Mail Avenger

David Mazières

New York University

Early design goals for email

- **Work over weakly connected networks**
  - E.g., early Internet, UUCP, etc.
  - Move mail closer to recipient whenever you can…
  - Because sender might not be available later on
- **Provide reliable transmission and delivery:**
  - "When the receiver-SMTP accepts a piece of mail… it is accepting responsibility for delivering or relaying the message. It must take this responsibility seriously… If there is a delivery failure after acceptance of a message, the receiver-SMTP MUST formulate and mail a notification message." – RFC 2821

Architectural consequences

- **Any random host can *send* email**
  - Dynamic/temporary IP address or NAT is just fine
  - No authentication, as any host may relay for any other
  - Don't even need your own domain name; just forge it
- **Only well-established servers can *receive* mail**
  - Need permanent domain name & listening TCP port
  - Anyone can identify the server for a recipient address
- **Servers must treat received mail as precious**
- **Surprise: Senders are abusing the system**

Stop the Insanity!

Revisiting email's design goals

- **Should email be reliable?**
  - **Yes!** People still count on reliable email delivery
  - Yet reliability is often a casualty of spam filtering
  - Even if stock filters happen to work on *your* mail...

    "Most people can safely delete e-mail with subject lines like 'small dick,' 'anal-to-mouth action,' or 'lesbian-animal sex.' Not me. I have to open those because they could be legit... questions that touch on those distressing topics."

    – Dan Savage, advice columnist
- **Should we accommodate weakly-connected, ephemeral clients?**
  - **No!** Not unless they're *your, authenticated* clients

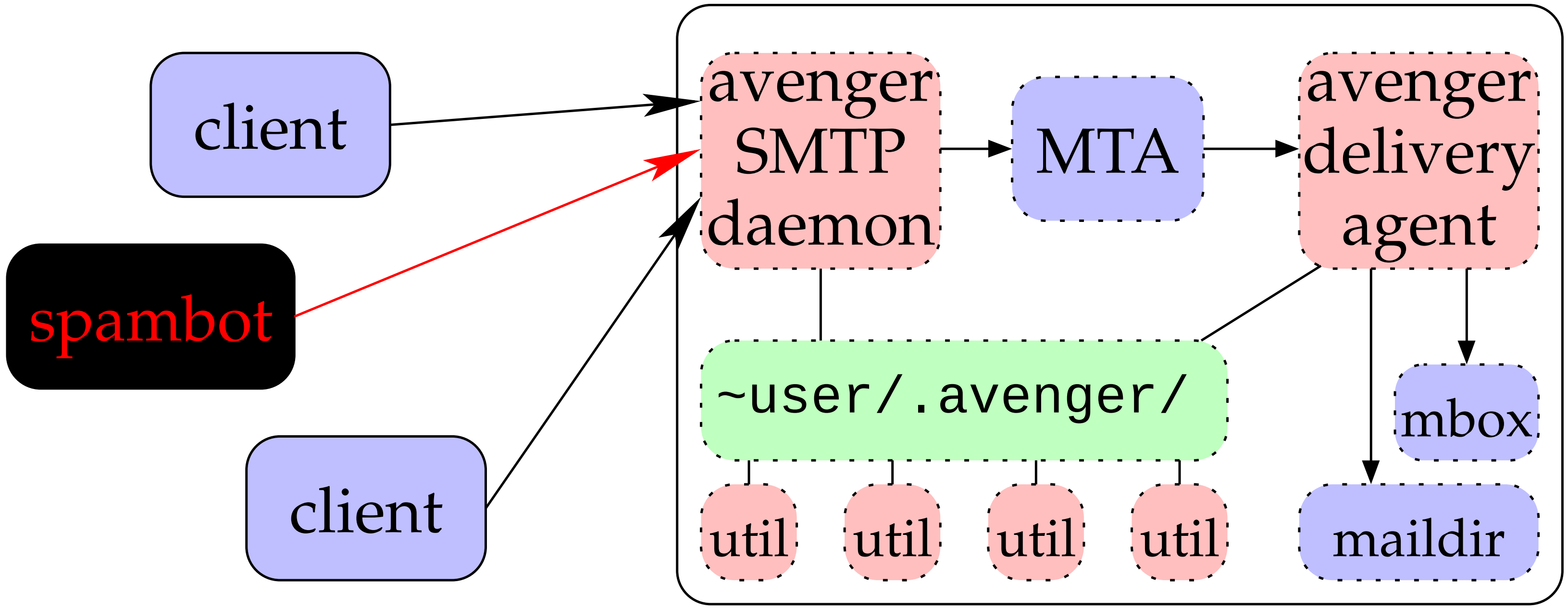
Principles

- **Never accept email until you're sure the sender can receive a bounce.**
- **Never perform spam filtering *after* accepting responsibility for a message from a client.**
    - Corollary: Filter at your organization's outermost mail relay
- **Different mailboxes need different mail acceptance policies.**
- **Individual users should be able to have multiple mailboxes with different policies.**
- **Make it easy to implement these new policies.**
    - Give users all possible information about incoming mail

Mail Avenger

- **Email transmitted using SMTP protocol**
  - MAIL FROM – client specifies sender address
  - RCPT TO – client specifies recipient
  - DATA – client sends body of mail message
- **Idea: Put recipients in control of SMTP responses**
  - Allow RCPT or DATA to succeed, fail, or return temporary error based on recipient's policy
- **Give users *extension addresses***
  - I.e., user `dm` sets policy for `dm+`*`anything`*`@mailavenger.org`
  - Can break policy into multiple files, just like qmail MTA
- **Easy to implement new policies**
  - Policy specification is just a shell script

Avenger architecture



- **SMTP daemon (*asmtpd*) enforces users' policies**
- **Delivery agent (*avenger.local*) handles extensions**
- **Useful utilities for use in policy shell scripts**
- **Uses existing MTA (sendmail, qmail, postfix, . . .)**

asmtpd checks

- **Check bounce addresses with DNS SPF records**
  - Can quickly reject forged mail "from" SPF-enabled domains
- **Check bounce addresses with SMTP servers**
  - Use *SMTP callbacks*
  - Start to send bounce, but stop after RCPT (no DATA)
  - If sender's server returns temporary/permanent error, do the same
- **Collect network-level information about client**
  - "SYN fingerprint" – usually identifies client OS
  - network route – identifies BGP-hijacked address space
- **Collect info on client's SMTP implementation**
  - E.g., eager pipelining, invalid "POST" command, ...

Avenger scripts

- **Policy scripts in user home directories**
  - `dm@host.tld` $\Longrightarrow$ `~dm/.avenger/rcpt`
  - `dm+ext@host.tld` $\Longrightarrow$ `~dm/.avenger/rcpt+ext`
  - Also `rcpt+default` catch all
- **Environment variables contain client information**
- **Script augmented with shell functions**
  - `accept` – RCPT command succeeds immediately
  - `reject` – RCPT command fails immediately
  - `defer` – RCPT fails w. temporary error
  - `bodytest` – specify script to run on DATA
  - Or fall through to default, or redirect to other user

Example: Preventing “Joe Jobs”

- **Problem: Viruses forge your email address**
  - You get tons of unwanted bounce messages
- **Solution: Reject bounces to your main address**
  - `macutil` utility generates temporary cookies
  - setenv MACUTIL_SENDER `dm+bounce+*@host.tld`
  - Send mail with `macutil --sendmail` (sendmail wrapper)
- `~/.avenger/rcpt:`

```
test -z "$SENDER" && reject "no bounces, please"
```

- `~/.avenger/rcpt+bounce+default:`

```
macutil --check "$SUFFIX" \
        || reject "<$RECIPIENT>.. user unknown"
```

Example: List-specific addresses

- **Want to subscribe to mailing lists at NYU & MIT**
  - But don't want your address passed on to others
- **Use SPF as a policy language to check client**
  - To reduce latency SPF and DNS requests are asynchronous
  - `setvars` command waits for them to complete
- **for** `dm+list@host.tld`**, use** `~/.avenger/rcpt+list`**:**

```
spf EDU_OK ptr:nyu.edu ptr:mit.edu mx:cs.nyu.edu/24
setvars
test "$EDU_OK" = pass && accept
test "$EDU_OK" = error && defer "Temp. DNS error"
reject "Address for NYU/MIT clients only"
```

Other Examples

- “Greylist” mail from Windows machines

```
match -q "*Windows*" "$CLIENT_SYNOS" && greylist
```

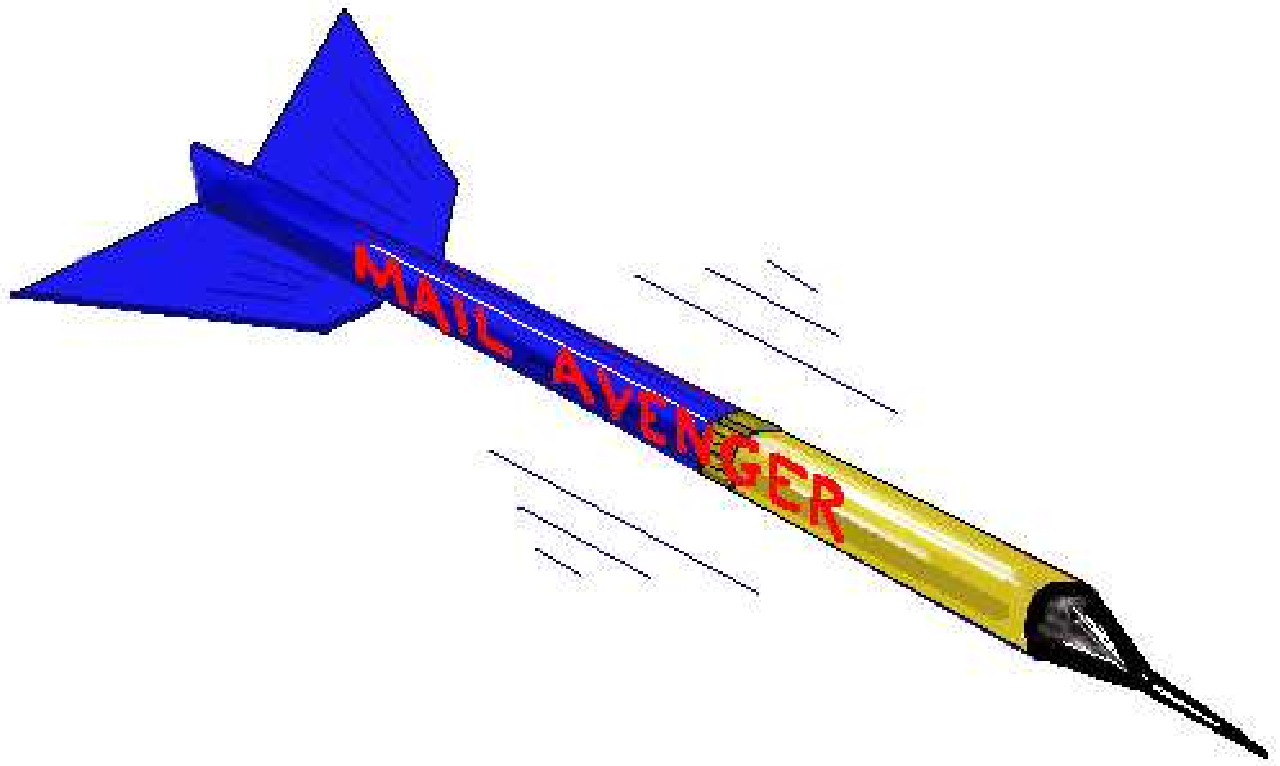
- Run spamassassin during SMTP session

```
errcheck
bodytest edinplace spamassassin -e 100
```

Conclusions

- **Filter spam before assuming responsibility for messages**
- **Don’t accept mail if sender won’t accept bounce**
  - Easy to originate TCP connections with viruses
  - Harder to set up domain and mail server to accept bounces
  - SPF adoption can prevent forgery…
  - and SMTP callbacks can encourage SPF adoption
- **Different recipients need different policies**
  - Individual users may even need multiple addresses
- **Implementing policies is easy with Mail Avenger**

Download it!



Mail Avenger is free software.

`http://www.mailavenger.org/`