Advanced Mail

hyili

Introduction

- What is Email SPAM?
  - Also known as junk email
  - Ex. Phishing mail, malware mail, and unsolicited email
- Problem of SPAM
  - In 2016, Over 50% of E-mails are SPAM!
- How to detect?
  - Client-based detection
  - Content-based detection
- Email Spoofing

Introduction – Client-based detection

- Spammer detection
  - Actually detect who is sending SPAM
- Rely on IP, domain name, or Email address to identify
  - Open relay servers
  - Zombie servers
  - Known spammers
  - Known proxy servers
  - …
- For example
  - Greylisting
  - DNSBL
  - RBL

Introduction – Content-based detection

- Spam detection
  - Actually detect if an email is SPAM or not
  - Rely on the email content to identify
  - Pattern of advertising
  - Malware pattern
  - …
- For example
  - Anti-Spam scan
  - Anti-Virus scan
  - …Machine learning

Introduction – Email Spoofing

- Sender information of the email can be spoof without check by default.
- Spammers may pretent you to send email.
- Countermeasure
    - SPF
    - DKIM
    - DMARC

Overview

- The following techniques are some (new) tools for an administrator to fight with spammers:
  - Greylisting
  - DNSBL
  - RBL
- The following is techniques for prevent Email Spoofing:
  - SPF
  - DKIM
  - DMARC

Greylisting

- Greylisting is a client-based method that can stop mails coming from some spamming programs.
- Behavior of different clients while receiving SMTP response codes

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

  - While spammers prefer to send mails to other recipients rather than keeping log and retrying later, MTAs have the responsibility of retrying a deferred mail.

Greylisting – Idea and Workflow

- Idea of greylisting:
  - Taking use of 4xx SMTP response code to stop steps of spamming programs.
- Steps:
  - A database to store (recipient, client-ip) pair.
  - 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.

Greylisting – Tool

- Tool: mail/postgrey (port or pacakge)
  - A policy service of postfix.
  - Daemon-based, like amavisd

Greylisting – Enable Greylisting and Configuration

- Setup
  - In /etc/rc.conf

```
postgrey_enable="YES"
```

  - service postgrey start
  - Run on TCP port 10023 by default
  - In main.cf

```
smtpd_recipient_restrictions = permit_mynetworks,
                                permit_sasl_authenticated,
                                reject_unauth_destination,
                    check_policy_service inet:127.0.0.1:10023
```

  - Reload Postfix

Greylisting – Log and Others

- When a mail is reject by postgrey, you can find it in /var/log/maillog

```
450 4.2.0 <hyili@cs.nctu.edu.tw>: Recipient address rejected: Greylisted, see
http://postgrey.schweikert.ch/help/cs.nctu.edu.tw.html (in reply to RCPT TO command)
```

- Whitelist Configuration
  - /usr/local/etc/postfix/postgrey_whitelist_clients
  - /usr/local/etc/postfix/postgrey_whitelist_recipients

Greylisting
– Problem of Greylisting

- It cannot handle the domain which has large server farms (MSA pools) without using white list.
    - Microsoft Exchange Online Office 365
    - Gmail
    - Outlook
    - …

Sender Policy Framework (SPF)

- A client-based method to detect whether a client is authorized or not.
- Checking for smtp.mailfrom (Return-Path)

Sender Policy Framework (SPF) – Idea and Workflow

- Idea of SPF
  - Using DNS TXT record to provide authorized server list for the query domain.
- Steps
  - A MTA connects to the server and sends an email.
  - Take the email's smtp.mailfrom's domain (ex. hyili@hyili.idv.tw) and the MTA's ip.
  - Query the domain's TXT record for authorized server list.
  - Check if that MTA is authorized to send email as hyili.idv.tw and see how to handle the email.

SPF Record Syntax – Tool

- Tool: mail/postfix-policyd-spf-perl (port or package)
  - A policy service of postfix.
  - Daemon-based, like amavisd

SPF Record Syntax
– Enable SPF Check in Postfix

- Setup
  - In /usr/local/etc/postfix/main.cf

```
spf-policy_time_limit = 3600
    smtpd_recipient_restrictions = permit_mynetworks,
            permit_sasl_authenticated,
            reject_unauth_destination,
            check_policy_service unix:private/spf-policy
```

  - In /usr/local/etc/postfix/master.cf

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

  - Reload Postfix
  - A policy service of postfix.
  - Daemon-based, like amavisd

Sender Policy Framework (SPF) – Backward Compatibility

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

```
spf=neutral (google.com: 140.131.188.43 is neither permitted nor denied by best guess record for
domain of student@hyili.idv.tw) smtp.mailfrom=hyili@hyili.idv.tw;
```

- Comparative result – when SPF record available.

```
spf=pass (google.com: domain of hyili@hyili.idv.tw designates 140.131.188.43 as permitted sender)
```

SPF Record Syntax – Mechanisms (1/3)

- 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>

SPF Record Syntax – Mechanisms (2/3)

- a
  - a
  - a/<prefix-length>
  - a:<domain>
  - a:<domain>/<prefix-length>
- mx
  - mx
  - mx/<prefix-length>
  - mx:<domain>
  - mx:<domain>/<prefix-length>

SPF Record Syntax – Mechanisms (3/3)

```
v=spf1 a mx ~all
```

- ptr
  - ptr
  - ptr:<domain>
- exists
  - exists:<domain>
- include
  - include:<domain>
  - Also lookup record from <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**.

SPF Record Syntax – Qualifiers & Evaluation

- Qualifiers
  - + Pass (default qualifier)
  - - Fail
  - ~ SoftFail
  - ? Neutral

```
v=spf1 a mx ~all
```

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"

SPF Record Syntax – Qualifiers & Evaluation

```
v=spf1 a mx ~all
```

- Evaluation
  - Mechanisms are evaluated in order: (first match rule)
    - 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”

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"

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

```
v=spf1 redirect=cs.nctu.edu.tw
```

- redirect
  - redirect=<doamin>
  - When mail server is outside from my 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.

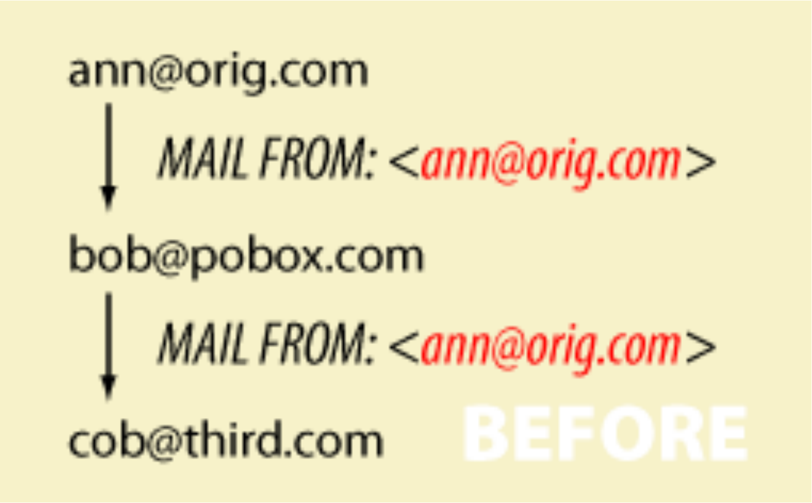
SPF Record Syntax – Modifier

```
v=spf1 mx a
exp=error.hyili.idv.tw
```

- exp
  - exp=<doamin>
  - Explaination
  - 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 a customized explanation.

Sender Policy Framework (SPF) – SPF and Forwarding

- What will happened if SPF meet mail forwarding?



ann@orig.com

MAIL FROM: <ann@orig.com>

bob@pobox.com

MAIL FROM: <ann@orig.com>

cob@third.com

Sender Policy Framework (SPF) – SPF and Forwarding

- If the email is forwarded without SRS

```
220 csmailer.cs.nctu.edu.tw ESMTP Postfix
MAIL FROM: hyili@cs.nctu.edu.tw
250 2.1.0 Ok
RCPT TO: hyili@hyili.idv.tw
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
SRS testing mail
.
250 2.0.0 Ok: queued as C3D9A18DB1
```

```
spf=softfail (google.com: domain of transitioning hyili@cs.nctu.edu.tw does not designate
140.131.188.43 as permitted sender) smtp.mailfrom=hyili@cs.nctu.edu.tw
```

- cs.nctu.edu.tw => hyili.idv.tw(140.131.188.43) => google.com

Sender Policy Framework (SPF) – Enable Sender Rewrite Scheme

- Tool: mail/postsrsd
- Setup
  - In /usr/local/etc/postfix/main.cf

```
sender_canonical_maps = tcp:127.0.0.1:10001
sender_canonical_classes = envelope_sender
recipient_canonical_maps = tcp:127.0.0.1:10002
recipient_canonical_classes = envelope_recipient,header_recipient
```

  - In /etc/rc.conf

```
postsrsd_enable=”YES”
postsrsd_flags=”...”
```

- Start postsrsd service
- Reload postfix

DomainKeys Identified Mail (DKIM)

- A content-based method to verify the source of a mail (with only few computation cost.)
- Checking for the connected MTA's domain

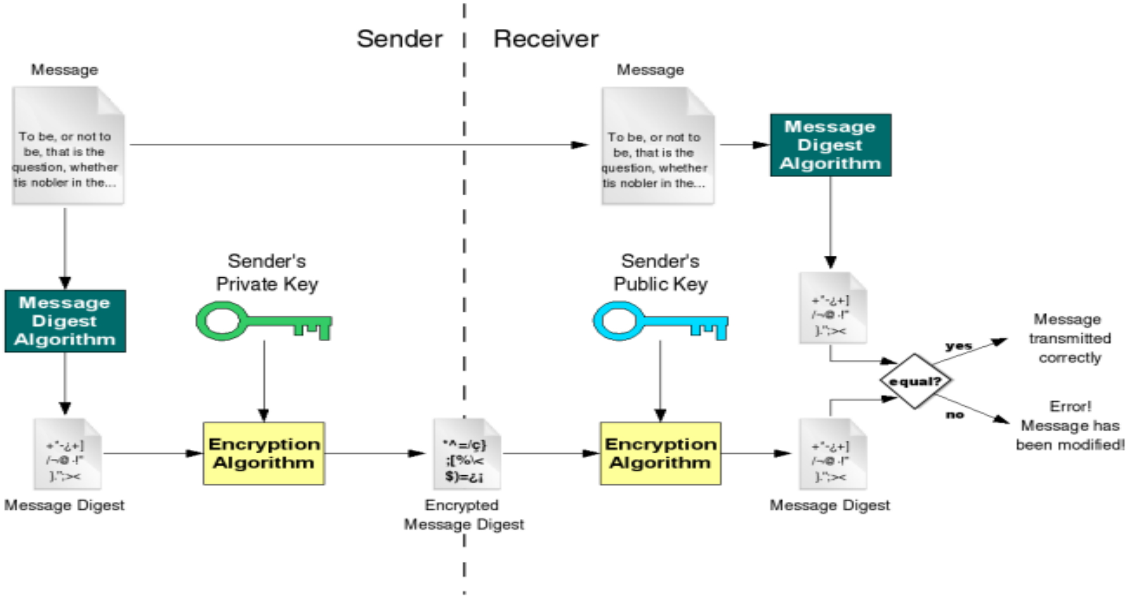
DomainKeys Identified Mail (DKIM) – Goals

- Validate message content itself
- Transparent to end users
  - No client User Agent upgrades *required*
  - But extensible to per-user signing
- Allow sender delegation
  - Outsourcing
- Low development, and use costs
  - Avoid large PKI, new Internet services
  - No trusted third parties (except DNS)

DomainKeys Identified Mail (DKIM) – Idea

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

※ Digital signatures



DomainKeys Identified Mail (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 (outgoing) or improperly signed mail (incoming)

DomainKeys Identified Mail (DKIM) – DKIM-Signature header (1/2)

- v= Version
- a= Hash/signing algorithm
- q= Algorithm for getting public key
- d= Signing domain
- i= Signing identity
- s= Selector
- c= Canonicalization algorithm (simple or relaxed)
- 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

DomainKeys Identified Mail (DKIM) – DKIM-Signature header (2/2)

- Example:

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple; d=hyili.idv.tw;  s=2017; t=1493246840;
bh=tlzeNLTwC0Zv4kvvPcSUFZ/AsgR4I2snpljs1thAmE8=; h=To:Subject:Date:From;
b=V+EeBrWY+1EP6fJPRc+jz+F41YL9EqEAUP5aOnktCQ0re+iQhNG2Z02WgSuKT+wY6
            FGQ5zXJfG25GSjxgxmwXB1VmCJUIE3Nv7NmhC54nPyfKh4EZnXs9KwK3XGF2iaBO52
            9kNS2qkEbSFi92+T1VCqGQ8IcMiXU6V/YRm8rNImczrLBAoNyIXu7zISA0Tezaqn2y
            6g7g/H8/VyyVMySzL9Gf70iWCKg4HhsgEAzMCEZHTtyinxXP8D5xH7AB5ec59N40An
            Atgo1+J/EOUg37Ddz/VLWPAYCvQIk4xWOXkaHcPpASImvFR+CRVabAmBqRUWigVEQc
            ZIHRLFc8aQtaUmuMf7jZ1n8Y2dTYWEQJPXY/m0IkWUGwEDbUiUc9W27O3KHt5FGLYs
            YU1bIzxI/M1ZOwRcsbWVIQmxCtcmpsWMcYbbU+WzR6cwftGluWEwyFX9HgZPcLYy8r
            bxvFcj3o2p77eyNxgAZ1ZPAA7pRGCAsSOpcT7gaBRNLgAnrU/0vPyfaWpWIjGia4L9
            JKfBk5rKAHwaLIW+fQzZYQLCdxExWdRsypRizZ7UGi/dSaBNKXUrr4xct5TC/zVhn9
            mP6NxcRYG9iEhb7AICpsE1EVAjoyPmEM/oDugIpIwxikHjhIkSN0Z247YI+r3k6vdg
            DAhS9g/Z4GfnmTqtHmWm1eKI=
```

- DNS query will be made to:

```
2017._domainkey.hyili.idv.tw
```

DomainKeys Identified Mail (DKIM) – Enable OpenDKIM (1)

- Setup
  - In /usr/local/etc/mail/opendkim.conf

```
Canonicalization  relaxed/simple
KeyTable          refile:/var/db/dkim/opendkim.keytable
LogWhy            yes
SigningTable      refile:/var/db/dkim/opendkim.signingtable
Socket            local:/var/run/dkim/opendkim.sock
SyslogSuccess     yes
UserID            opendkim:opendkim
```

DomainKeys Identified Mail (DKIM) – Enable OpenDKIM (2)

- Setup
  - Preparing environment

```
#add user opendkim:opendkim
#add postfix to opendkim group
mkdir -p /var/run/dkim /var/db/dkim
touch /var/db/dkim/opendkim.keytable
touch /var/db/dkim/opendkim.signingtable
chown opendkim:opendkim /var/run/dkim /var/db/dkim
chmod 0755 /var/run/dkim
```

DomainKeys Identified Mail (DKIM) – Enable OpenDKIM (3)

- Setup
  - Generate key file and TXT record

```
export domain=hyili.idv.tw
export selector=2017
mkdir -p /usr/local/etc/mail/keys/$domain
cd /usr/local/etc/mail/keys/$domain
opendkim-genkey --selector=$selector --domain=$domain --subdomains −b 4096 -v
chown -R opendkim:opendkim /usr/local/etc/mail/keys/$domain
echo "$selector._domainkey.$domain
$domain:$selector:/usr/local/etc/mail/keys/$domain/$selector.private" | tee
/var/db/dkim/opendkim.keytable
echo "*@$domain $selector._domainkey.$domain" | tee /var/db/dkim/opendkim.signingtable
```

DomainKeys Identified Mail (DKIM) – Enable OpenDKIM (4)

- Setup
  - In /etc/rc.conf

```
milteropendkim_enable="YES"
milteropendkim_uid="opendkim”
milteropendkim_cfgfile="/usr/local/etc/mail/opendkim.conf"
```

  - In /usr/local/etc/postfix/main.cf

```
smtpd_milters = unix:/var/run/dkim/opendkim.sock
non_smtpd_milters = $smtpd_milters
milter_default_action = accept
```

  - Start milter-opendkim service
  - Reload postfix

DMARC

- A client-based method that can provide expand control policy for your domain.
- Checking for header.from (which would be shown as sender in gmail GUI)

DMARC – Idea and Workflow

- Idea of DMARC
  - Like SPF, DMARC using TXT record to list policies.
  - Based on SPF and dkim
- Steps
  - A MTA connects to the server and sends an email.
  - After SPF and DKIM have been done.
  - Take the email's header.from's domain (ex. hyili@hyili.idv.tw).
  - Query _dmarc.hyili.idv.tw's TXT record for domain policies.
  - Check if that MTA is authorized to send email as hyili.idv.tw and see how to handle the email.
  - Decide to inform the domain owner or not.

DMARC – Common Tags

- 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
  - Mandatory and must be the second tag=value pair. Defines the policy the sending MTA advises the receiving MTA to follow.

DMARC – Common Tags

- 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 – Common Tags

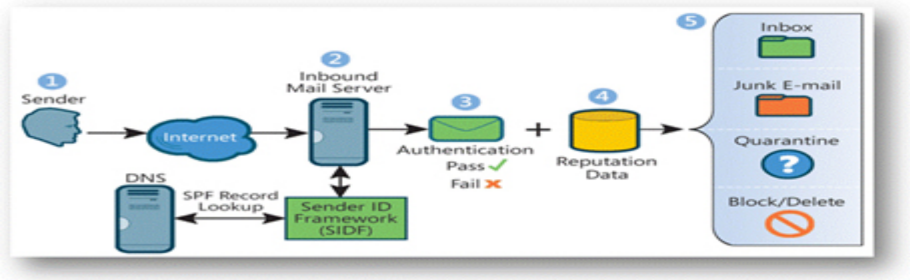
- rua=<@mail>
  - <@mail>: Optional. A comma delimited list of URI(s) to
  - which aggregate mail reports should be sent.
- ruf=<@mail>
  - <@mail>: Optional. A comma delimited list of URI(s) to which detailed failure reports should be sent.
- pct=<percent>
  - <percent>: Number from 0 to 100
  - Optional. Defines the percentage of mail to which the DMARC policy applies.

Advanced Mail

Anything else? Of course!

Sender ID

- RFC4406, 4405, 4407, 4408
- Caller ID for E-mail + Sender Policy Framework (SPF 2.0)
- http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx



Sender ID – paypal.com example

```
knight:~ -lwhsu- dig paypal.com txt

;; ANSWER SECTION:
paypal.com.            3600    IN      TXT     "v=spf1 mx
include:spf-1.paypal.com  include:p._spf.paypal.com
include:p2._spf.paypal.com  include:s._spf.ebay.com
include:m._spf.ebay.com  include:c._spf.ebay.com
include:thirdparty.paypal.com  ~all"
paypal.com.            3600    IN      TXT     "spf2.0/pra mx
include:s._sid.ebay.com  include:m._sid.ebay.com
include:p._sid.ebay.com  include:c._sid.ebay.com
include:spf-2._sid.paypal.com
include:thirdparty._sid.paypal.com  ~all"
```

Other MTA?

- qmail
- exim
- Sendmail X
  - http://www.sendmail.org/sm-X/
- MeTA1
  - http://www.meta1.org/