Server

- Client mail program does A/AAAA hostname lookup of submission server
- TLSA `_465._tcp.`*`servername`* to check server SSL certificate

Sending to receiving server



- MX on recipient domain
  - To find recipient server(s)
- TLSA and MTA-STS checks to see if TLS expected

Finding the recipient server

```
icann.org. MX 10 pechora1.icann.org.
icann.org. MX 10 pechora6.icann.org.
icann.org. MX 10 pechora7.icann.org.
icann.org. MX 10 pechora8.icann.org.

pechora1.icann.org. A   192.0.33.71
pechora1.icann.org. AAAA
     2620:0:2d0:201::1:71
```

- MX provides a level of direction
- Has priority (lower is better) and hostname
- If no MX but A/AAAA
  - Fake `domain MX 0 domain`
- Large systems usually have multiple MX for round robin
- Or multiple A/AAAA, or both

Relay mail flow

C: MX, A/AAAA lookup for host

C: TLSA lookup for cert or MTA-STS

```
S: 220 relay.example ESMTP
C: EHLO sender.example
S: 250-relay.example
   250-8BITMIME
   250 STARTTLS
C: STARTTLS
S: 220 2.0.0 Ready to start TLS
… TLS negotiation …
C: EHLO server.example
S: 250-relay.example
C: MAIL FROM:<test@examp1e.com>
S: 250 2.1.0 Sender accepted.
```

S: A/AAAA sending hostname

S: MX/ A/AAAA sending domain

```
C: RCPT TO:<bob@example.net>
S: 250 2.1.5 Recipient accepted.
C: DATA
S: 354 Send your message
C: … message headers and body …
C: .
S: 250 2.6.0 Accepted message
C: QUIT
S: 221 2.0.0 Goodbye.
```

Is this the right server? TLSA

```
S: 220 relay.example ESMTP
C: EHLO sender.example
S: 250-relay.example
   250 STARTTLS
C: STARTTLS
S: 220 2.0.0 Ready to start TLS
... TLS negotiation ...
```

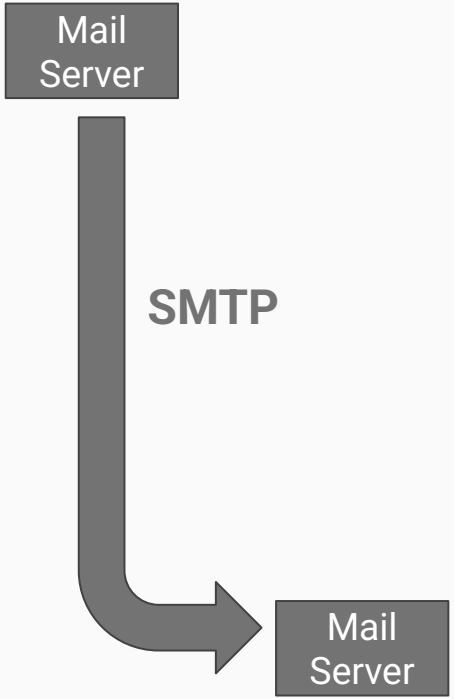
- Mail servers used self-signed certs
  - Clients didn't try to verify them
  - And MITM was still easy
- Now signed certs, can publish TLS policy
- TLSA check MX name
  - Fail if no STARTTLS, or wrong cert
- This is **entirely voluntary**
- Domains that publish TLSA accept plain text mail

Is this the right server? MTA-STS

- A guy at Google doesn't like DNSSEC
- MTA-STS was the result
- TXT record says there's a policy
  `_mta-sts.examp1e.com. TXT "v=STSv1; id=m142;"`
- Fetch it from URL
  `https://mta-sts.examp1e.com/.well-known/mta-sts.txt`
- File contains list of MXes, and flags
- Also **entirely voluntary**
- Domains that publish MTA-STS accept plain text mail

```
S: 220 relay.example ESMTP
C: EHLO sender.example
S: 250-relay.example
   250 STARTTLS
C: STARTTLS
S: 220 2.0.0 Ready for TLS
... TLS negotiation ...
```

Receiving server checks



- A/TXT DNSBLs on connecting IP
  - Hosts you don't want to hear from
- A/AAAA EHLO hostname, IP must match
  - Weeds out a lot of bots
- MX/A/AAAA MAIL FROM hostname
  - Weeds out a lot of spam
- SPF TXT lookup on MAIL FROM hostname

Checks about the sending IP

- DNSBL check

```
2.0.0.127.nlpnxs3kb2evd2dhnttmcae.zen.dq.spamhaus.net. IN A
;; ANSWER SECTION:
2.0.0.127.nlpnxs3kb2evd2dhnttmcae.zen.dq.spamhaus.net. 1 IN A 127.0.0.2
2.0.0.127.nlpnxs3kb2evd2dhnttmcae.zen.dq.spamhaus.net. 1 IN A 127.0.0.10
2.0.0.127.nlpnxs3kb2evd2dhnttmcae.zen.dq.spamhaus.net. 1 IN A 127.0.0.4
```

- No answer is good, 127/x values indicate bad
- Generally freemium model

```
2.0.0.127.nlpnxs3kb2evd2dhnttmcae.zen.dq.spamhaus.net.
```

Checks about the sending hostname

- EHLO name should match f/rDNS

  ```
  EHLO ppa3.lax.icann.org
  ```

  ```
  ppa3.lax.icann.org. A 192.0.33.78
  78.33.0.192.in-addr.arpa. PTR ppa3.lax.icann.org.
  ```

- Not strictly required by RFC but you'll be sorry if you don't

Checks about the sending domain

```
MAIL FROM:<steve@icann.org>
```

- Check that domain exists, MX or A/AAAA

  ```
  icann.org. MX  10 pechora1.icann.org.
  ```

- And it's not in DNSBLs

  ```
  icann.org.dbl.spamhaus.net  NXDOMAIN
  ```

SPF on the sending domain

```
MAIL FROM:<steve@icann.org>
```

- Check SPF record allows sending IP

```
icann.org. TXT    "v=spf1 ip4:192.0.32.0/20 ip4:199.91.192.0/21
ip4:64.78.40.0/27 ip4:162.216.194.0/27 ip4:64.78.33.5/32
ip4:64.78.33.6/31 ip4:64.78.48.205/32 ip4:64.78.48.206/31
ip6:2620:0:2d0::0/48 ip6:2620:0:2830::0/48 ip6:2620:0:2ed0::0/48
include:salesforce.icann.org -all"

salesforce.icann.org. TXT "v=spf1 include:_spf.salesforce.com -all"

_spf.salesforce.com. TXT "v=spf1 exists:%{i}._spf.mta.salesforce.com -all"
```

Body checks

DKIM

- Sign hash of headers and body
- Signing domain and selector
  - Selector only for key management
- Validation key in the DNS

```
mailman._domainkey.icann.org. TXT
 "v=DKIM1; h=sha256; k=rsa;
s=email; p=MIIBIjANBg…DAQAB"
```

- Means “I handled this mail”
  - Not I wrote it, or it’s not spam, or …
  - Each domain can sign along the way
- Receivers use domain reputation

```
Received: from icann.org (mm4.lax.icann.org
 [10.32.0.180]) by smtp.lax.icann.org
 (Postfix) with ESMTPS id ABD9DE062E;
  Wed,  3 Jun 2026 20:22:38 +0000 (UTC)

DKIM-Signature: v=1; a=rsa-sha256;
 c=relaxed/simple; d=icann.org; s=mailman;
 t=1780518158; bh=lvXYJv9GfG2LNE…cv/A=;
 h=Date:References:In-Reply-To:To:Cc:Subject:
 From:Reply-To:From; b=c1Qnotuf262j…g9qC

Received: from [172.18.0.8] by icann.org
  (Postfix) with ESMTP id 6336B3191769;
  Wed,  3 Jun 2026 20:22:38 +0000 (UTC)
```

DMARC

`From: bob@sales.examp1e.com`

- Domain owner can say all my mail has my SPF or DKIM
  - Advice what to do otherwise
- Tree walk to“organizational” domain
- Aggregate reports
  - XML of what came from where
  - Many services to analyze it
- Works OK, wrecks mailing lists

- Policy record in DNS

```
_dmarc.sales.examp1e.com TXT NOERROR

_dmarc.examp1e.com TXT v=DMARC1; p=none;
rua=mailto:tutnr7vw@ag.us.dmarcian.com
```

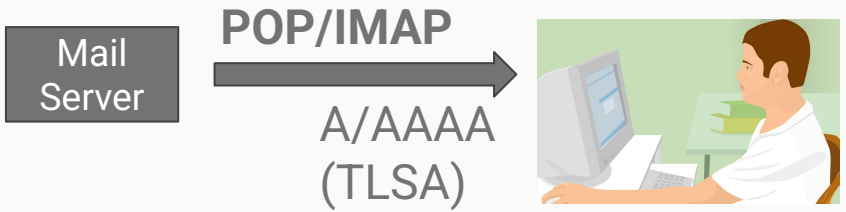
ARC (don’t do this)

- Intended to fix DMARC damage
- Chain shows history of auth
- Easy to fake
- Surprisingly easy to screw up
- IETF is deprecating
- DKIM2 will try to do it right

```
ARC-Seal: i=1; a=rsa-sha256; cv=none;
d=lists.iecc.com; s=775a.6a20c240.k2606;
t=1780531776;b=gaiKQ5HHDygJYoFvHJ0fj4EOWNcK1p
XXtRgsdHV/deJ54V8PA3yIGAGuujuy+NpwOALOjxY7OcY
SfXK3+S+W+ng4...pGwrA==
ARC-Message-Signature: i=1; a=rsa-sha256;
c=relaxed/relaxed; d=lists.iecc.com;
h=date:from:to:message-id:references:   ;
s=75a.6a20c240.k2606; bh=Tqeb4aDEL...uxO3Xt4=;
b=doMH...Gnfg==
ARC-Authentication-Results: i=1; iecc.com;
arc=none; spf=pass; dkim=pass
header.d=yahooinc.com header.s=pps1
header.a=rsa-sha256 header.b="W+qkPwkq";
dkim=pass policy.dmarc=reject
```

Mail pickup

- Client uses POP or IMAP to retrieve mail from server
- Same setup as before, using POP/IMAP rather than submission
- A/AAAA to find host, perhaps TLSA to check certificate



Odds and Ends

More than one MX

- A domain can have several MX with different priorities:

```
examp1e.com.  MX 10 mail1.examp1e.com.
              MX 10 mail2.examp1e.com.
              MX 20 mail.backup.example.
```

- Multiple at same priority and/or well connected backup for flaky one
- Backups much less common now
  - Spam filtering on backup is much harder
- Multiple at same priority common for large systems
  - Yahoo has three MXs, each with six As

Hosts that don’t handle mail

- Any host with an A/AAAA record is potentially a mail server
  - RFC 974 said use WKS to check, but nobody did
- Misdirected mail could take a week to bounce
  - Repeated connection failures and retries
- We fixed it with Null MX in 2015

  ```
  www.example.com. IN MX 0 .
  ```
- Special case MX record that didn’t mean anything before

Summary

A lot of DNS

- Each stage uses DNS to find and validate the next stage
- Many security hacks layered on
  - STARTTLS (TLSA/MTA-STS), DKIM, SPF, DMARC, …
- If we had designed this rather than evolving over 40 years, it’d be a lot cleaner

How Mail Uses the DNS

John Levine | ICANN 86 Sevilla |