Sieve Filter Rule Metadata

Presenter: Hans-Jörg Happel
happel@audriga.com

IETF 116, Yokohama, JP

Introduction

- Sieve [RFC 5228] is a formal language for email filtering
  - It is used by both end users and machines (e.g., for Spam processing)
  - It is the only notable standardized filter language and it is widely supported

- ManageSieve [RFC 5804] is a HTTP-based protocol which lets users retrieve and store scripts on a server

Sieve usage by end users

- Direct usage (mostly used by expert users)
  - Via direct file access
  - Via ManageSieve

- Indirect usage (default for regular users)
  - Filter editing user interfaces (guide/restrict editing): test/action style
  - Indirect creation (e.g., add to allowlist from email context menu)
  - Special-use filters (Forward, Vacation notice, Spam, Allowed/Blocked sender) → hidden/uneditable in filter editing UI

- Indirect usage is likely the dominant form and it is the focus of this talk

(Indirectly created) Sieve rules in practice

- Underlying Sieve scripts have a special (modularized) structure
  - Typically no “ELSE”, “ELSEIF” usage
  - Restricted depth of rule nesting
- Modular rules have metadata annotated in header comments
  - Name, description
  - Type (Special use)
  - Order
- Modular rules can often be disabled by the user, which is realized as:
  - Commenting out individual rules
  - Wrapping rules into an “AllOf(false, …)” statement
  - Moving a rule into a deactivated script

Issues with the current state of practice

- Interoperability
  - Sieve clients need to support various (undocumented) vendor-specific metadata schemes to avoid messing up scripts (e.g., rule name; messing with special use filter rules)
  - Dealing with deactivated rules

- Data portability
  - Scripts might break in migration scenarios for similar reasons described above
  - In addition, special use types supported might differ between systems

Normative dimension

- Define a standard set of rule comments?
- Define a recommended way of deactivation?
- Define script header comments to identify generator/type of script and supported cases of special use
  - Generator might also be inferred via ManageSieve GREETINGs

Summary and discussion

- Summary
  - Indirectly created, modular Sieve scripts are probably most common
  - Most (?) 3rd party Sieve editors do not address the special format

- Discussion
  - Is a normative part needed?
  - How to obtain more examples from vendors?
  - Do other Sieve usages need to be considered?
  - Could all this be a severe barrier for rule usage?
  - Side topic: does this imply the need for a “Sieve light”?