Sender Policy Framework

Preventing E-Mail Sender Address Forgery

I)ruid <druid@caughq.org>

Problem: Sender Address Forgery

- Nearly all abusive email messages carry fake sender addresses
- Victims of forged sender addresses are the people who's addresses are being abused:
  - Damages victims' reputations
  - Victims often receive bounce messages to messages that were allegedly sent by them, but weren't.
- Sender address forgery is a threat to everyone:
  - Erodes confidence in email's authenticity and reliability

Forged Email Origins

- ***Spammers*** want to avoid receiving DSN (delivery status notifications) reporting non-delivery to their real addresses
- ***Fraudsters*** want to cover their tracks and remain anonymous
- ***Email worms*** want to cause confusion or just don’t care about which sender addresses they use
- ***Phishers*** want to impersonate well-known, trusted identities in order to steal sensitive information from targets

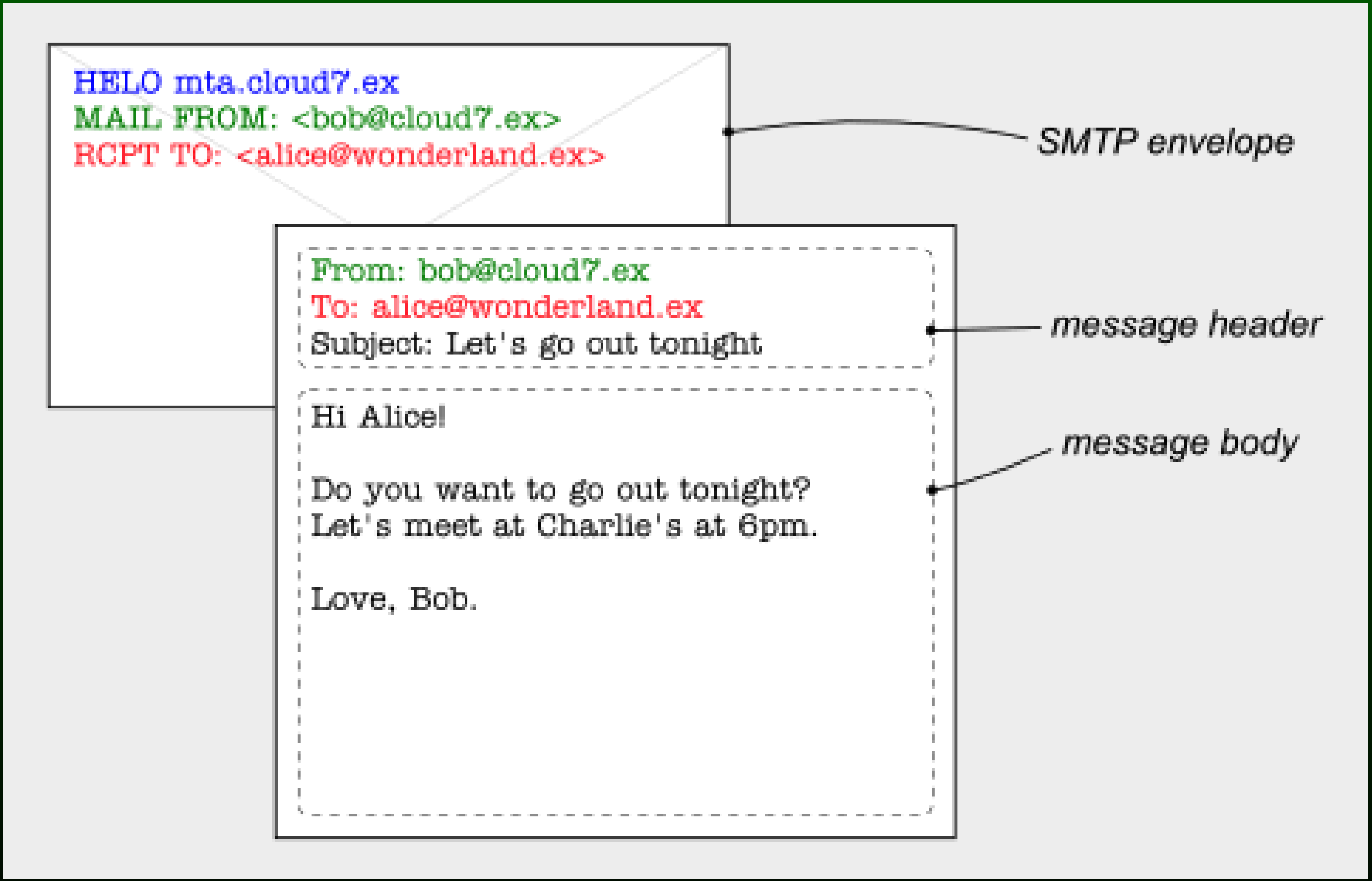
Solution: SPF

- Sender Policy Framework (RFC-4408)
- Open Standard
- Technical method to prevent sender address forgery
- RFC-2821 layer, or, SMTP layer protocol
- Protects the *envelope sender address*

Sender Addresses in Email

- Envelope Sender Address (see RFC-2821)
  - Contained in the “MAIL FROM” and “HELO” SMTP commands
  - Usually stored in the “Return-Path” email header
  - Used during transport of a message between mail servers
  - Used to return the message to the sender in case of delivery failure
  - Usually not displayed to the user by mail programs
- Header Sender Addresses (see RFC-2822)
  - Contained in the “From” or “Sender” email headers
  - Is displayed to users by mail programs
  - Generally, mail servers don’t care about this address; it’s not relevant to delivery

Envelope vs. Header Illustrated



HELO mta.cloud7.ex
MAIL FROM: <bob@cloud7.ex>
RCPT TO: <alice@wonderland.ex>

SMTP envelope

From: bob@cloud7.ex
To: alice@wonderland.ex
Subject: Let's go out tonight

message header

Hi Alice!

Do you want to go out tonight?
Let's meet at Charlie's at 6pm.

Love, Bob.

message body

What does SPF do?

- Allows the owner of a domain to specify which mail servers are allowed to send email from their domain
- Restores confidence in the origin of email messages from those domains

How does it work?

- The domain owner publishes an SPF record in DNS identifying authorized sending mail servers for their domain
- When a mail server receives a message claiming to be from that domain:
  - It looks up the sending domain's SPF record in DNS
  - Checks to see if the sending server is authorized by the sending domain's policy
  - If the message comes from an unauthorized server, it can be considered a forgery

It Takes Two to Tango

- Published domain sender policies in DNS are not worth much alone...
- Receiving mail servers still have to enforce them

Example Policy

“v=spf1 mx a:druid.example.net include:gmail.com -all”

- v=spf1
  - SPF Version 1 TXT record identifier
- mx
  - The incoming mail servers (MXes) of the domain are authorized to also send mail for example.net
- a:druid.example.net
  - The server druid.example.net is authorized also
- include:gmail.com
  - Everything authorized by gmail.com is also authorized for example.net
- -all
  - All other servers are **NOT** authorized (note the “-” sign)

SPF Record Syntax

- Many more *mechanisms* available than those shown in the previous example, including:
  - ptr
  - ip4
- Each *mechanism* can be prefixed with one of four qualifiers:
  - - fail
  - ~ softfail
  - + pass
  - ? neutral

Evaluating an SPF Record

- Works like a firewall policy:
  - Evaluate mechanisms in order from first to last
  - If the mechanism results in a match, it’s prefix value is used (default is pass (+))
  - If no mechanism or modifier matches, the default result is neutral
  - Most SPF records end in a catch-all rule called “all”
    - Prefixed with a “+” (+all), this rule is an ALLOW all
    - Prefixed with a “-” (-all), this rule is a DENY all

Example Evaluation

“v=spf1 a mx a:druid.example.net -all”

- a
  - (+a) Does the sending server match the domain’s A record?
  - PASS
- mx
  - (+mx) Does the sending server match one of the domain’s MX records?
  - PASS
- a:druid.example.net
  - (+a) Does the sending server match this particular A record?
  - PASS
- -all
  - Does the sending server match everything (all)?
  - FAIL

Drawbacks

- Most servers do not yet support SPF checking natively
- In the special case of mail-forwarding MTA’s, SPF requires that the sender address be rewritten

Benefits

- Most MTAs, both commercial and open-source, have SPF extensions available
- Sender Rewriting Scheme (SRS) has been developed specifically for mail-forwarding MTA's, and was discovered to have the additional benefit of being able to identify illegitimate DSNs
- An SPF check can be performed before any message data is sent to the receiving mail server (checks vs. MAIL FROM)

Why do I care about SPF?

- Because I helped design Version 1
  - Eat your own dog food... or something...
- It works
  - I prevent a couple thousand SPF FAIL messages from entering my server weekly
- It's quick and easy to deploy
  - Even my unmotivated ass was able to get it implemented!

Who else cares about SPF?

- AOL is requesting all of their whitelist partners switch to SPF to remain on their whitelist.
- SpamAssassin (among many, many other anti-spam tools) uses SPF as one of it’s weighted tests alongside RBLs and other metrics

SPF vs. Sender-ID

- Sender-ID (RFC-4406) is Microsoft’s abomination of SPF
- RFC-2822 layer, or, Header Layer protocol
- Validates a header sender address (purported responsible address (RFC-4407))
- Thank god we talked them out of using XML in DNS...
- But now they use SPF record syntax... confusing!
- Sender-ID spec. directly violates the SPF spec.
- Microsoft refuses to fix it (go figure)

SPF vs. DKIM

- DomainKeys Identified Mail (DKIM)
- Merger of Yahoo! DomainKeys and Cisco’s IIM
- RFC-2822 layer, or, Header Layer protocol
- Validates an accountable identity associated with a message when it is transferred over the Internet
- Cryptographically signs the email body and *some* of the headers
- Domain public key stored in DNS TXT record under _domainkey subdomain

Everything You'll Need

http://www.openspf.org