

Recipient Empowered Email

Lucas Fisher, Thomas Heinis, Dan Noland
{ljfisher,theinis,nolandd}@purdue.edu

Overview

- ◆ Problem:
 - ◆ The sender has most of the power.
 - ◆ How can we give recipients power over the type of email they receive?
 - ◆ Must be compatible with existing mail system.
- ◆ Solution:
 - ◆ Require senders to state a policy their mail abides by.
- ◆ Inspired by Tripoli concept from People For Internet Responsibility
- ◆ Implement the solution using the Exim mail server and Chord

Goals

- ◆ Allow recipient to identify category of email message.
- ◆ Allow recipient to create policy that defines what email should be accepted.
- ◆ Senders assert attributes about the mail they send.
- ◆ Enforce the sender asserted attributes.
- ◆ Reject messages which violate recipient's policy with minimal use of resources.

Related Work

- ◆ Statistical and heuristic filters
 - ◆ Spam Assassin, CRM114
- ◆ Anti-spoofing technologies
 - ◆ Microsoft Caller-Id, Yahoo DomainKeys, SPF
- ◆ Sender pays
 - ◆ CAMRAM, Microsoft Penny Black Project
- ◆ Legislation
 - ◆ CAN-SPAM, various state laws

Internet Mail Background

- ◆ Internet Message Format (RFC 2822)
 - ◆ ASCII
 - ◆ Header fields: subject, to, from, date, received
 - ◆ Body
- ◆ Mail User Agent (MUA)
 - ◆ mail client
 - ◆ Outlook, Pine, Webmail
- ◆ Mail Transfer Agent (MTA)
 - ◆ mail server
 - ◆ delivers email to the recipient
- ◆ Simple Mail Transfer Protocol (SMTP)
 - ◆ we will come back to this

Recipient Empowered Email

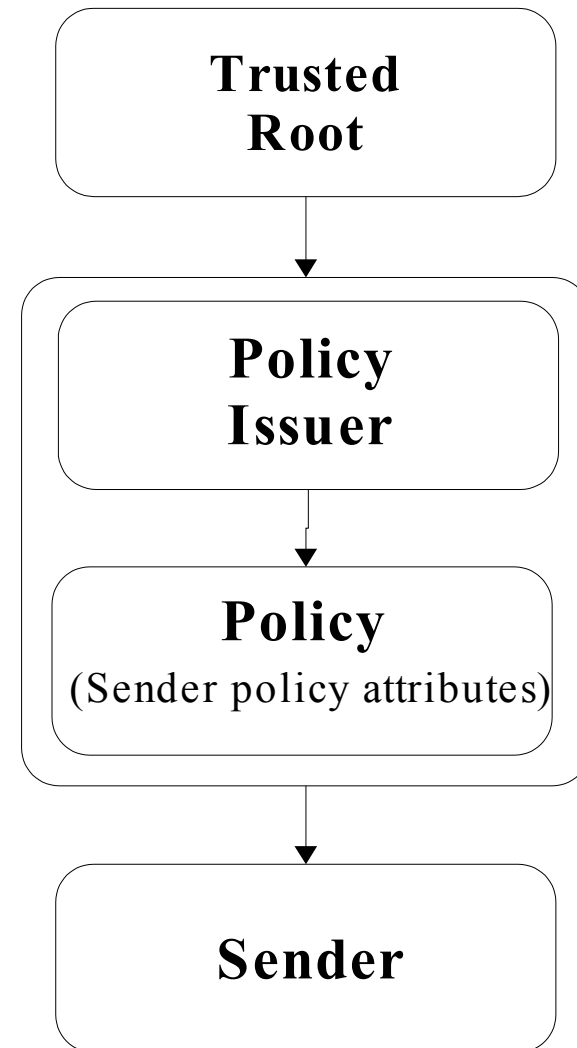
- ◆ ISPs and other third-parties already have policies that describe the type of messages their customers or members can send. Take advantage of this by relating these policies to recipients.
- ◆ Need to:
 - ◆ Define the sender's policy.
 - ◆ Strongly tie policy to a mail message.
 - ◆ Define the recipient's acceptance policy.
 - ◆ Extend existing protocols to support these requirements.
 - ◆ Enforce sender's policies.

Sender Email Policy

- ◆ Sender email policy – set of name/value attributes embedded in a X.509 certificate.
- ◆ Policy issuer – third-party which specifies a policy and provides policy certificates.
- ◆ Policy certificate contains the policy in a X.509v3 extension: senderEmailPolicy
- ◆ Uses existing Internet PKI.

Certificate Hierarchy

- ◆ Each box a certificate.
- ◆ Type is in bold.
- ◆ Policy issuer and policy certificates could be the same.



Payload Information Token (PIT)

- ◆ PIT is a “key” that a sender must present before an MTA will unlock a recipient's mailbox.
- ◆ Keep as small as possible.
- ◆ PIT is sent before the email message using a new SMTP extension.
- ◆ PIT must be resilient to attacks such as theft and replay.

PIT Contents

- ◆ List of email recipients
- ◆ Cryptographic hash of to, from, cc, subject, date headers, message body.
- ◆ URI of the certificate chain used to sign the PIT.
- ◆ Timestamp of signature
- ◆ Example contents:

```
pit-recipients: bob@bobisp.com,alice@aliceisp.com,  
               charles@charlesisp.com  
pit-cert-url:  http://myisp.com/certs/8820bd0d.0  
pit-mdigest:  SHA1:ScJ10VeXbx0yIQ+05mxRR9Dtz98=
```

PIT Encoding and Signing

- ◆ Signed by sender.
- ◆ Need a standard way to specify signature algorithm, signer, value of signature, etc.
- ◆ PKCS#7 from RSA Security
- ◆ Also used by the S/MIME standard.
- ◆ But, this is binary so translate to PEM encoding.
- ◆ Encoded PIT is about 1000 bytes
- ◆ Example...

Acceptance Policies

- ◆ Recipient defined actions on receipt of a message.
- ◆ Three possible actions:
 - ◆ Accept message
 - ◆ Reject a message – MTA drops SMTP connection with sending MTA.
 - ◆ Delay acceptance – MTA delays receipt of message until recipient approves or rejects message policy.

Exim Filters as Acceptance Policies

- ◆ Three new commands:
 - ◆ `policy_accept`, `policy_reject`, `policy_delay`
- ◆ Access to policy attributes through variables
 - ◆ `$policy_<policy attribute name>`
- ◆ Access to reputation ratings
 - ◆ `$sender_reputation`, `$policy_reputation`
- ◆ Example:

```
if $policy_emailtype is "person-to-person" then
    policy_accept
elsif $policy_emailtype is "commercial-advertisement" then
    policy_reject
elsif $policy_emailtype is "non-profit-advertisement" then
    policy_accept
else
    policy_delay
endif
```

Sending the PIT

- ◆ But first, a little about SMTP....
- ◆ Protocol used by MUA to send messages to MTA and between MTAs.
- ◆ Request/Response pattern
- ◆ Example session....

Example SMTP Session

```
220 mail.mydomain.net SMTP Server
HELO myisp.com
250 Hello myisp.com
MAIL FROM: customer@myisp.com
250 OK
RCPT TO: user@mail.mydomain.net
250 OK
DATA
354 Enter message, ending with . on a line by itself
From: customer@myisp.com
To: user@mail.mydomain.net
Subject: what happened?

Hey did you get my last email message?
.
250 OK
QUIT
221 mail.mydomain.net closing connection
```

Sending the PIT: SMTP extension

- ◆ Modified the Exim MTA.
- ◆ STOK – Send the PIT.
 - ◆ Similar to DATA command
 - ◆ Must be given after MAIL FROM and before any RCPT TO command.
- ◆ Responses:
 - ◆ recipient accepted message
 - ◆ recipient rejected message
 - ◆ recipient delayed receipt, try again later
- ◆ Example....

Sending the PIT: Example

```
220 mail.mydomain.net SMTP Server
HELO myisp.com
250 Hello myisp.com
MAIL FROM: customer@myisp.com
250 OK
STOK
356 Send token, ending with . on a line by itself
-----BEGIN PKCS7-----
MIICiAYJKoZIhvcNAQcCoIICeTCCAnUCAQExCzAJBgUrDgMCGGUAMIGdBgkqhkiG
9w0BBwGggY8EgYxwaXQtY2VydC11cmw6IGh0dHA6Ly9teWlzcC5jb20vY2VydHMv
....
DtedaecGkFzMOQGR2RIb3HExfCZrBBgH8829BclXURa1iGiIsj61yQiSC8xKyUPa
Eh3F9paf1EEEM3NNjT+6KRnvTF3t3LKXWrUcJA==
-----END PKCS7-----
.
250 OK
RCPT TO: user@mail.mydomain.net
250 OK
DATA
354 Enter message, ending with . on a line by itself
Hey did you get my last email message?
.
250 OK
221 mail.mydomain.net closing connection
```

Complaints and Reputation

- ◆ Assumed that we trust policy issuers to enforce their policies, but this is not always the case.
- ◆ Complaint filer
 - ◆ recipient sending the complaint
- ◆ Complaint target
 - ◆ sender of the offending message or policy issuer of this sender
- ◆ Supervisor
 - ◆ processes complaints against a sender or policy issuer
- ◆ Offending message
 - ◆ message in which sender lied about policy

Complaints

- ◆ Filer must be a recipient of the offending message.
- ◆ Fields similar to those in PIT:
 - ◆ *pit-signature*: the PIT from the offending message
 - ◆ *pit-cert-url*: certificate chain of the complaint filer
 - ◆ *pit-violations*: list of policy attributes violated by the sender
- ◆ Same encoding method as PIT
- ◆ Signed with the certificate associated with the address at which the filer received the message.

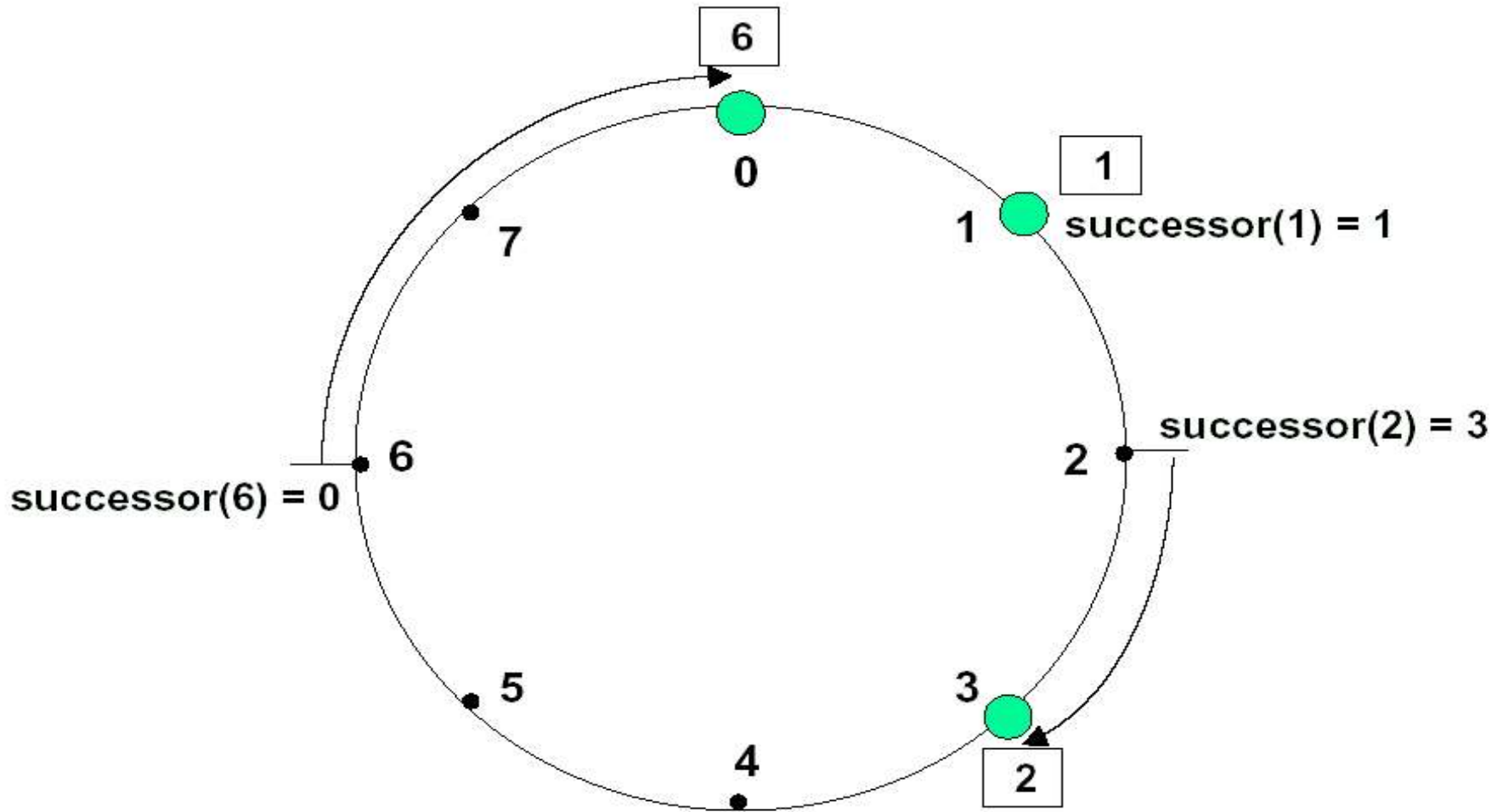
Reputation Service

- ◆ Filer sends complaint to target's supervisors.
- ◆ Each supervisor calculates the reputation based on the complaints it receives.
- ◆ Reputation is number of complaints received.
 - ◆ More complaints means worse reputation
- ◆ Reputation improves over time
- ◆ Recipient's acceptance policy can reject messages based on the reputation of the sender or policy issuer.
- ◆ Implemented using Chord distributed hash table.

Chord Overview

- ◆ A distributed, highly available, well balanced hash table.
- ◆ Each Chord node is assigned an id.
- ◆ Node ids and keys from same set of values.
- ◆ Nodes organized in a circle ordered by id.
- ◆ Storing a value:
 - ◆ Find node id closest to key
 - ◆ Send value to this node, who will also distribute value to N-1 neighboring nodes.
- ◆ Retrieving a value:
 - ◆ Find node id closest to key
 - ◆ Node finds value and returns to client.

Chord Overview (2)



Reputation Service using Chord

- ◆ Chord key: fingerprint of target's private key
- ◆ Chord node = supervisor
- ◆ Storing a complaint:
 - ◆ Find closest node to key and its N neighbors.
Send complaint to each node.
- ◆ Retrieving a reputation rating:
 - ◆ Find closest node to key and its N neighbors.
Get rating from each node.
 - ◆ Final rating is value agreed upon by majority of nodes.

Summary

- ◆ Give email recipients the power to choose what type of message they want to receive.
- ◆ Senders agree to send mail following a given policy.
- ◆ Recipients accept mail based on sender's policy
- ◆ Modified Exim to process PITs
- ◆ Modified Chord to implement a reputation service.

Questions?

