Advanced Topics of Mail Service

Deal with Malicious Mails in the Real World

tsaimh (2024, CC-BY)
lwhsu (2020-2023, CC-BY)
? (?-2019)

國立陽明交通大學資工系資訊中心
Information Technology Center of Department of Computer Science, NYCU

The Source of the Term “Spam”

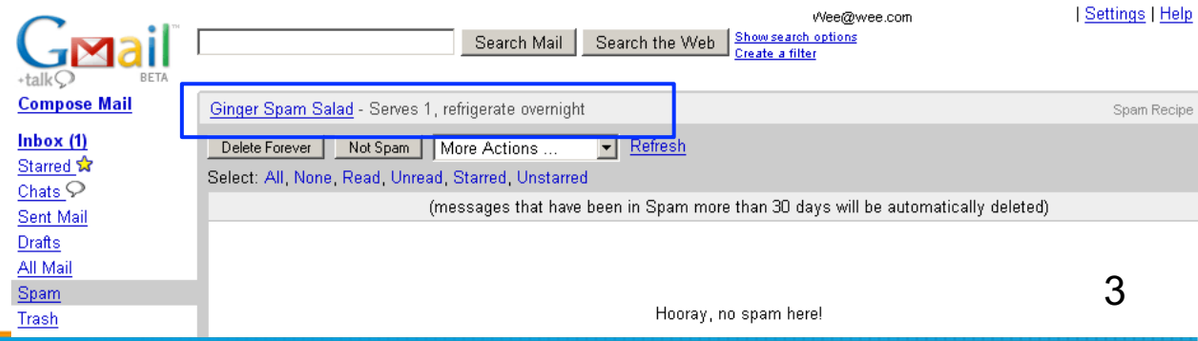
- SPAM is a brand of processed canned pork and ham made by Hormel Foods Corp..
- It was introduced in 1937 and gained popularity worldwide after its use during World War II.
- Ken Daigneau, the brother of a company executive, won a $100 prize in 1937 in a competition to name the new item.
- The meaning of the name is known by only a small circle of former Hormel Foods executives, but a popular belief is that the name is a contraction of "spiced ham"
- The billionth can of Spam was sold in 1959, and the eight billionth can was sold in 2012.

Source: https://en.wikipedia.org/wiki/Spam_(food)

The Source of the Term “Spam” (cont.)

- In 1970, spam was featured in episode 12 of series 2 of Monty Python’s Flying Circus (the episode is titled “Spam”).
- With a chorus of Vikings boisterously singing a song– "Lovely Spam, Wonderful Spam", which, by the 1990s, led to "Spam" being adopted as a term for unsolicited electronic messages, especially spam email, because in the song, the repeated singing of the word "Spam" drowns out all other communication.

https://www.youtube.com/watch?v=anwy2MPT5RE



Nature of Spam

- Spam – Simultaneously Posted Advertising Message
  - UBE – Unsolicited Bulk Email
  - UCE – Unsolicited Commercial Email
- Common Features of Spam
  - There is no relationship between receiver and
    - Sender
    - Message content
  - Opt-out instruction
  - Conceal trail
    - False return address
    - Forged header information
  - Use misconfigured mail system to be an accomplice
  - Circumvent spam filters either encode message or insert random letters

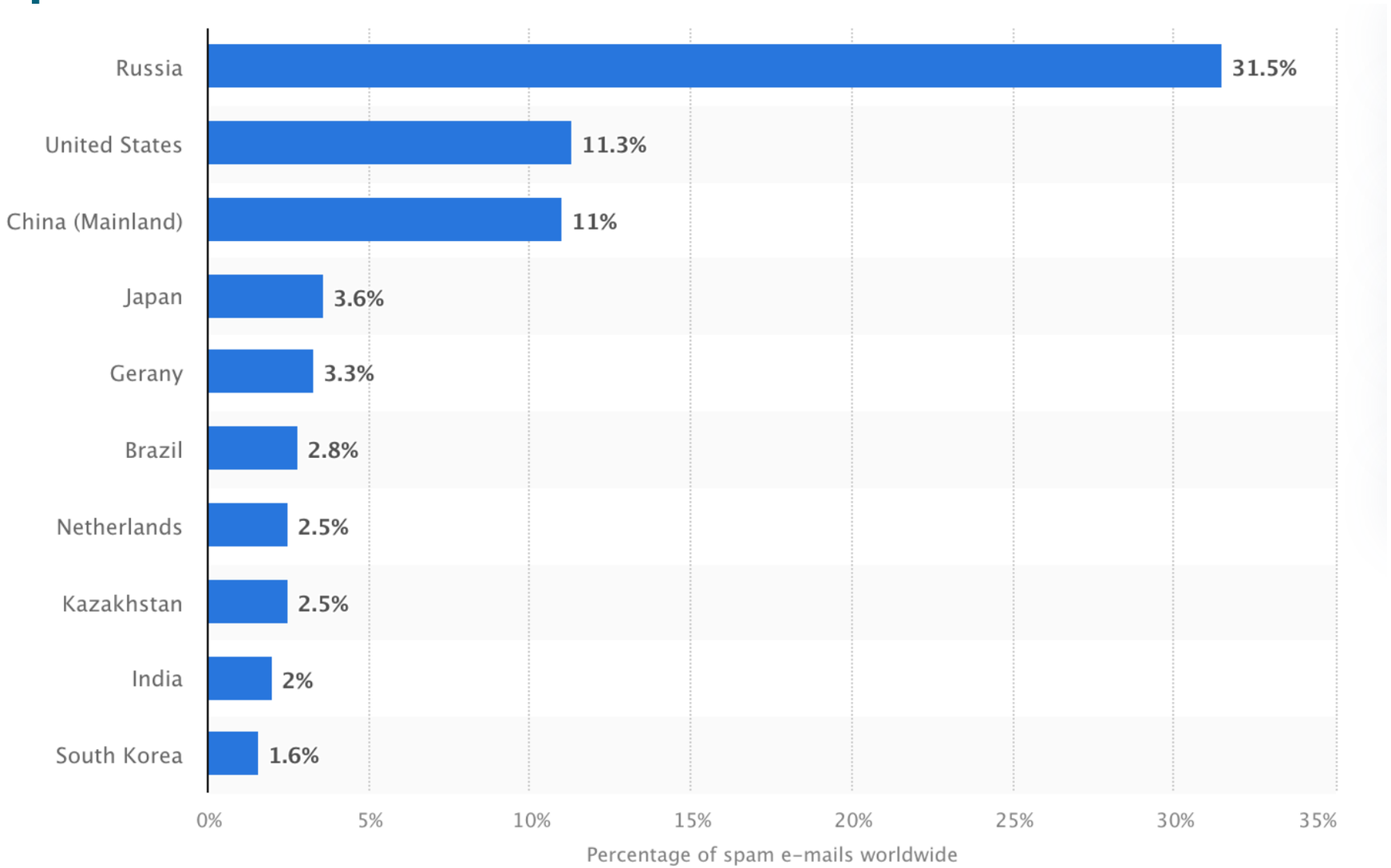
Problems of Spam

- Cost
  - Waste bandwidth and disk space
  - DoS like side-effect
  - Waste time
  - False deletion
  - Bounce messages of nonexistent users
    - Nonexistent return address
    - Forged victim return address
- Detection
  - Aggressive spam policy may cause high false positive

SPAM detection

- SPAM vs. non-SPAM
  - Mail sent by spammer vs. non-spammer
- Problem of SPAM mail
  - About 90% of E-mail are SPAM! Useless for mankind!
- SPAM detection
  - Client-based detection
    - ➢spammer detection
    - ➢cost-effective, which can easily reach over 95% accuracy
  - Content-based detection
    - ➢spam detection
    - ➢costly with less than 90% accuracy, needing training and computation
  - Who is the winner? Client-based? Content-based? (or Spammer?)
  - Endless war between the administrators and spammers.

Leading Countries of Origin for Spam E-mails in 2023



Source: https://www.statista.com/statistics/263086/countries-of-origin-of-spam/

Anti–SPAM – Client – Based Detection

- Client-blocking
  - Check their IP address, hostnames, email address, and/or behavior when the client connect to send a message
  - Problems
    - IP address, hostname, email address are forged
    - Innocent victim open relay host
- Techniques
  - DNSBL/WL (DNS Blacklists and Whitelists)
    - RFC 5782
  - Greylisting
  - SPF – Sender Policy Framework
  - DomainKeys/DKIM
  - Sender ID
  - …

Anti–SPAM – Content – Based Detection

- Spam patterns in message header/body
  - Encrypted
  - Encoded
- Techniques
  - Pattern detection
  - Bayesian spam filtering
  - …
- Difficulties
  - Embed HTML codes within words of their message to break up phrases
  - Randomly inserted words
  - Slower and resource consumption

Anti–SPAM – Action

- When you suspect that a mail is spam, you can:
  - Reject
    - immediately during the SMTP conversation
    - directly discard the mail without notifying someone else
  - Save spam into a suspected spam repository
  - Label spam and deliver it with some kind of spam tag
  - Ex:
    - X-Spam-Status: Yes, hits=18.694 tagged_above=3 required=6.3
    - X-Spam-Level: **********
    - X-Spam-Flag: YES

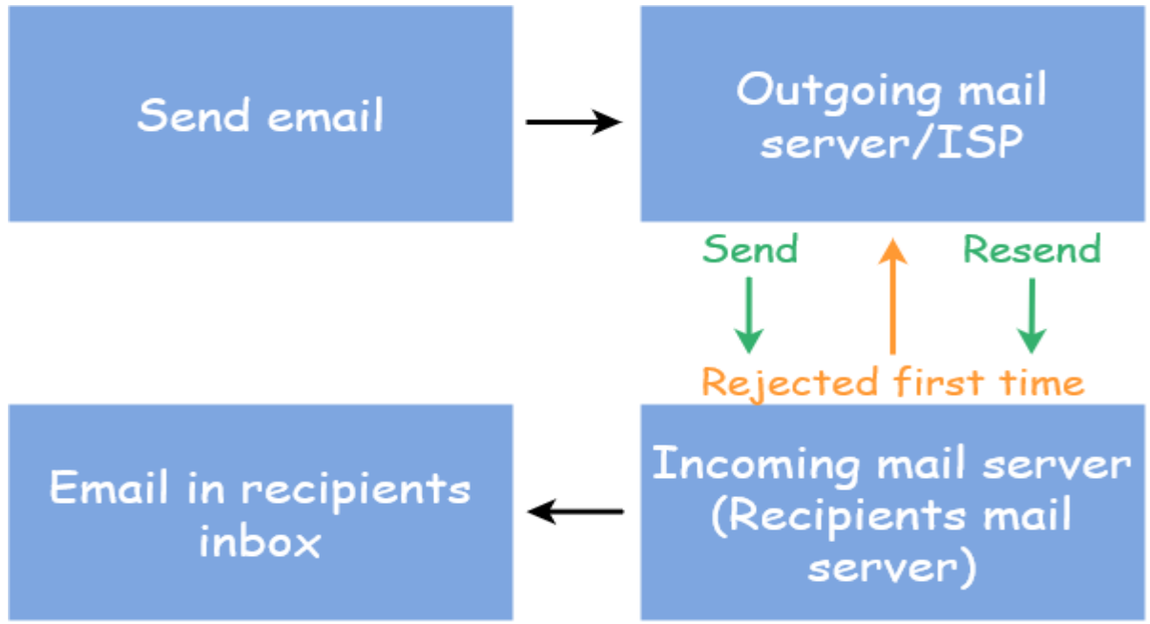
Client – based Detections

- Fight with spammers:
  - DNSBL/WL
    - DNS-based blacklist/whitelist for suspected/trusted senders(IP address)
  - Greylisting
    - client-based method that can stop mail coming from some spamming programs
  - SPF (Sender Policy Framework; defined in RFC 4408)
    - A client-based method to detect whether a client is authorized or not
    - Sender ID (obsoleted; although defined in RFC 4406)
      - NOT the new SPF
      - http://www.open-spf.org/SPF_vs_Sender_ID/

DNSxL

- What DNSBL/WL maintainers do
  - Suppose cs.nctu.edu.tw has a DNSxL database
    - DNSBL Domain "dnsbl.cs.nctu.edu.tw"
  - If 140.112.23.118 is detected as open relay
    - Add resource record 118.23.112.140.dnsbl.cs.nctu.edu.tw
  - When we receive a connection from 140.112.23.118
    - DNS query for 118.23.112.140.dnsbl.cs.nctu.edu.tw
      - A 127.0.0.2 (**SHOULD** in 127.0.0.0/8)
        - https://www.spamhaus.org/zen/
        - TXT Reason
  - Right-Hand Sde Blacklist (RHSBL)
- Using DNSBL
  - Review their service options and policies carefully
  - https://www.dnsbl.info/dnsbl-database-check.php

Greylisting (1/2)



- https://greylisting.org
- Client-based (receiver) method that can stop (slowdown) some spammers
- Different behaviors against SMTP response codes

Response Codes	2xx	4xx	5xx
Normal MTA	Success	Retry later	Give-up
Most Spammers	Success	Ignore and send another	Give-up

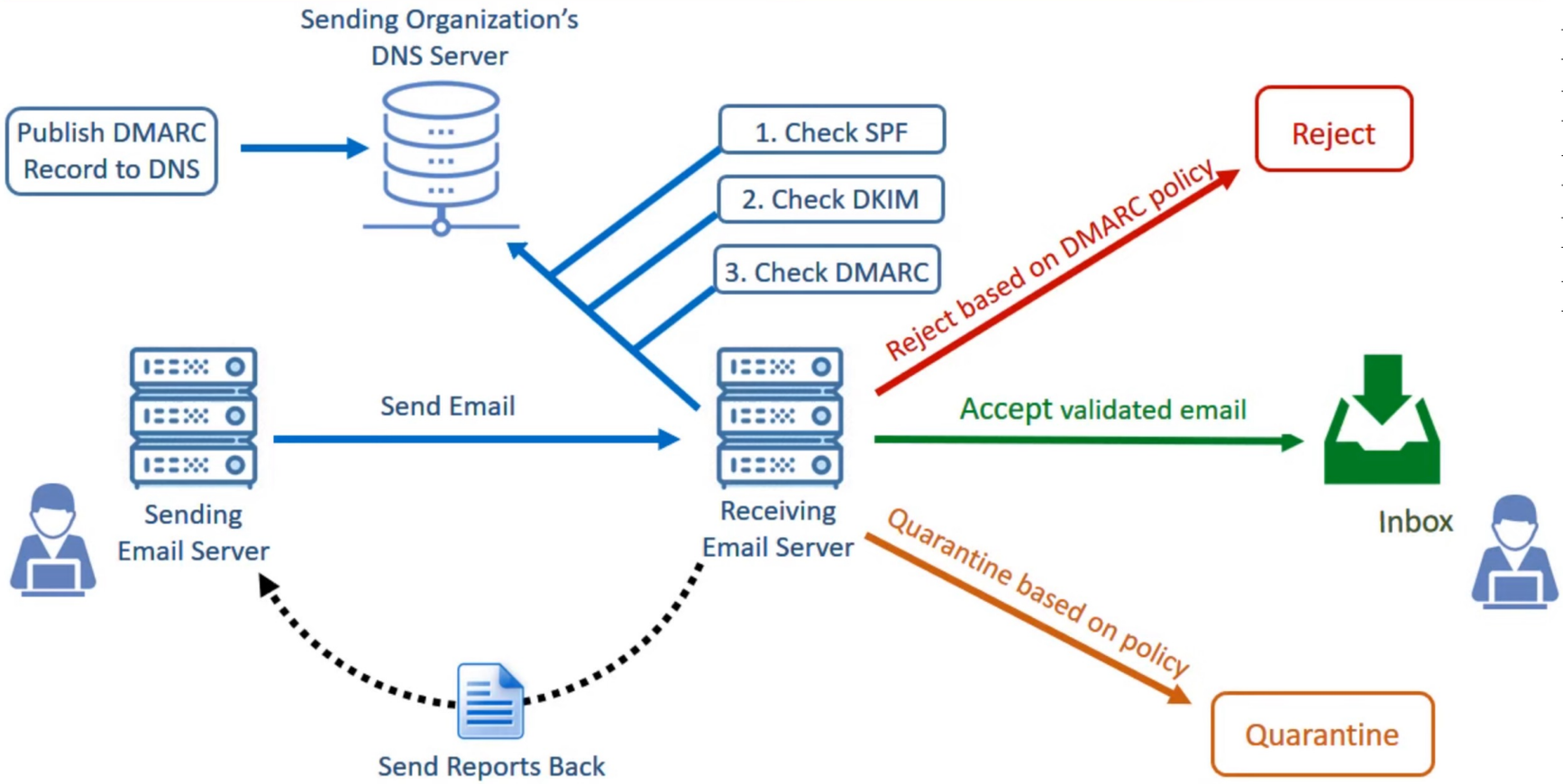
- While spammers prefer to send mail to other recipients rather than keeping log and retrying later, MTAs have the responsibility of retrying a deferred mail (in 10-30 mins)
  - ➢ Combine with other spam mitigations and network security features

Greylisting (2/2)

- Idea of greylisting:
  - Taking use of 4xx SMTP response code to stop steps of spamming programs
- Steps:
  - Pair (recipient, client-ip)
  - Reply a 4xx code for the first coming of every (recipient, client-ip) pair.
  - Allow retrial of this mail after a period of time (usually 5~20 mins)
    - Suitable waiting time will make the spamming programs giving up this mail
- Limitation
  - Can NOT detect “open relay” mail servers

SPF, DKIM and DMARC

SPF, DKIM and DMARC



SPF

RFC 4408 (2006, obsoleted)

RFC 7208 (2014)

DKIM

RFC 4871 (2007, obsoleted)

RFC 6376 (2011)

DMARC

RFC 7489 (2015)

Source: https://www.youtube.com/watch?v=c9fLp5uIxp8

Sender Policy Framework (SPF)

- A client-based method to detect whether a client is authorized or not
- http://www.open-spf.org/
  - RFC 4408

Sender Policy Framework (SPF) – Is following mail questionable?

```
Delivered-To: lwhsu.gmail@gmail.com
Received: by 10.204.137.3 with SMTP id u3cs64867bkt;
        Sat, 21 May 2011 13:19:49 -0700 (PDT)
Received: by 10.68.58.38 with SMTP id n6mr1407584pbq.5.1306009188186;
        Sat, 21 May 2011 13:19:48 -0700 (PDT)
Return-Path: <lwhsu@cs.nctu.edu.tw>
Received: from zfs.cs.nctu.edu.tw (zfs.cs.nctu.edu.tw [140.113.17.215])
        by mx.google.com with ESMTP id
a2si4001228pbs.91.2011.05.21.13.19.46;
        Sat, 21 May 2011 13:19:46 -0700 (PDT)
Received: from zfs.cs.nctu.edu.tw (localhost [127.0.0.1])
      by zfs.cs.nctu.edu.tw (Postfix) with ESMTP id 50E2A4ABC5
      for <lwhsu.gmail@gmail.com>; Sun, 22 May 2011 04:16:08 +0800 (CST)
Date: Sun, 22 May 2011 04:12:57 +0800
From: Li-Wen Hsu <lwhsu@cs.nctu.edu.tw>
To: Li-Wen Hsu <lwhsu.gmail@gmail.com>
Subject: test
Message-ID: <20110521201257.GA58179@zfs.cs.nctu.edu.tw>

this is a test
```

Sender Policy Framework (SPF) – SMTP trace

```
zfs-$ telnet zfs.cs.nctu.edu.tw 25
220 zfs.cs.nctu.edu.tw ESMTP Postfix
helo zfs.cs.nctu.edu.tw
250 zfs.cs.nctu.edu.tw
mail from: <lwhsu@cs.nctu.edu.tw>
250 2.1.0 Ok
rcpt to: <lwhsu.gmail@gmail.com>
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Date: Sun, 22 May 2011 04:12:57 +0800
From: Li-Wen Hsu <lwhsu@cs.nctu.edu.tw>
To: Li-Wen Hsu <lwhsu.gmail@gamil.com>
Subject: test
Message-ID: <20110521201257.GA58179@zfs.cs.nctu.edu.tw>

this is a test
.
250 2.0.0 Ok: queued as 50E2A4ABC5
```

Sender Policy Framework (SPF) – With SPF detection

```
Delivered-To: lwhsu.gmail@gmail.com
Received: by 10.204.137.3 with SMTP id u3cs64867bkt;
        Sat, 21 May 2011 13:19:49 -0700 (PDT)
Received: by 10.68.58.38 with SMTP id n6mr1407584pbq.5.1306009188186;
        Sat, 21 May 2011 13:19:48 -0700 (PDT)
Return-Path: <lwhsu@cs.nctu.edu.tw>
Received: from zfs.cs.nctu.edu.tw (zfs.cs.nctu.edu.tw [140.113.17.215])
        by mx.google.com with ESMTP id a2si4001228pbs.91.2011.05.21.13.19.46;
        Sat, 21 May 2011 13:19:46 -0700 (PDT)
Received-SPF: softfail (google.com: domain of transitioning
lwhsu@cs.nctu.edu.tw does not designate 140.113.17.215 as permitted sender)
client-ip=140.113.17.215;
Authentication-Results: mx.google.com; spf=softfail (google.com: domain of
transitioning lwhsu@cs.nctu.edu.tw does not designate 140.113.17.215 as
permitted sender) smtp.mail=lwhsu@cs.nctu.edu.tw
Received: from zfs.cs.nctu.edu.tw (localhost [127.0.0.1])
      by zfs.cs.nctu.edu.tw (Postfix) with ESMTP id 50E2A4ABC5
      for <lwhsu.gmail@gmail.com>; Sun, 22 May 2011 04:16:08 +0800 (CST)
Date: Sun, 22 May 2011 04:12:57 +0800
From: Li-Wen Hsu <lwhsu@cs.nctu.edu.tw>
To: Li-Wen Hsu <lwhsu.gmail@gmail.com>
```

Sender Policy Framework (SPF) – The idea

- For a domain administrator, they can claim which mail servers will be used in his environment
  - Ex. For cs.nctu.edu.tw, {csmailer,csmailgate,csmail}.cs.nctu.edu.tw are the authorized mail servers
    - Mail out from these servers are authorized mail (under control of administrator)
    - Other mail might be forged and have higher probability to be SPAMs
- SPF technique specifies all possible outgoing mail clients in the TXT/SPF record of DNS service to claim the authorized mail servers
- When destination MTA receives a mail, it will check the client IP:
  - For a mail out from authorized servers, it should be safe.
  - For a mail out from unauthorized servers, it might be forged.

SPF Record Syntax – Mechanisms (1/2)

TXT/SPF record: v=spf1 [qualifier][mechanism]

- all
  - Always matches
  - Usually at the end of the SPF record
- ip4 **(NOT ipv4)**
  - ip4: <ip4-address>
  - ip4: <ip4-network>/<prefix-length>
- ip6 **(NOT ipv6)**
  - ip6:<ip6-address>
  - ip6:<ip6-network>/<prefix-length>
- a
  - a
  - a/<prefix-length>
  - a:<domain>
  - a:<domain>/<prefix-length>

```
% dig +noall +answer txt nycu.edu.tw
nycu.edu.tw.            604787 IN       TXT
"v=spf1 ip4:140.113.2.64/26 ip4:211.76.241.6
ip4:140.113.98.162 ip4:140.113.98.175
ip4:140.113.9.141 ip4:140.113.7.200
include:_spf.google.com ~all"
```

The content of this page and following are from http://www.open-spf.org/SPF_Record_Syntax

SPF Record Syntax – Mechanisms (2/2)

- mx
  - mx
  - mx/<prefix-length>
  - mx:<domain>
  - mx:<domain>/<prefix-length>
- ptr
  - ptr
  - ptr:<domain>
- exists
  - exists:<domain>
    - Does A record exist?
- include
  - include:<domain>
    - Warning: If the domain does not have a valid SPF record, the result is a **permanent error**. Some mail receivers will *reject* based on a **PermError**

```
% dig +noall +answer txt nycu.edu.tw
nycu.edu.tw.            604787 IN       TXT
"v=spf1 ip4:140.113.2.64/26 ip4:211.76.241.6
ip4:140.113.98.162 ip4:140.113.98.175
ip4:140.113.9.141 ip4:140.113.7.200
include:_spf.google.com ~all”
% dig +noall +answer txt _spf.google.com
_spf.google.com.        30      IN      TXT
"v=spf1 include:_netblocks.google.com
include:_netblocks2.google.com
include:_netblocks3.google.com ~all"
```

SPF Record Syntax - Qualifiers & Evaluation

- Qualifiers
  - + Pass (default qualifier)
  - - Fail
  - ~ SoftFail
  - ? Neutral
- Evaluation
  - Mechanisms are evaluated in order: (first-matching)
    - ➢If a mechanism results in a hit, its qualifier value is used
    - ➢If no mechanism or modifier matches, the default result is "Neutral"
  - Ex.
    - "v=spf1 +a +mx -all"
    - "v=spf1 a mx -all"

SPF Record Syntax - Evaluation Results

Result	Explanation	Intended action
Pass	The SPF record designates the host to be allowed to send	Accept
Fail	The SPF record has designated the host as NOT being allowed to send	Reject
SoftFail	The SPF record has designated the host as NOT being allowed to send but is in transition	Accept but mark
Neutral	The SPF record specifies explicitly that nothing can be said about validity	Accept
None	The domain does not have an SPF record or the SPF record does not evaluate to a result	Accept
PermError	A permanent error has occurred (eg. Badly formatted SPF record)	Unspecified
TempError	A transient error has occurred	Accept or reject

SPF Record Syntax – Modifier

- redirect
  - redirect=<domain>
  - The SPF record for domain replace the current record. The macro-expanded domain is also substituted for the current-domain in those look-ups
- exp
  - exp=<domain>
  - If an SMTP receiver rejects a message, it can include an explanation. An SPF publisher can specify the explanation string that senders see. This way, an ISP can direct nonconforming users to a web page that provides further instructions about how to configure SASL
  - The domain is expanded; a TXT lookup is performed. The result of the TXT query is then macro-expanded and shown to the sender. Other macros can be used to provide an customized explanation

Sender Policy Framework (SPF)
– Example of mail from an authorized server

- On bsd2.cs.nctu.edu.tw
  - From: lwhsu@cs.nctu.edu.tw
  - To: lwhsu.gmail@gmail.com

- Related SPF Record:

cs.nctu.edu.tw
"v=spf1 a mx a:csmailer.cs.nctu.edu.tw a:csmailgate.cs.nctu.edu.tw a:csmail.cs.nctu.edu.tw ~all"

Sender Policy Framework (SPF)
– Example of mail from an authorized server

```
Delivered-To: lwhsu.gmail@gmail.com
Received: by 10.90.56.12 with SMTP id e12cs464421aga;
        Sun, 10 May 2009 12:12:00 -0700 (PDT)
Received: by 10.210.91.17 with SMTP id o17mr7881766ebb.3.1241982719273;
        Sun, 10 May 2009 12:11:59 -0700 (PDT)
Return-Path: <lwhsu@cs.nctu.edu.tw>
Received: from csmailer.cs.nctu.edu.tw (csmailer.cs.nctu.edu.tw [140.113.235.130])
        by mx.google.com with ESMTP id 10si4213172eyz.41.2009.05.10.12.11.58;
        Sun, 10 May 2009 12:11:59 -0700 (PDT)
Received-SPF: pass (google.com: domain of lwhsu@cs.nctu.edu.tw
        designates 140.113.235.130 as permitted sender) client-ip=140.113.235.130;
Authentication-Results: mx.google.com; spf=pass (google.com: domain of
        lwhsu@cs.nctu.edu.tw designates 140.113.235.130 as permitted sender)
        smtp.mail=lwhsu@cs.nctu.edu.tw
Received: from bsd2.cs.nctu.edu.tw (bsd2 [140.113.235.132])
        by csmailer.cs.nctu.edu.tw (Postfix) with ESMTP id 189DA3F65E
        for <lwhsu.gmail@gmail.com>; Mon, 11 May 2009 03:11:57 +0800 (CST)
Received: (from lwhsu@localhost)
        by bsd2.cs.nctu.edu.tw (8.14.3/8.14.2/Submit) id n4AJBuTM000652
        for lwhsu.gmail@gmail.com; Mon, 11 May 2009 03:11:56 +0800 (CST)
        (envelope-from lwhsu)
Date: Mon, 11 May 2009 03:11:56 +0800
From: Li-Wen Hsu <lwhsu@cs.nctu.edu.tw>
To: lwhsu.gmail@gmail.com
Subject: test if SPF record works
```

Sender Policy Framework (SPF) – Example of Forged Headers

- On zfs.cs.nctu.edu.tw
- Envelope From: lwhsu@zfs.cs.nctu.edu.tw
- Mail Headers
  - From: lwhsu@cs.nctu.edu.tw
  - To: lwhsu.gmail@gmail.com
- Related SPF Records:

cs.nctu.edu.tw	zfs.cs.nctu.edu.tw
"v=spf1 a mx a:csmailer.cs.nctu.edu.tw a:csmailgate.cs.nctu.edu.tw a:csmail.cs.nctu.edu.tw ~all"	"v=spf1 a ~all"

Sender Policy Framework (SPF) – Example of Forged Headers

```
Delivered-To: lwhsu.gmail@gmail.com
Received: by 10.223.112.14 with SMTP id u14cs45092fap;
        Mon, 23 May 2011 03:08:04 -0700 (PDT)
Received: by 10.236.80.65 with SMTP id j41mr2678377yhe.192.1306145283043;
        Mon, 23 May 2011 03:08:03 -0700 (PDT)
Return-Path: <lwhsu@zfs.cs.nctu.edu.tw>
Received: from zfs.cs.nctu.edu.tw (zfs.cs.nctu.edu.tw [140.113.17.215])
        by mx.google.com with ESMTP id 57si13494424yhl.14.2011.05.23.03.08.01;
        Mon, 23 May 2011 03:08:02 -0700 (PDT)
Received-SPF: pass (google.com: domain of lwhsu@zfs.cs.nctu.edu.tw designates
        140.113.17.215 as permitted sender) client-ip=140.113.17.215;
Authentication-Results: mx.google.com; spf=pass (google.com: domain of
        lwhsu@zfs.cs.nctu.edu.tw designates 140.113.17.215 as permitted sender)
        smtp.mail=lwhsu@zfs.cs.nctu.edu.tw
Received: by zfs.cs.nctu.edu.tw (Postfix, from userid 1001)
       id EBCF04B638; Mon, 23 May 2011 18:04:23 +0800 (CST)
Date: Mon, 23 May 2011 18:04:23 +0800
From: Li-Wen Hsu <lwhsu@cs.nctu.edu.tw>
To: lwhsu.gmail@gmail.com
Subject: test SPF

This is a SPF test.
```

Sender Policy Framework (SPF) – SPF and Forwarding

- Does SPF break forwarding?
  - Yes, but only if the receiver checks SPF without understanding their mail receiving architecture
  - Workaround
    - http://www.open-spf.org/FAQ/Forwarding
- SRS: Sender Rewriting Scheme
  - Forwarders should apply Sender Rewriting Scheme (SRS) to rewrite the sender address after SPF checks
    - http://www.open-spf.org/SRS

Sender Policy Framework (SPF) – Forwarding Example (no sender rewrite)

- On gmail (lwhsu.gmail's account)
  - Envelope From: lwhsu.gmail@gmail.com
- Mail Headers
  - From: lwhsu@cs.nctu.edu.tw
  - To: lwhsu@lwhsu.org
- On knight.lwhsu.org (lwhsu.org's mx)
  - ~lwhsu/.forward:
    - liwenhsu.gmail@gmail.com

gmail.com	**_spf.google.com**
"v=spf1 redirect=_spf.google.com"	"v=spf1 ip4:216.239.32.0/19 ip4:64.233.160.0/19 ip4:66.249.80.0/20 ip4:72.14.192.0/18 ip4:209.85.128.0/17 ip4:66.102.0.0/20 ip4:74.125.0.0/16 ip4:64.18.0.0/20 ip4:207.126.144.0/20 ip4:173.194.0.0/16 ?all"

```
Delivered-To: liwenhsu.gmail@gmail.com
Received: by 10.229.81.4 with SMTP id v4cs221969qck;
        Sun, 10 May 2009 11:09:26 -0700 (PDT)
Received: by 10.216.2.84 with SMTP id 62mr2907141wee.217.1241978964147;
        Sun, 10 May 2009 11:09:24 -0700 (PDT)
Return-Path: <lwhsu.gmail@gmail.com>
Received: from knight.lwhsu.ckefgisc.org (lwhsusvr.cs.nctu.edu.tw [140.113.24.67])
        by mx.google.com with ESMTP id 24si6143118eyx.13.2009.05.10.11.09.22;
        Sun, 10 May 2009 11:09:23 -0700 (PDT)
Received-SPF: neutral (google.com: 140.113.24.67 is neither permitted nor denied by domain
        of lwhsu.gmail@gmail.com) client-ip=140.113.24.67;
Authentication-Results: mx.google.com; spf=neutral (google.com: 140.113.24.67 is neither
        permitted nor denied by domain of lwhsu.gmail@gmail.com)
        smtp.mail=lwhsu.gmail@gmail.com;
Received: by knight.lwhsu.ckefgisc.org (Postfix)
        id 47F571143E; Mon, 11 May 2009 02:09:21 +0800 (CST)
Delivered-To: lwhsu@lwhsu.org
Received: from an-out-0708.google.com (an-out-0708.google.com [209.85.132.243])
        by knight.lwhsu.ckefgisc.org (Postfix) with ESMTP id D832B11431
        for <lwhsu@lwhsu.org>; Mon, 11 May 2009 02:09:20 +0800 (CST)
Received: by an-out-0708.google.com with SMTP id d14so1324869and.41
        for <lwhsu@lwhsu.org>; Sun, 10 May 2009 11:09:19 -0700 (PDT)
Sender: lwhsu.gmail@gmail.com
Received: by 10.100.248.4 with SMTP id v4mr14373811anh.121.1241978954295; Sun,
        10 May 2009 11:09:14 -0700 (PDT)
Date: Mon, 11 May 2009 02:09:13 +0800
Message-ID: <ef417ae30905101109j5c7b27bcy70a5bcf6d58092ab@mail.gmail.com>
Subject: test SPF
From: Li-Wen Hsu <lwhsu@cs.nctu.edu.tw>
To: lwhsu@lwhsu.org
```

Sender Policy Framework (SPF) – Some More Examples

- Outgoing Mail Gateway
  - List all authorized senders of cs.nctu.edu.tw

```
cs.nctu.edu.tw.          3600    IN      TXT     "v=spf1 a mx
a:farewell.cs.nctu.edu.tw a:csmailer.cs.nctu.edu.tw
a:tcsmailer.cs.nctu.edu.tw a:tcsmailer2.cs.nctu.edu.tw ~all"
```

- Incoming Mail Gateway

```
csmx1.cs.nctu.edu.tw.    3600    IN      TXT     "v=spf1 a -all"
csmx2.cs.nctu.edu.tw.    3600    IN      TXT     "v=spf1 a -all"
csmx3.cs.nctu.edu.tw.    3600    IN      TXT     "v=spf1 a -all"
```

When a mail server sends a bounce message (returned mail), it uses a null MAIL FROM: <>, and a HELO address that's supposed to be its own name. SPF will still operate, but in "**degraded mode**" by using the **HELO** domain name instead.

- BIND releases from 9.4.0 support the SPF RR type

Sender Policy Framework (SPF) – Backward Compatibility (1/2)

- When there is no SPF record, guess by A record

```
Delivered-To: lwhsu.gmail@gmail.com
Received: by 10.90.56.12 with SMTP id e12cs719147aga;
        Tue, 12 May 2009 00:49:39 -0700 (PDT)
Received: by 10.224.2.85 with SMTP id 21mr5508548qai.262.1242114578996;
        Tue, 12 May 2009 00:49:38 -0700 (PDT)
Return-Path: <lwhsu@freebsd.cs.nctu.edu.tw>
Received: from FreeBSD.cs.nctu.edu.tw (FreeBSD.cs.nctu.edu.tw [140.113.17.209])
        by mx.google.com with ESMTP id 7si4128629qwf.35.2009.05.12.00.49.38;
        Tue, 12 May 2009 00:49:38 -0700 (PDT)
Received-SPF: pass (google.com: best guess record for domain of
        lwhsu@freebsd.cs.nctu.edu.tw designates 140.113.17.209 as permitted sender)
        client-ip=140.113.17.209;
Authentication-Results: mx.google.com; spf=pass (google.com: best guess record for
        domain of lwhsu@freebsd.cs.nctu.edu.tw designates 140.113.17.209 as permitted
        sender) smtp.mail=lwhsu@freebsd.cs.nctu.edu.tw
Received: by FreeBSD.cs.nctu.edu.tw (Postfix, from userid 1058)
        id 6D98E61DBC; Tue, 12 May 2009 15:49:37 +0800 (CST)
Date: Tue, 12 May 2009 15:49:37 +0800
From: Li-Wen Hsu <lwhsu@FreeBSD.org>
To: lwhsu.gmail@gmail.com
Subject: test tw.freebsd.org SPF
```

Sender Policy Framework (SPF) – Backward Compatibility (2/2)

- Comparative result – when SPF record available:

```
Delivered-To: lwhsu.gmail@gmail.com
Received: by 10.90.56.12 with SMTP id e12cs719801aga;
        Tue, 12 May 2009 00:56:27 -0700 (PDT)
Received: by 10.224.74.84 with SMTP id t20mr5499756qaj.328.1242114987266;
        Tue, 12 May 2009 00:56:27 -0700 (PDT)
Return-Path: <lwhsu@freebsd.cs.nctu.edu.tw>
Received: from FreeBSD.cs.nctu.edu.tw (FreeBSD.cs.nctu.edu.tw [140.113.17.209])
        by mx.google.com with ESMTP id 5si4111810qwh.54.2009.05.12.00.56.26;
        Tue, 12 May 2009 00:56:27 -0700 (PDT)
Received-SPF: pass (google.com: domain of lwhsu@freebsd.cs.nctu.edu.tw
        designates 140.113.17.209 as permitted sender) client-ip=140.113.17.209;
Authentication-Results: mx.google.com; spf=pass (google.com: domain of
        lwhsu@freebsd.cs.nctu.edu.tw designates 140.113.17.209 as permitted sender)
        smtp.mail=lwhsu@freebsd.cs.nctu.edu.tw
Received: by FreeBSD.cs.nctu.edu.tw (Postfix, from userid 1058)
        id 78CD461DB0; Tue, 12 May 2009 15:56:25 +0800 (CST)
Date: Tue, 12 May 2009 15:56:25 +0800
From: Li-Wen Hsu <lwhsu@FreeBSD.org>
To: lwhsu.gmail@gmail.com
Subject: test tw.freebsd.org SPF (2)
```

Sender Policy Framework (SPF)
– Example of include mechanism

```
nctucs [~] -wangth- dig pixnet.net txt

;; ANSWER SECTION:
pixnet.net.                86400   IN      TXT     "v=spf1
include:aspmx.googlemail.com include:amazonses.com ip4:60.199.247.0/24
ip4:103.23.108.0/24 ip4:103.23.109.0/24 ip4:113.196.243.0/26 ~all"
```

DomainKeys and DKIM

- Verify the source of a mail
  - Allows an organization to claim responsibility for transmitting a message, in a way that can be validated by a recipient
  - With few computation cost
- Consortium spec
  - Derived from Yahoo DomainKeys and Cisco Identified Internet Mail
  - RFCs
    - DKIM Service Overview, RFC 5585
    - DKIM Signatures, RFC 6376
    - DomainKeys Identified Mail (DKIM) Development, Deployment and Operations, RFC 5863
    - DKIM Author Domain Signing Practices (ADSP), RFC 5617
  - https://www.dkim.org/

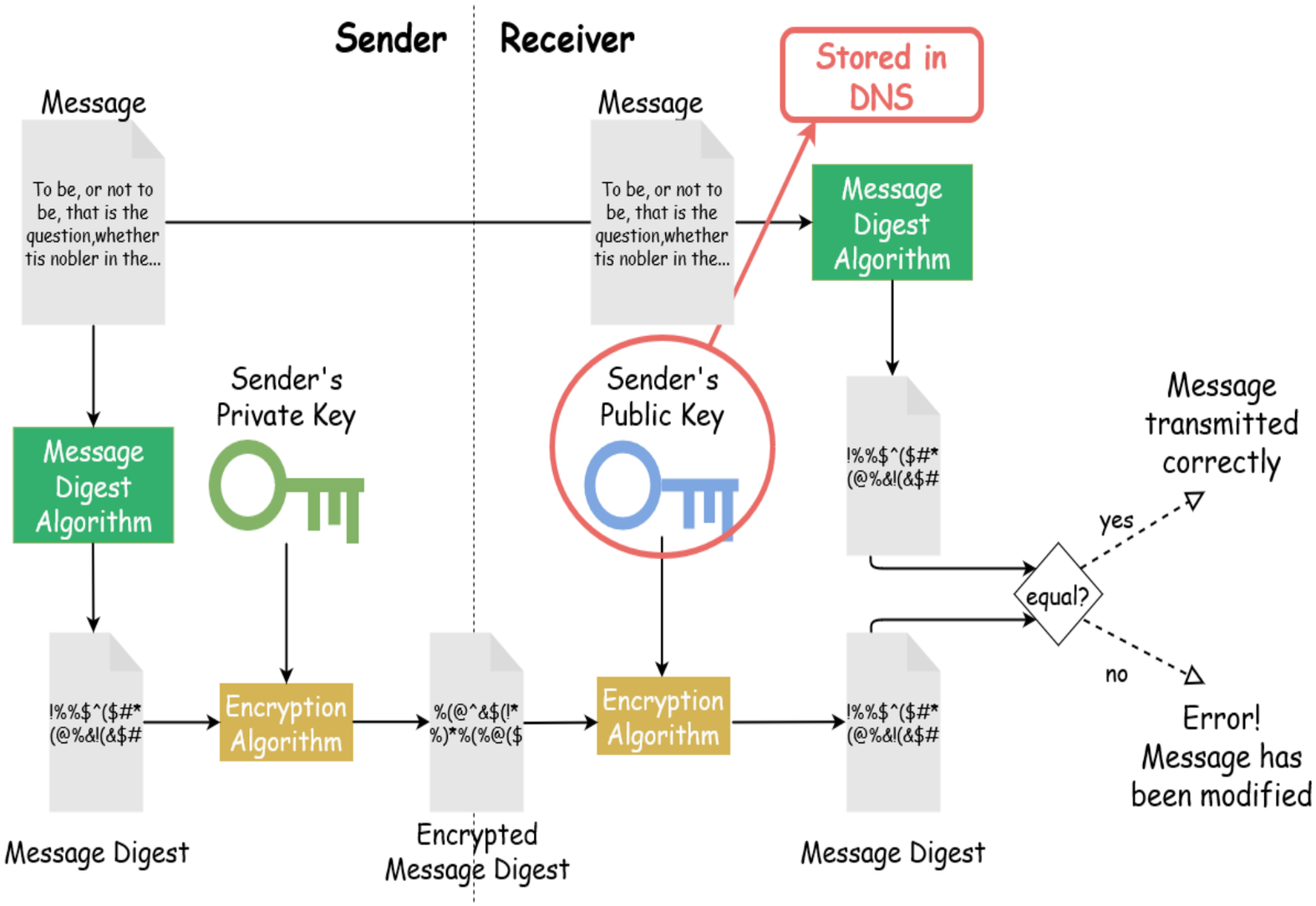
DKIM: Goals

- Validate message content, itself
  - Not related to path
- Transparent to end users
  - No client User Agent upgrades are required
  - But extensible to per-user signing
- Allow sender delegation
  - Outsourcing
- Low development, deployment, use costs
  - Avoid large PKI, new Internet services
  - No trusted third parties (except DNS)

DKIM: Idea

- Msg header authentication
  - DNS identifiers
  - Public keys in DNS
- End-to-end
  - Between origin/receiver administrative domains.
  - Not path-based

※ Digital signatures



DKIM: Technical High-points

- Signs body and selected parts of header
- Signature transmitted in DKIM-Signature header
- Public key stored in DNS
  - In _domainkey subdomain
  - New RR type, fall back to TXT
- Namespace divided using selectors
  - Allows multiple keys for aging, delegation, etc.
- Sender Signing Policy lookup for unsigned or improperly signed mail

DKIM – Signature header (1/3)

- v= Version
- a= Hash/signing algorithm
- q= Algorithm for getting public key
- d= Signing domain
- i= Signing identity
- s= Selector
- c= Canonicalization algorithm
- t= Signing time (seconds since 1/1/1970)
- x= Expiration time
- h= List of headers included in signature; dkim-signature is implied
- b= The signature itself
- bh= Body hash

DKIM – Signature header (2/3)

- Example:

```
DKIM-Signature: a=rsa-sha1; q=dns;
  d=example.com;
  i=user@eng.example.com;
  s=jun2005.eng; c=relaxed/simple;
  t=1117574938; x=1118006938;
  h=from:to:subject:date;
  b=dzdVyOfAKCdLXdJOc9G2q8LoXSlEniSb
    av+yuU4zGeeruD00lszZVoG4ZHRNiYzR
```

- DNS query will be made to:

```
jun2005.eng._domainkey.example.com
```

DKIM – Signature header (3/3)

- Example: Signature from cs.nycu.edu.tw

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=cs.nycu.edu.tw;
     s=20211117.mailer3.cs.nycu; t=1715745201;
     bh=LzwUKG6izezAD/gf5gU8TQz8uEmbIYlfPrtCEyGvuUs=;
     h=From:To:References:In-Reply-To:Subject:Date;
     b=VPmQdLkLf4MqEmoNbAZqSgVA2QOzwnuSW35WHtosts/mhVayIc2+WvbilJdoFxynV
     ESjOs1KmG9rezHLHpmFK9o7jSAOK/CoWfKLf8YANKtkD5ZJo/eiCTuyh7yrRN4HrVh
     3J36YTf6ey6LQBsKRhxgMA5HoRnijT6TOwTavdvg=
```

```
% dig +noall +answer txt
20211117.mailer3.cs.nycu._domainkey.cs.nycu.edu.tw
20211117.mailer3.cs.nycu._domainkey.cs.nycu.edu.tw. 86400 IN TXT
"v=DKIM1; k=rsa; "
"p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCslkD0L0hs/Yp/k2wkhNAHNTAJO5FwN
xjym2xJevF2r7qdh7CKHk9K2UDgMxhjq+jSIIJaBE0r0b1tmjSI7ht7kuqe2XbJns14xNgWf
gbULsXBWYheswH9qJAhXRb4I1Y1050bAJ6NshbwDBHTPYZPgs3WJ1NzmZw2gfszPDtocQIDA
QAB"
```

DKIM DNS Records

- Related DNS Records (RFC 6376)
  - v= Version (plain-text; REQUIRED).
  - k= Key type (plain-text; OPTIONAL, default is "rsa").
  - p= Public-key data (base64; REQUIRED).

```
% dig +noall +answer txt
20211117.mailer3.cs.nycu._domainkey.cs.nycu.edu.tw
20211117.mailer3.cs.nycu._domainkey.cs.nycu.edu.tw. 86400 IN TXT
"v=DKIM1; k=rsa; "
"p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCslkD0L0hs/Yp/k2wkhNAHNTAJO5FwN
xjym2xJevF2r7qdh7CKHk9K2UDgMxhjq+jSIIJaBE0r0b1tmjSI7ht7kuqe2XbJns14xNgWf
gbULsXBWYheswH9qJAhXRb4I1Y1050bAJ6NshbwDBHTPYZPgs3WJ1NzmZw2gfszPDtocQIDA
QAB"
```

DKIM Signature Verification

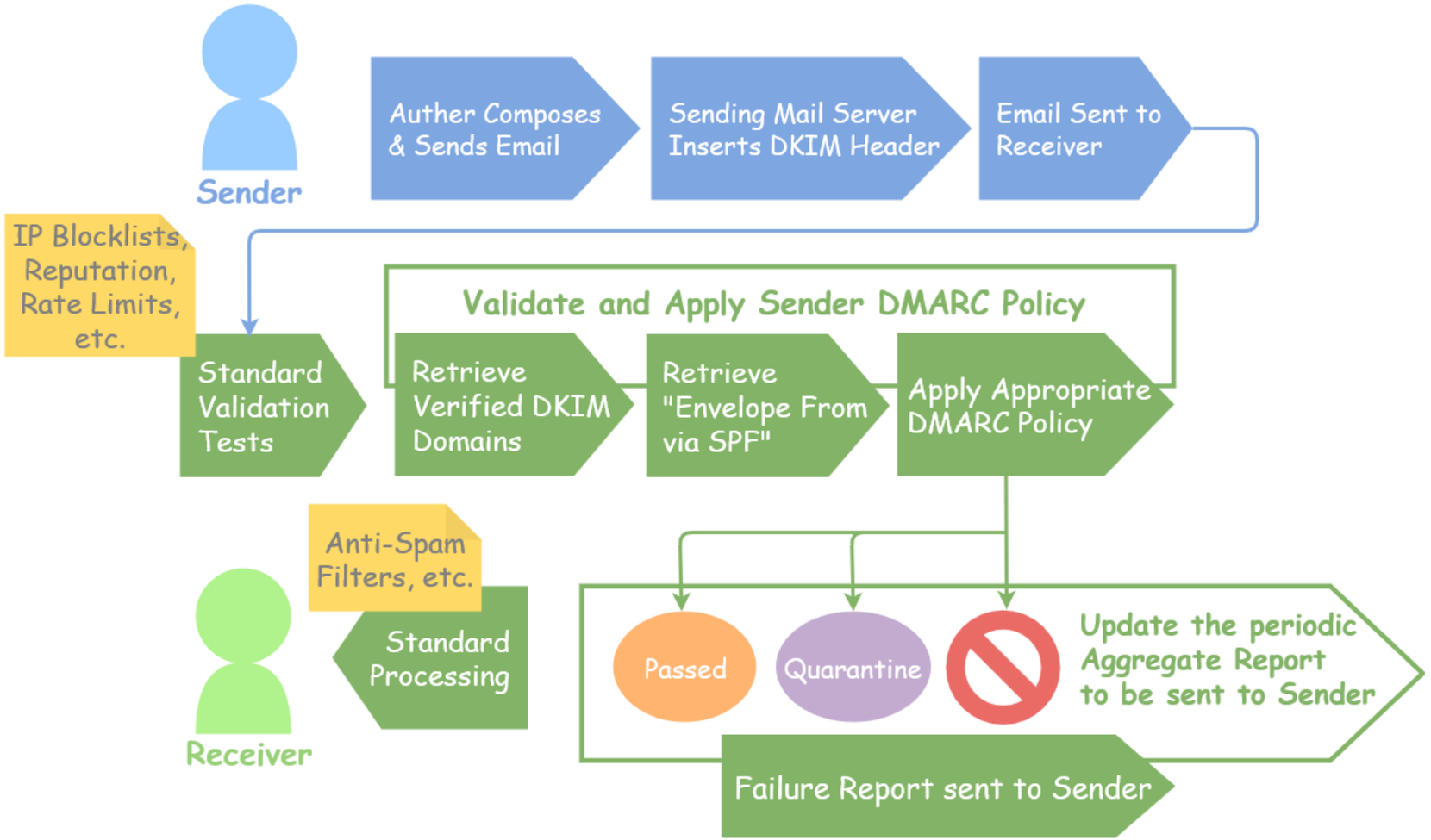
```
Received-SPF: pass (google.com: domain of wtchiang@cs.nctu.edu.tw designates
140.113.235.122 as permitted sender) client-ip=140.113.235.122;
Authentication-Results: mx.google.com;
       dkim=pass header.i=@cs.nycu.edu.tw header.s=20211117.mailer3.cs.nycu
header.b=VPmQdLkL;
       spf=pass (google.com: domain of wtchiang@cs.nctu.edu.tw designates
140.113.235.122 as permitted sender) smtp.mailfrom=wtchiang@cs.nctu.edu.tw;
       dmarc=pass (p=NONE sp=NONE dis=NONE) header.from=cs.nycu.edu.tw
Received: from mailer3.cs.nctu.edu.tw (localhost [127.0.0.1]) by mailer3.cs.nctu.edu.tw
(Postfix) with ESMTP id AE6A7415BA for <tsaimh@nycu.edu.tw>; Wed, 15 May 2024 11:53:21
+0800 (CST)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=cs.nycu.edu.tw;
s=20211117.mailer3.cs.nycu; t=1715745201;
bh=LzwUKG6izezAD/gf5gU8TQz8uEmbIYlfPrtCEyGvuUs=; h=From:To:References:In-Reply-
To:Subject:Date; b=VPmQdLkLf4MqEmoNbAZqSgVA2QOzwnuSW35WHtosts/mhVayIc2+WvbilJdoFxynV
        ESjOs1KmG9rezHLHpmFK9o7jSAOK/CoWfKLf8YANKtkD5ZJo/eiCTuyh7yrRN4HrVh
        3J36YTf6ey6LQBsKRhxgMA5HoRnijT6TOwTavdvg=
```

DMARC

- Domain-based Message Authentication, Reporting & Conformance
  - An email authentication, policy, and reporting protocol
  - It builds on SPF and DKIM protocols to provide greater assurance on the identity of the sender of a message
  - Provides feedback data to Domain Owners
  - Allow for blocking of unauthorized email
  - Policies are published as TXT record of DNS Service _dmarc.example.com
- https://dmarc.org

DMARC – The Email Authentication Process

- DMARC is designed to fit into an organization's existing inbound email authentication process



DMARC Record Syntax – Tag (1/3)

```
% dig +noall +answer txt _dmarc.cs.nycu.edu.tw
_dmarc.cs.nycu.edu.tw. 86012   IN        TXT     "v=DMARC1;
p=none; rua=mailto:dmarc-report-rua@cs.nycu.edu.tw;
sp=none; fo=1; adkim=s; aspf=s"
```

- v=<version>
  - <version>: DMARC1
  - Mandatory. This must be the first supplied tag=value within the dmarc specific text and, while DMARC tag=value pairs are not case sensitive, this one must have the explicit upper-case value DMARC1
- p=<policy>
  - <policy>: none, quarantine, reject
    - none: Monitoring, no impact on mail flows
    - quarantine: Deliver to spam folder
    - reject: Block mail that fails the DMARC check
  - Mandatory and must be the second tag=value pair. Defines the policy the sending MTA advises the receiving MTA to follow

DMARC Record Syntax – Tag (2/3)

- sp=<sub-domain policy>
  - <sub-domain policy>: none, quarantine, reject
  - Optional. If the following DMARC RR is present:

```
$ORIGIN example.com.
...
_dmarc       IN    TXT   "v=DMARC1;p=reject;sp=quarantine"
```

- Then failed mail from user@example.com would be rejected but
  - mail from user@a.example.com or user@b.a.example.com or
  - user@anything.example.com would be quarantined

DMARC Record Syntax – Tag (3/3)

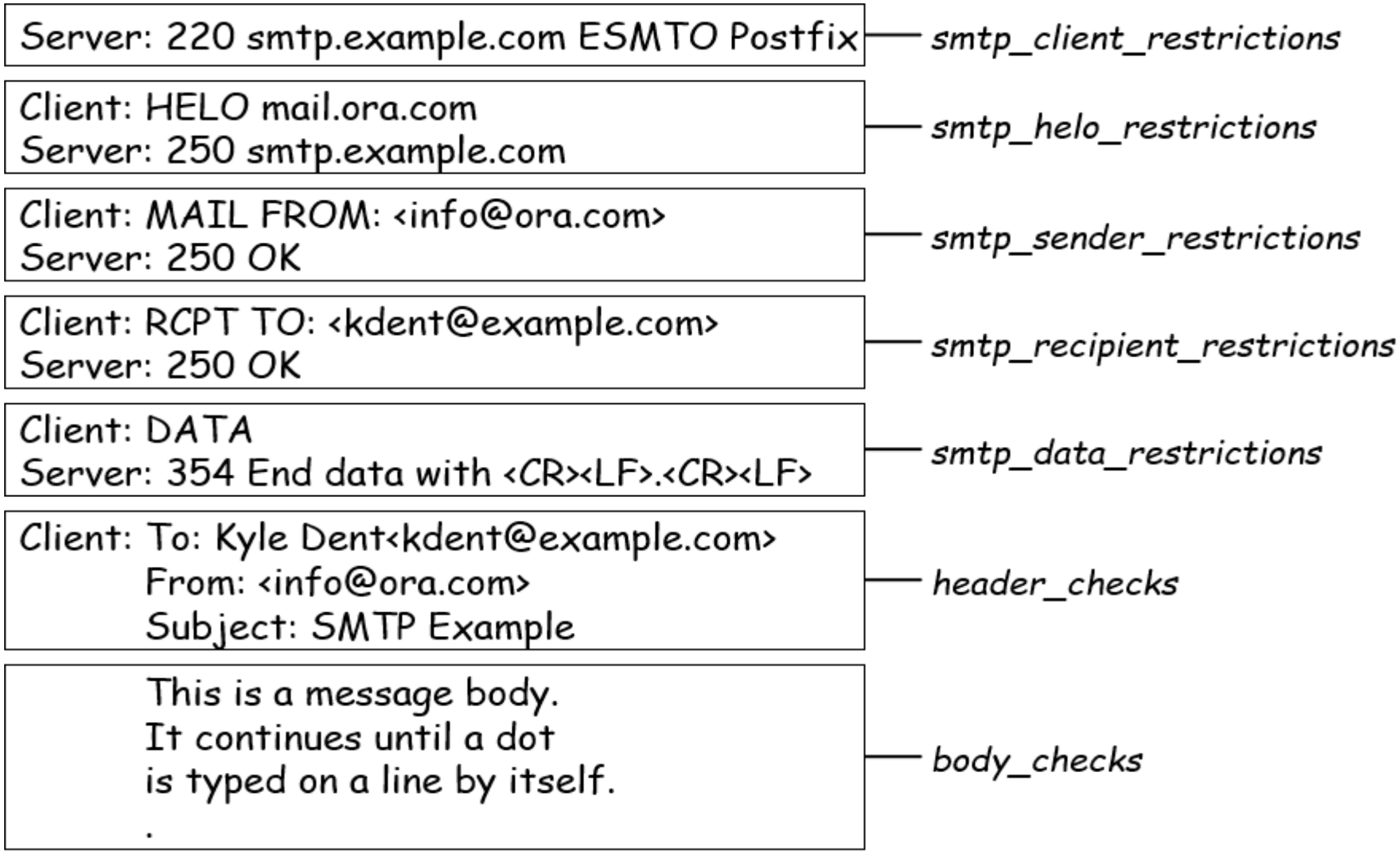
```
% dig +noall +answer txt _dmarc.cs.nycu.edu.tw
_dmarc.cs.nycu.edu.tw. 86012   IN        TXT      "v=DMARC1;
p=none; rua=mailto:dmarc-report-rua@cs.nycu.edu.tw;
sp=none; fo=1; adkim=s; aspf=s"
```

- rua=<@mail>
  - A comma delimited list of URI(s) for aggregate mail reports
- ruf=<@mail>
  - A comma delimited list of URI(s) for detailed failure reports
- fo (failure reporting)
  - fo=1: a DMARC failure/forensic report is sent to you when your email fails either SPF or DKIM alignment
- adkim / aspf (Alignment mode for DKIM/SPF)
  - either Strict (s) or Relaxed (r). E.g., in Strict mode, subdomains won't pass validation.

Handling Malicious Mail in Postfix

Postfix Anti – Spam configuration

- The SMTP Conversation
  - info@ora.com → smtp.example.com → kdent@example.com



Postfix Anti – Spam configuration – Client Detection Rules (1)

- Four rules in relative detection position
  - Rules and their default values
    - smtpd_client_restrictions =
    - smtpd_helo_required = yes
    - smtpd_helo_restrictions =
    - smtpd_sender_restrictions =
    - smtpd_recipient_restrictions =
      - permit_mynetworks, reject_unauth_destination
  - Each restriction check result can be:
    - OK (Accept in this restriction)
    - REJECT (Reject immediately without further check)
    - DUNNO (do next check)
  - Other options
    - disable_vrfy_command = yes

Postfix Anti – Spam configuration – Client Detection Rules (2)

- DNSBL/WL
  - smtpd_client_restrictions
- Greylisting
  - smtpd_recipient_restrictions
- SPF
  - smtpd_recipient_restrictions

Postfix Anti – Spam configuration – Client Detection Rules (3)

1. Access maps – access(5)
   - List of IP addresses, hostnames, email addresses
   - Can be used in:

smtpd_client_restrictions = check_client_access hash:/usr/local/etc/postfix/access
smtpd_helo_restrictions = check_helo_access hash:/usr/local/etc/postfix/helohost
smtpd_sender_restrictions = check_sender_access hash:/usr/local/etc/postfix/sender_access
smtpd_recipient_restrictions = check_recipient_access hash:/usr/local/etc/postfix/rcpt_access

   - Actions
     - OK, REJECT, DUNNO
     - FILTER (redirect to content filter)
     - HOLD (put in hold queue)
     - DISCARD (report success to client but drop)
     - 4xx message or 5xx message

Postfix Anti – Spam configuration – Client Detection Rules (4)

- Example of access maps
  - check_client_access hash:/etc/access

    nctu.edu.tw OK
    127.0.0.1 OK
    61.30.6.207 REJECT
    /\.dynamic\./ REJECT (regexp:)
  - check_helo access hash:/postfix/helohost

    greatdeals.example.com REJECT
    oreillynet.com OK
  - check_sender_access hash:/usr/local/etc/postfix/sender_access

    sales@viagra.com 553 Please contact +886-3-5712121-54707.
    viagra.com 553 Invalid MAIL FROM
    .viagra.com 553 Invalid MAIL FROM
    manager@ 553 Invalid MAIL FROM
  - check_recipient_access hash:/usr/local/etc/postfix/recipient_access

    bin@cs.nctu.edu.tw 553 Invalid RCPT TO command
    ftp@cs.nctu.edu.tw 553 Invalid RCPT TO command
    man@cs.nctu.edu.tw 553 Invalid RCPT TO command

Postfix Anti – Spam configuration – Client Detection Rules (5)

2. Special client-checking restrictions
   - permit_auth_destination
     - Mostly used in "smtpd_recipient_restrictions"
     - Permit request if destination address matches:
       - The postfix system's final destination setting
         - mydestination, inet_interfaces, vitual_alias_domains, virtual_mailbox_domains
       - The postfix system's relay domain
         - relay_domains
     - Found ➔ OK, UnFound ➔ DUNNO
   - reject_unauth_destination
     - Opposite to permit_auth_destination
     - Found ➔ REJECT, UnFound ➔ DUNNO
   - permit_mynetworks
     - Allow a request if client IP match any address in "mynetworks"
       - Usually used in smtpd_recipient_restrictions

Postfix Anti – Spam configuration – Client Detection Rules (6)

3. Strict syntax restrictions
   > Restrictions that does not conform to RFC

- reject_invalid_helo_hostname
  - Reject hostname with bad syntax
- reject_non_fqdn_helo_hostname
  - Reject hostname not in FQDN format
- reject_non_fqdn_sender
  - For "MAIL FROM" command
- reject_non_fqdn_recipient
  - For "RCPT TO" command

Postfix Anti – Spam configuration – Client Detection Rules (7)

4. DNS restrictions
   > Make sure that clients and email envelope addresses have valid DNS information

- reject_unknown_client_hostname
  - Reject if the DNS records related to the client IP unreasonable
- reject_unknown_helo_hostname
  - Reject if EHLO hostname has no DNS MX or A record
- reject_unknown_sender_domain
  - Reject if MAIL FROM domain name has no DNS MX or A record
- reject_unknown_recipient_domain
  - Reject if RCPT TO domain name has no DNS MX or A record

Postfix Anti – Spam configuration – Client Detection Rules (8)

5. Real-time blacklists
   - Check with DNSxL services
   - permit_dnswl_client list.dnswl.org
     - http://www.dnswl.org/
   - reject_rbl_client domain.tld[=d.d.d.d]
     - Reject if client IP is detect in DNSBL
   - reject_rhsbl_client domain.tld[=d.d.d.d]
     - Reject if client hostname has an A record under specified domain
   - reject_rhsbl_sender domain.tld[=d.d.d.d]
     - Reject if sender domain in address has an A record under specified domain

   - smtpd_client_restrictions =
     hash:/etc/access, reject_rbl_client relays.ordb.org
   - smtpd_sender_restrictions =
     hash:/usr/local/etc/postfix/sender_access,
     reject_rhsbl_sender dns.rfc-ignorant.org

Postfix Anti – Spam configuration – Client Detection Rules (9)

6. Policy Service
   - Postfix SMTP server sends in a delegated SMTPD access policy request to one special service (policy service).
   - Policy service replies actions allowed in Postfix SMTPD access table.
   - Usage:
     - check_policy_service *servicename*
   - Example: Greylisting (Using Postgrey)
     - mail/postgrey
       - /usr/local/etc/postfix/postgrey_whitelist_clients
       - /usr/local/etc/postfix/postgrey_whitelist_recipients
     - postgrey daemon runs on port 10023
     - In main.cf
       - smtpd_recipient_restrictions = …, reject_unauth_destination, check_policy_service inet:127.0.0.1:10023

Postfix Anti – Spam configuration – Client Detection Rules (10)

- Example: SPF Checking (Using postfix-policyd-spf-perl)
  - mail/postfix-policyd-spf-perl
    - /usr/local/etc/postfix/postgrey_whitelist_clients
    - /usr/local/etc/postfix/postgrey_whitelist_recipients
  - SPF policy service daemon runs on a Unix domain socket
  - In master.cf

```
policyd-spf unix - n n - 0 spawn user=nobody argv=/usr/local/libexec/postfix-policyd-spf-perl
```

  - In main.cf
    - smtpd_recipient_restrictions = …, reject_unauth_destination, check_policy_service unix:private/policy-spf
    - spf-policy_time_limit = 3600

Postfix Anti – Spam configuration – Client Detection Rules (11)

- smtpd_client_restrictions
  - check_client_access
  - reject_unknown_client_hostname
  - permit_mynetworks
  - reject_rbl_client
  - reject_rhsbl_client
- smtpd_helo_restrictions
  - check_helo_access
  - reject_invalid_helo_hostname
  - reject_unknown_helo_hostname
  - reject_non_fqdn_helo_hostname
- smtpd_sender_restrictions
  - check_sender_access
  - reject_unknown_sender_domain
  - reject_rhsbl_sender
- smtpd_recipient_restrictions
  - check_recipient_access
  - permit_auth_destination
  - reject_unauth_destination
  - reject_unknown_recipient_domain
  - reject_non_fqdn_recipient
  - check_policy_service

Postfix Anti – Spam configuration – Content Inspection

- before queue, built-in, light-weight
  - header_checks, body_checks
- after queue, external, heavy-weight
  - Use smtp, pipe, etc. to inject mail to filters
    - content_filter
  - Accept: Re-inject mail back into Postfix
  - Reject: Discard mail / Reject mail
- before queue, external, medium-weight
  - Method 1: SMTP proxy (smtp)
    - smtpd_proxy_filter
  - Method 2: Sendmail Milter (milter protocol)
    - SMTP-only: Invoked by smtpd(8), for mail arriving via smtpd(8) server
      - smtpd_milters, milter_*
    - non-SMTP: Invoked by cleanup(8), for mail arriving via sendmail(1), i.e. local mail
      - non_smtpd_milters, milter_*
- Pros and cons
  - http://www.postfix.org/documentation.html "Content inspection" Section

Postfix Anti – Spam configuration – Content – Checking rules (1)

- 4+ rules – header_checks(5)
  - header_checks
    - Check for message headers
  - mime_header_checks
    - Check for MIME headers
  - nested_header_checks
    - Check for attached message headers
  - body_check
    - Check for message body
- All rules use lookup tables
  - Ex:

```
header_checks = regexp:/usr/local/etc/postfix/header_checks
body_checks = pcre:/usr/local/etc/postfix/body_checks
```

Postfix Anti – Spam configuration – Content – Checking rules (2)

- Content-checking lookup table
  - Regular_Expression Action
- Actions
  - REJECT message
  - WARN message
    - Log a “warning:” record, for debugging
  - IGNORE
    - Delete matched line of headers or body
  - HOLD message
    - Stay there until the administrator intervenes
  - DISCARD message
    - Claim successful delivery but silently discard
  - FILTER message
    - Send message through a separate content filter

Postfix Anti – Spam configuration – Content – Checking rules (3)

- Example of header check
  - header_checks = regexp:/usr/local/etc/postfix/header_checks
  - In /usr/local/etc/postfix/header_checks

```
/take advantage now/          REJECT
/repair your credit/          REJECT
```

- Example of body check
  - body_checks = regexp:/usr/local/etc/postfix/body_checks
  - In /usr/local/etc/postfix/body_checks

```
/lowest rates.*\!/            REJECT
/[:alpha:]<!--.*-->[:alpha:]/ REJECT
```

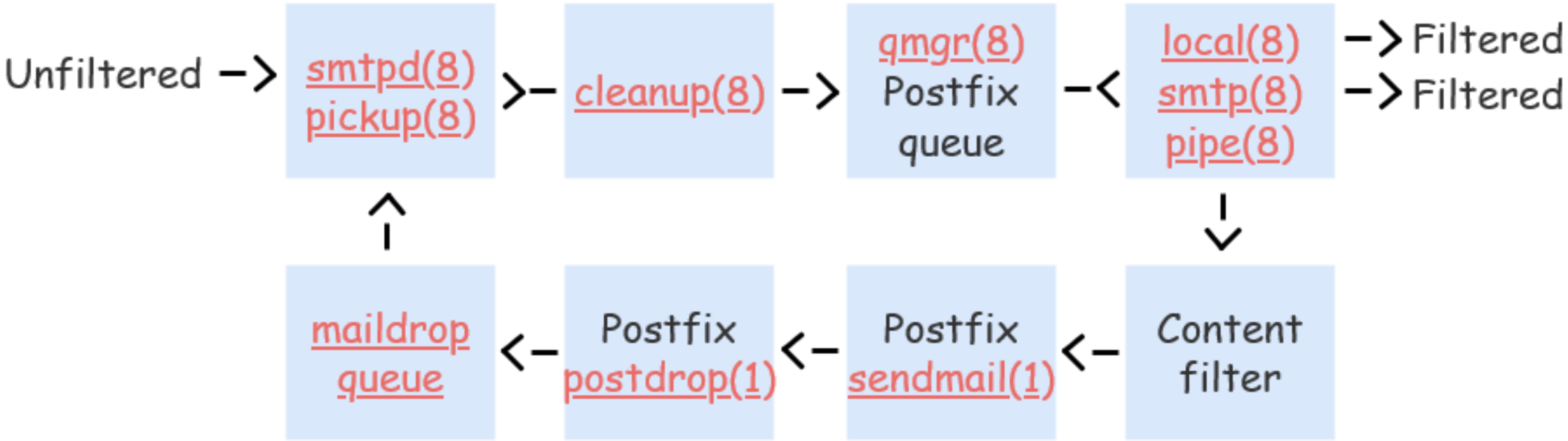
External Filters (After-queue) – (1)

- After-queue filters can be done on
  - MTA
  - MDA
  - MUA

  ※ Combination of MTA and MUA
  - ➢ Adding some extra headers or modifying subject in MTA, and filtering in MUA.

- Types of after-queue external filters
  - Command-based filtering
    - ➢ New process is started for every message
    - ➢ Accept message from STDIN
  - Daemon-based filtering
    - ➢ Stay resident
    - ➢ Accept message via SMTP or LMTP

External Filters (After-queue) – (2)



- https://www.postfix.org/FILTER_README.html

MDA Filter: Procmail (1)

- Install Procmail (port or package)
- Enable Procmail in Postfix
  - In main.cf

```
mailbox_command = /usr/local/bin/procmail
```

- Create configuration file
  - Create /usr/local/etc/procmailrc
- Create log files
  - touch /var/log/procmail.log
- Create directories (optional)
  - mkdir -p /tmp/trash

```
VERBOSE=off
LOGFILE=/var/log/procmail.log

:0b
* ^Subject:.*GGWP.*
/dev/null

:0b
* ^Subject:.*LOL.*
/tmp/trash
```

procmailrc

MDA Filter: Procmail (2-1) - Filter Chinese Text

- Encoding problem
  - We need to set two types of encoded Chinese text
  - Base64 and Quote-Printable
- Tool: mmencode (port or package)
- Generate encoded text
  - Filter “減肥”
  - Generate Base64 code

```
> echo -n "減肥" | mmencode
5rib6IKl
```

  - Generate QP code

```
> echo -n "減肥" | mmencode -q
=E6=B8=9B=E8=82=A5=
```

MDA Filter: Procmail (2-2) - Filter Chinese Text

- Write two rules to filter Chinese text

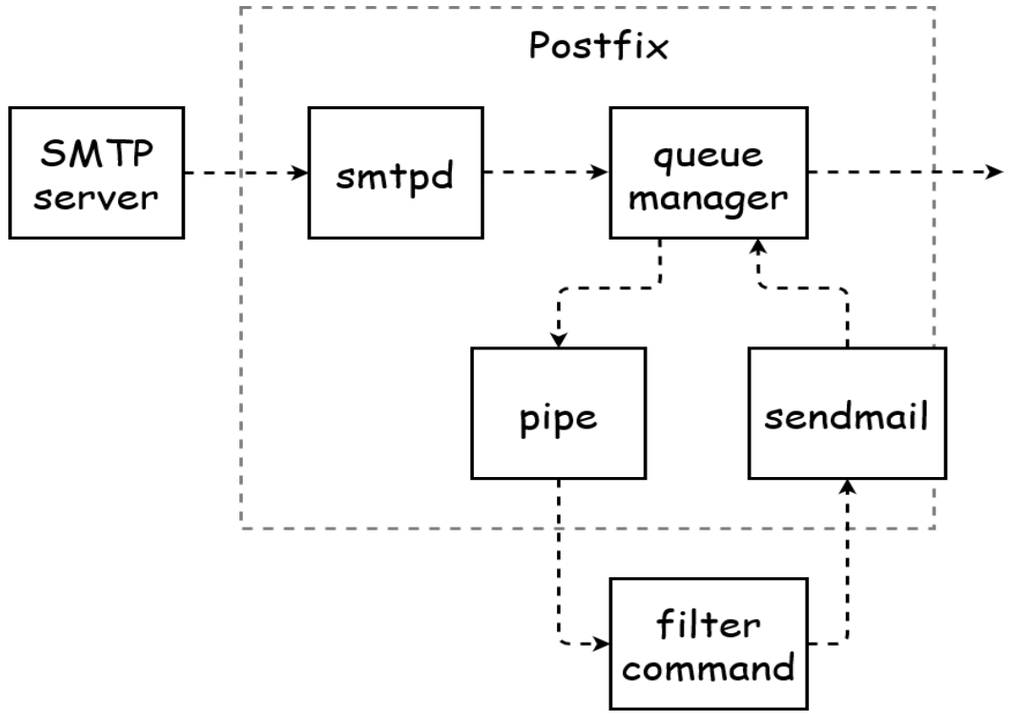
```
# Base64
:0b
* ^Subject:.*5rib6IKl.*
/dev/null

# Quote-Printable
:0b
* ^Subject:.*=E6=B8=9B=E8=82=A5=.*
/dev/null
```

- Log file

```
From lctseng@nasa.lctseng.nctucs.net  Wed Mar  9 12:14:46 2016
 Subject: =?UTF-8?B?5rib6IKl?=
  Folder: /dev/null                                        1
```

Command-Based Filtering (1)

- Usage
  - Postfix delivers message to this filter via “pipe” mailer
  - Program that accepts content on its STDIN
  - Program gives the filtered message back to Postfix using the “sendmail” command (with same queue ID)



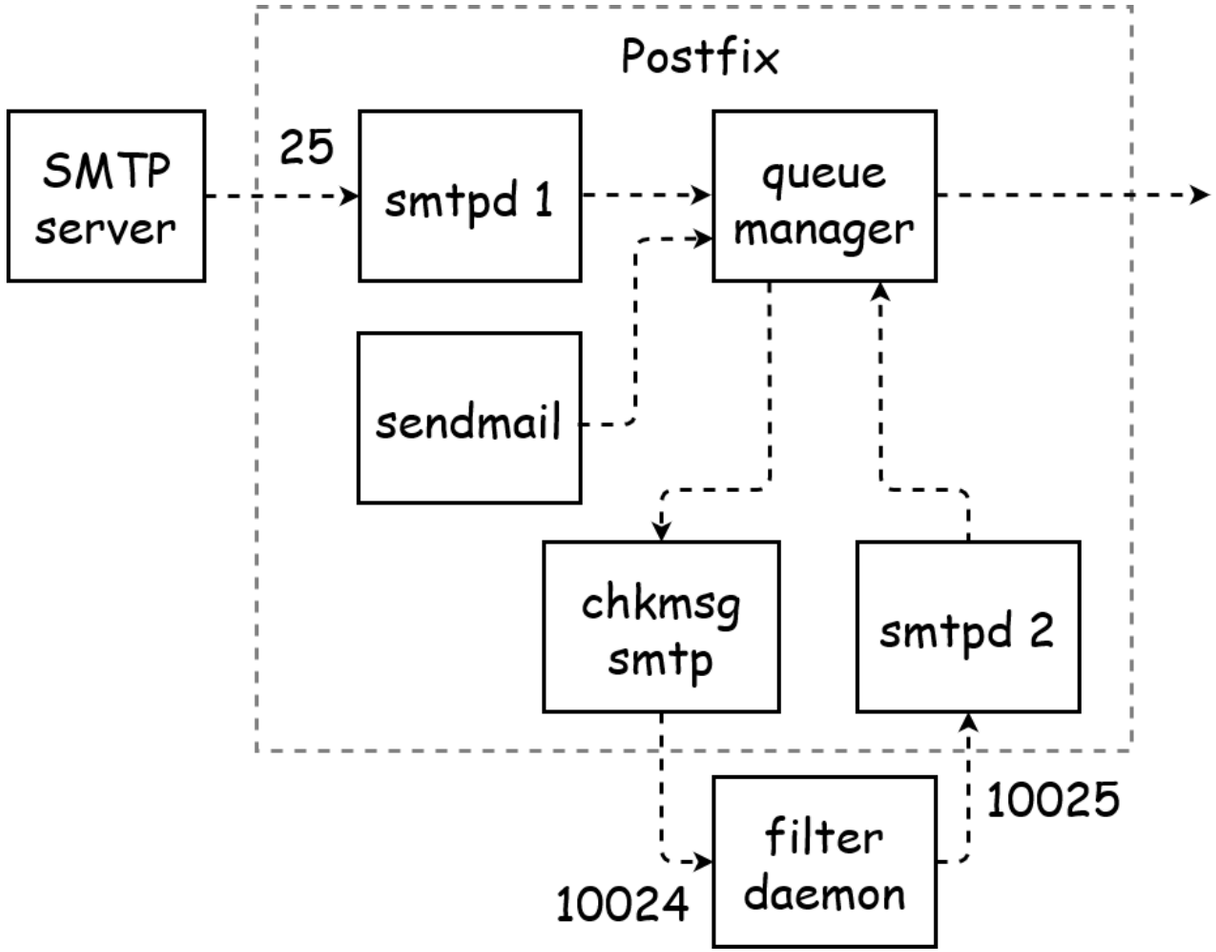
Command-Based Filtering (2)

- Configuration
  - Prepare your filter program (/usr/local/bin/simple_filt)
  - Modify master.cf

```
#==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#==========================================================================
filter  unix  -         n       n       -       -       pipe
        flags=Rq user=filter argv=/usr/local/bin/simple_filt -f ${sender} - -${recipient}
smtpd   inet  n         -       n       -       -       smtpd
        -o content_filter=filter:
```

Daemon-Based Filtering (1)

- Usage
  - Message is passed back and forth between Postfix and filtering daemon via SMTP or LMTP



Daemon-Based Filtering (2) - amavisd-new

- Primary daemon: amavisd-new
  - Cooperate with other programs
  - ClamAV (anti-virus), SpamAssassin (anti-spam)
- Configuration for amavisd
  - Install and configure your content filter
    - security/amavisd-new (port or package)
    - Modify amavisd.conf to send message back

```
$forward_method = 'smtp:127.0.0.1:10025';
```

  - Edit /etc/rc.conf

```
amavisd_enable="YES"
```

  - Edit main.cf to let postfix use filtering daemon

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Daemon-Based Filtering (3) - amavisd-new

- Configuration
  - Edit master.cf to add two additional services

```
smtp-amavis unix -      -       n       -       10      smtp
    -o smtp_data_done_timeout=1200s
    -o smtp_never_send_ehlo=yes
    -o notify_classes=protocol,resource,software
127.0.0.1:10025 inet n  -       n       -       -       smtpd
    -o content_filter=
    -o mynetworks=127.0.0.0/8
    -o local_recipient_maps=
    -o notify_classes=protocol,resource,software
    -o myhostname=localhost
    -o smtpd_client_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o smtpd_tls_security_level=
```

Daemon-Based Filtering (4) - amavisd-new

- Now, your amavisd-new is ready
  - With SpamAssassin installed
  - Run “sa-update” to update the SpamAssassin rules
  - Edit SpamAssassin configuration in amavisd.conf
    - E.g. Change spam detect level

```
$sa_tag2_level_deflt = 3.0;
```

Daemon-Based Filtering (5) - amavisd-new

- The mail source in SPAM-detected mail

```
Received: from demo1.nasa.lctseng.nctucs.net (localhost [127.0.0.1])
        by localhost (Postfix) with ESMTP id 1A945274
        for <lctseng@nasa.lctseng.nctucs.net>; Wed,  9 Mar 2016 14:14:39 +0800 (CST)
X-Virus-Scanned: amavisd-new at nasa.lctseng.ncatucs.net
X-Spam-Flag: YES
X-Spam-Score: 4.85
X-Spam-Level: ****
X-Spam-Status: Yes, score=4.85 tagged_above=2 required=3
        tests=[FREEMAIL_ENVFROM_END_DIGIT=0.25, FREEMAIL_FROM=0.001,
        HTML_FONT_LOW_CONTRAST=0.001, HTML_MESSAGE=0.001,
        RCVD_IN_DNSWL_LOW=-0.7, RCVD_IN_MSPIKE_H3=-0.01,
        RCVD_IN_MSPIKE_WL=-0.01, T_REMOTE_IMAGE=0.01, URIBL_ABUSE_SURBL=1.948,
        URIBL_BLACK=1.7, URIBL_WS_SURBL=1.659] autolearn=no autolearn_force=no
Authentication-Results: demo1.nasa.lctseng.nctucs.net (amavisd-new);
        dkim=pass (2048-bit key) header.d=gmail.com
Received: from demo1.nasa.lctseng.nctucs.net ([127.0.0.1])
        by demo1.nasa.lctseng.nctucs.net (demo1.nasa.lctseng.nctucs.net [127.0.0.1])
(amavisd-new, port 10024)
        with SMTP id CjRyliYl5l6x for <lctseng@nasa.lctseng.nctucs.net>;
        Wed,  9 Mar 2016 14:14:38 +0800 (CST)
```

Daemon-Based Filtering (6) - amavisd-new+ ClamAV

- amavisd-new supports lots of anti-virus scanner
- Anti-virus with ClamAV
  - Install security/clamav (port or package)
  - Edit /etc/rc.conf

```
clamav_clamd_enable="YES"
```

  - Update virus database
    - Run “freshclam”
  - Specify to use clamav in amavisd.conf

```
@av_scanners = (

['ClamAV-clamd',
   \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.sock"],
   qr/\bOK$/m, qr/\bFOUND$/m,
   qr/^.*?: (?!Infected Archive)(.*) FOUND$/m ],
);
```

Daemon-Based Filtering (7) - amavisd-new+ ClamAV

- Set alias for “virusalert” user
  - When there is an infected mail, it will send a notification to this user
  - Alias to “root” or “postmaster”
- Start ClamAV and restart amavisd-new
  - service clamav-clamd start
  - service amavisd restart
- Send a test virus by EICAR organization
  - Plain text

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

  - Reference: https://en.wikipedia.org/wiki/EICAR_test_file

Daemon-Based Filtering (8) - amavisd-new+ ClamAV

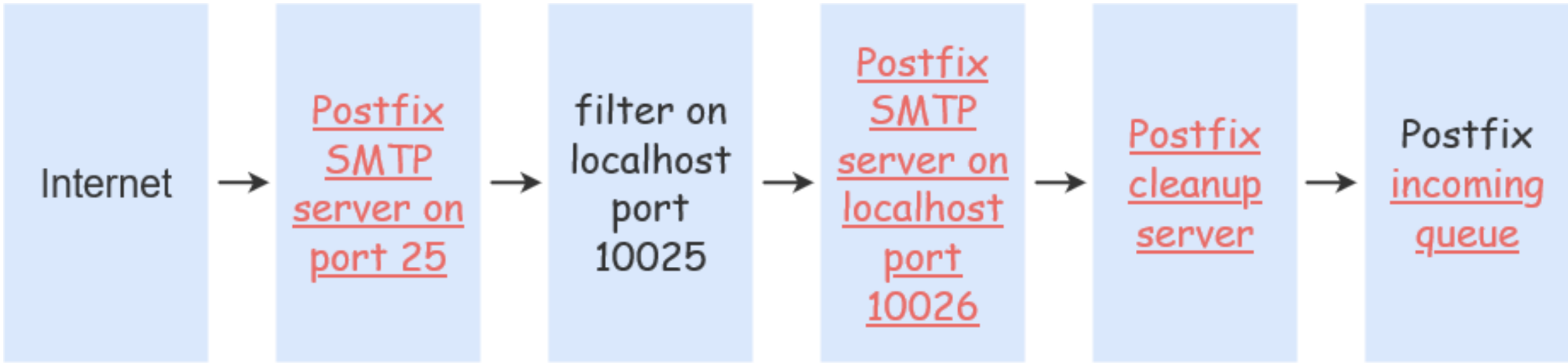
- Result of sending EICAR test mail

從 Content-filter at demo1.nasa.lctseng.nctucs.net <virusalert@nasa.lctseng.nctucs.net>
主旨 VIRUS (Eicar-Test-Signature) in mail FROM [127.0.0.1] <lctseng@nasa.lctseng.nctucs.net>
給 virusalert@nasa.lctseng.nctucs.net

```
A virus was found: Eicar-Test-Signature

Scanner detecting a virus: ClamAV-clamscan

Content type: Virus
Internal reference code for the message is 93683-01/SIxGUR_-RBuT

First upstream SMTP client IP address: [127.0.0.1]

Received trace: ESMTPSA://140.113.209.205

Return-Path: <lctseng@nasa.lctseng.nctucs.net>
From: Liang-Chi Tseng <lctseng@nasa.lctseng.nctucs.net>
Message-ID: <56DFCCE9.2010608@nasa.lctseng.nctucs.net>
Subject: CC
The message has been quarantined as: virus-SIxGUR_-RBuT

The message WAS NOT relayed to:
<lctseng@nasa.lctseng.nctucs.net>:
   250 2.7.0 ok, discarded, id=93683-01 - infected: eicar-test-signature

Virus scanner output:
  p001: Eicar-Test-Signature FOUND
```

External Filters (Before-queue) – (1)

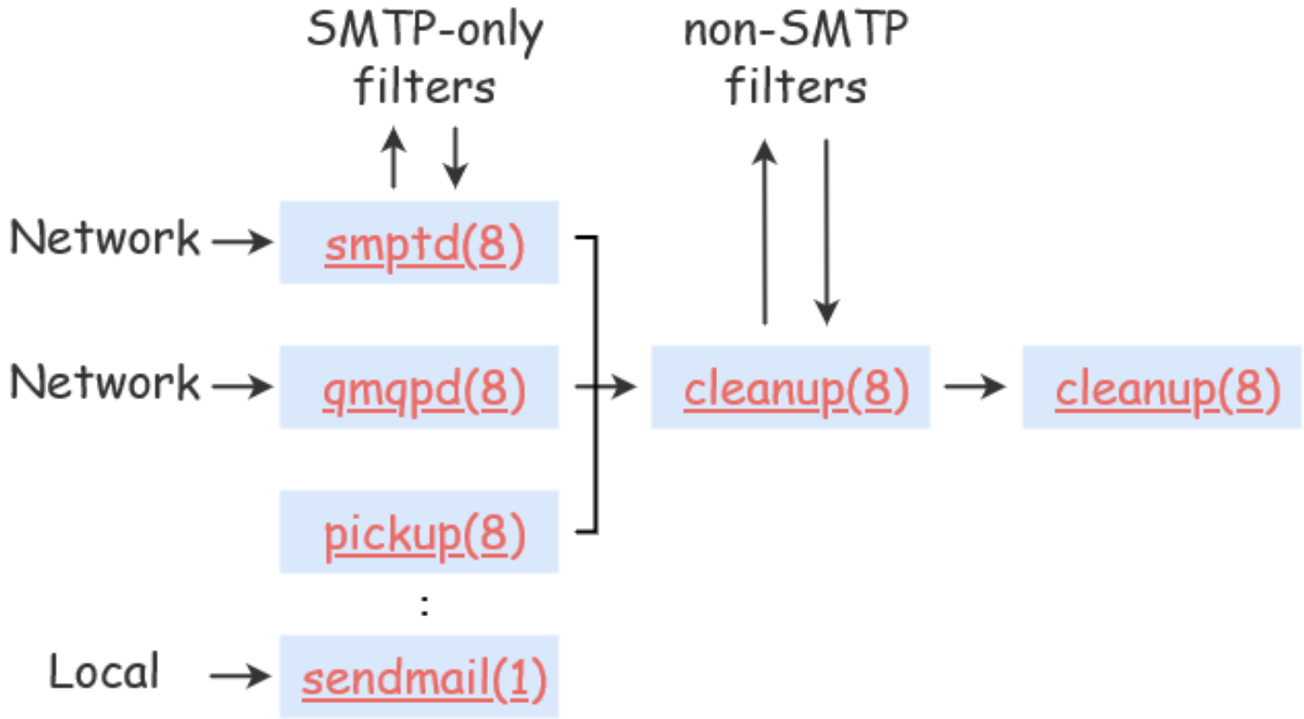
- Types of before-queue external filters
  - SMTP proxy (smtp)
    - smtpd_proxy_filter
  - From after-queue to before-queue (Software support)
    - content_filter → smtpd_proxy_filter



- http://www.postfix.org/SMTPD_PROXY_README.html

External Filters (Before-queue) – (2)

- Types of before-queue external filters
  - Sendmail Milter (milter protocol)
    - SMTP-only: Invoked by smtpd(8), for mail arriving via smtpd(8) server
      - smtpd_milters, milter_*
    - non-SMTP: Invoked by cleanup(8), for mail arriving via sendmail(1), i.e. local mail
      - non_smtpd_milters, milter_*



https://www.postfix.org/MILTER_README.html

Appendix

Postfix Postscreen
Rspamd

postscreen – Postfix zombie blocker

- Postscreen (Postfix ≥ 2.8)
  - Provide additional protection against mail server overload
  - Handle multiple inbound SMTP connections in one process
  - Decide which clients may talk to the Postfix SMTP server process
- How it works?
  - Maintain a temporary whitelist for clients passing its tests
  - Allow whitelisted clients to skip tests
- CAUTION
  - Not be used on SMTP ports that receive mail from MUAs
  - postscreen is used on port 25
  - MUAs submit mail via the submission service (port 587)
    - Separate IMG/OMG: MX settings
  - http://www.postfix.org/POSTSCREEN_README.html

postscreen – Basic idea

- Most mails are spam
  - Spend most resources not receiving mail
- Mail challenge: Keep zombies away
  - Make an is-it-a-zombie decision
  - Whitelist while deciding a client not-a-zombie to avoid further delay
- Zombies' challenge:
  - Only a limited amount of time to deliver spam before being blacklisted
  - To speed up
    - Speak before their turn
    - Ignore response from SMTP servers
- To recognize zombies
  - Determine if the remote SMTP client IP is blacklisted
  - Look for protocol compromises

postscreen – General operation

- postscreen
  - Involve a number of tests
- Some tests introduce a delay of a few seconds
  - Maintain a temporary whitelist for clients passing its tests
  - Minimize its impact on legitimate email traffic
- Default
  - Hand off all connections to the SMTP server after logging
  - Useful for non-destructive testing
- Typical production setting
  - Reject mail from clients failing one or more tests
  - Log helo, sender, and recipient information

postscreen – Quick tests

- Query local blacklists/whitelists
  - Permanent whitelist/blacklist test
    - postscreen_access_list = permit_mynetworks,
      cidr:postscreen_access.cidr
    - In postscreen_access.cidr (first-matching)
      192.168.0.1 permit / dunno
      192.168.0.0/16 reject
    - **WHITELISTED** *[address]:port*
      BLACKLISTED *[address]:port*
  - Temporary whitelist test
    - **PASS OLD** *[address]:port*
  - MX policy test
    - postscreen_whitelist_interfaces = !168.100.189.8 static:all
    - **CONNECT from** *[address]:port* **to [168.100.189.8]:25**
      WHITELIST VETO *[address]:port*

postscreen – Tests before greeting – (1)

- The SMTP server should speak before the client
  - A short delay before "220 …" server greeting
    - For DNSWL/BL lookup results to arrive
    - postscreen_greet_wait = ${stress?2}${stress:6}s
- Pregreet test
  - Detect zombies that speak before their turn
  - postscreen_greet_banner = $smtpd_banner
    - "220-text …" vs. "220 text …“
    - Disable the teaser banner
      - postscreen_greet_banner =
  - **PREGREET** *count* **after** *time* **from** *[address]:port text...*

postscreen – Tests before greeting – (2)

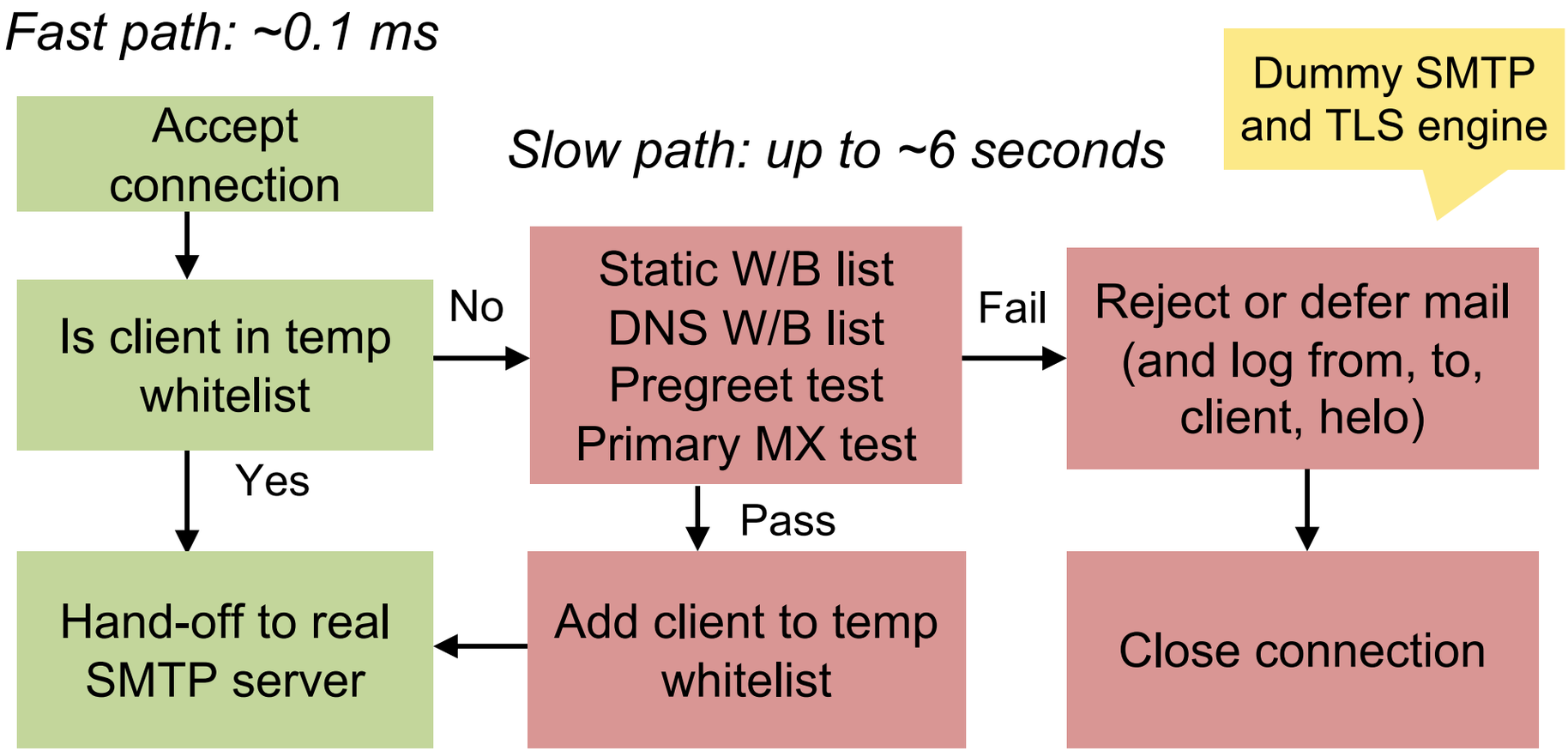
- DNSWL/BL test
  - postscreen_dnsbl_sites = highqualityblacklist.example.com*2
    lowerqualityblacklist.example.net
    list.dnswl.org*-5
    example.com=127.0.0.4
  - postscreen_dnsbl_threshold = 1
    - Determine when postscreen_greet_wait time has elapsed
  - postscreen_dnsbl_reply_map = texthash:dnsbl_reply
    - In dnsbl_reply

      secret.zen.spamhaus.org zen.spamhaus.org
  - **DNSBL rank** *count* **for** *[address]:port*
  - Wietse needed new material for a LISA conference presentation in November 2010, so he added support for DNSBL weights and filters in August

postscreen – Tests fail before greeting

- Actions
  - ignore (default)
  - enforce
    - Allow other tests to complete, reply 550, and log helo/sender/recipient
  - drop
    - Reply 521 immediately
- postscreen_*_action
  - postscreen_blacklist_action
    - Match permanent blacklist
  - postscreen_greet_action
    - Fail pregreet test
  - postscreen_dnsbl_action
    - DNSBL score is equal to or greater than the threshold

postscreen – Workflow before SMTP



postscreen – Multi-layer defense

- Layer 1
  - Block connections from zombies and other spambots
  - Single process
  - 90% of all spams
- Layer 2
  - Complex SMTP access checks
  - Postfix SMTP server, policy daemons, Milter applications
- Layer 3
  - Light-weight content inspection
  - header_checks, body_checks
- Layer 4
  - Heavy-weight content inspection with external content filters

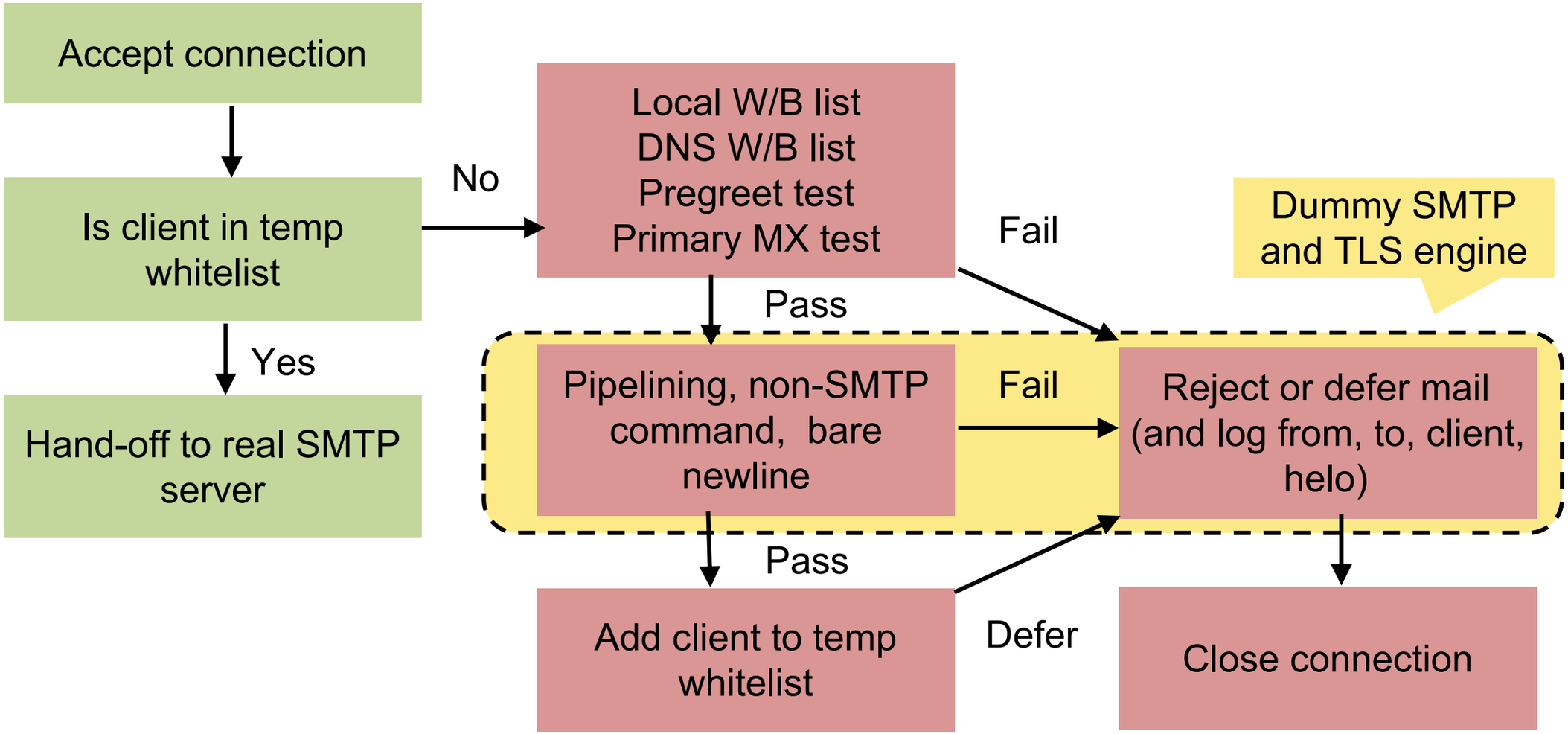
postscreen – Tests after greeting – (1)

- "Deep protocol" tests
  - Use an SMTP protocol engine built into postscreen
  - When a good client passes the tests
    - Add the client to the temporary whitelist
    - CAN***NOT*** hand off the live connection to the SMTP server
    - Reply 4xx status
  - Built-in SMTP engine does ***NOT*** implement
    - AUTH
      - May be added in the feature
      - (Workaround) Not enable tests after greeting
      - (Workaround) End-user should connect directly to the submission service
    - XCLIENT
    - XFORWARD

postscreen – Tests after greeting – (2)

- Command pipelining test
  - Not announce support for ESMTP command pipelining
    - postscreen_pipelining_enable
    - postscreen_pipelining_action = enforce
- Non-SMTP command test
  - Block clients sending commands in postscreen_forbidden_commands
    - postscreen_non_smtp_command_enable
    - postscreen_non_smtp_command_action = drop
- Bare newline test
  - Block clients whose sending lines ended with '\n' instead of '\r\n'
    - postscreen_bare_newline_enable
    - postscreen_bare_newline_action = ignore

postscreen – Workflow before/after SMTP



postscreen – Other errors

- Too many connections
  - postscreen_client_connection_count_limit = $smtpd_client_connection_count_limit = 50
    - **NOQUEUE: reject: CONNECT from** *[address]:port*: **too many connections**
  - postscreen_pre_queue_limit = $default_process_limit = 100
    - **NOQUEUE: reject: CONNECT from** *[address]:port*: **all server ports busy**
- Others
  - **HANGUP after** *time* **from** *[address]:port* **in** *test name*
  - **COMMAND TIME/COUNT/LENGTH LIMIT from** *[address]:port*

postscreen – When all tests succeed

- Create a temporary whitelist entry
  - Controlled with the postscreen_*_ttl parameters
  - **PASS NEW** *[address]:port*
- No "deep protocol tests"
  - Hand off the "live" connection to the SMTP server
  - The client can continue as if postscreen never existed
- When using "deep protocol tests"
  - Reply 4xx
  - Log helo, sender, and recipient
  - Mitigate the impact by giving long TTL

postscreen – Turning on – (1)

- In master.cf

```
#smtp          inet  n       -       n       -       -       smtpd
smtp           inet  n       -       n       -       1       postscreen
smtpd          pass  -       -       n       -       -       smtpd
dnsblog        unix  -       -       n       -       0       dnsblog
#tlsproxy      unix  -       -       n       -       0       tlsproxy
#submission    inet  n       -       n       -       -       smtpd
```

- Original smtp: smtpd → postscreen
- New smtpd: smtpd
  - Handle SMTP connections handed off by postscreen
- New dnsblog: dnsblog
  - DNSBL/WL lookups
- New tlsproxy: tlsproxy
  - Support STARTTLS
  - The implementation led to the discovery of a new class of vulnerabilities
- New submission: smtpd
  - Listen on 587, and wait for MUAs

postscreen – Turning on – (2)

- Blocking mail with postscreen
  - postscreen_blacklist_action
  - postscreen_greet_action
  - postscreen_dnsbl_action

  - For testing postscreen functionality
    - soft_bounce=yes
    - In master.cf
      - -o soft_bounce=yes

Rspamd

- Rapid spam filtering system (https://rspamd.com)
  - https://github.com/rspamd/rspamd
  - https://rspamd.com/features.html
  - https://rspamd.com/comparison.html
  - https://rspamd.com/doc/integration.html
  - https://www.rspamd.com/doc/modules/antivirus.html

```
#smtpd_milters = unix:/var/lib/rspamd/milter.sock
# or for TCP socket
smtpd_milters = inet:localhost:11332

# skip mail without checks if something goes wrong
milter_default_action = accept

# 6 is the default milter protocol version;
# prior to Postfix 2.6 the default protocol was 2.
# milter_protocol = 6
```