Rspamdの紹介

ENOG51 Meeting

2018/7/20

創風システム 外山 文規

注意

メールシステムに関する知識があることを前提とした内容となります、一部用語について説明を省くことがあります

Rspamd 1.7.6 時点での話になります

より新しいバージョンにおいて仕様が変わっている場合があります

迷惑メール対策どうしてます？

- クラウド型の迷惑メール対策サービスを利用
- アプライアンス、OSSを利用して自力で運用
- しないありのままを受け入れる

迷惑メール対策どうしてます？

- クラウド型の迷惑メール対策サービスを利用
- アプライアンス、OSSを利用して自力で運用
- しないありのままを受け入れる

OSSな迷惑メールフィルタといったら

Spamassassin(以下SA)

なんですが、

別の実装であるRspamdもみてみようというが
今日のお話

Rspamdなんじゃらほい?

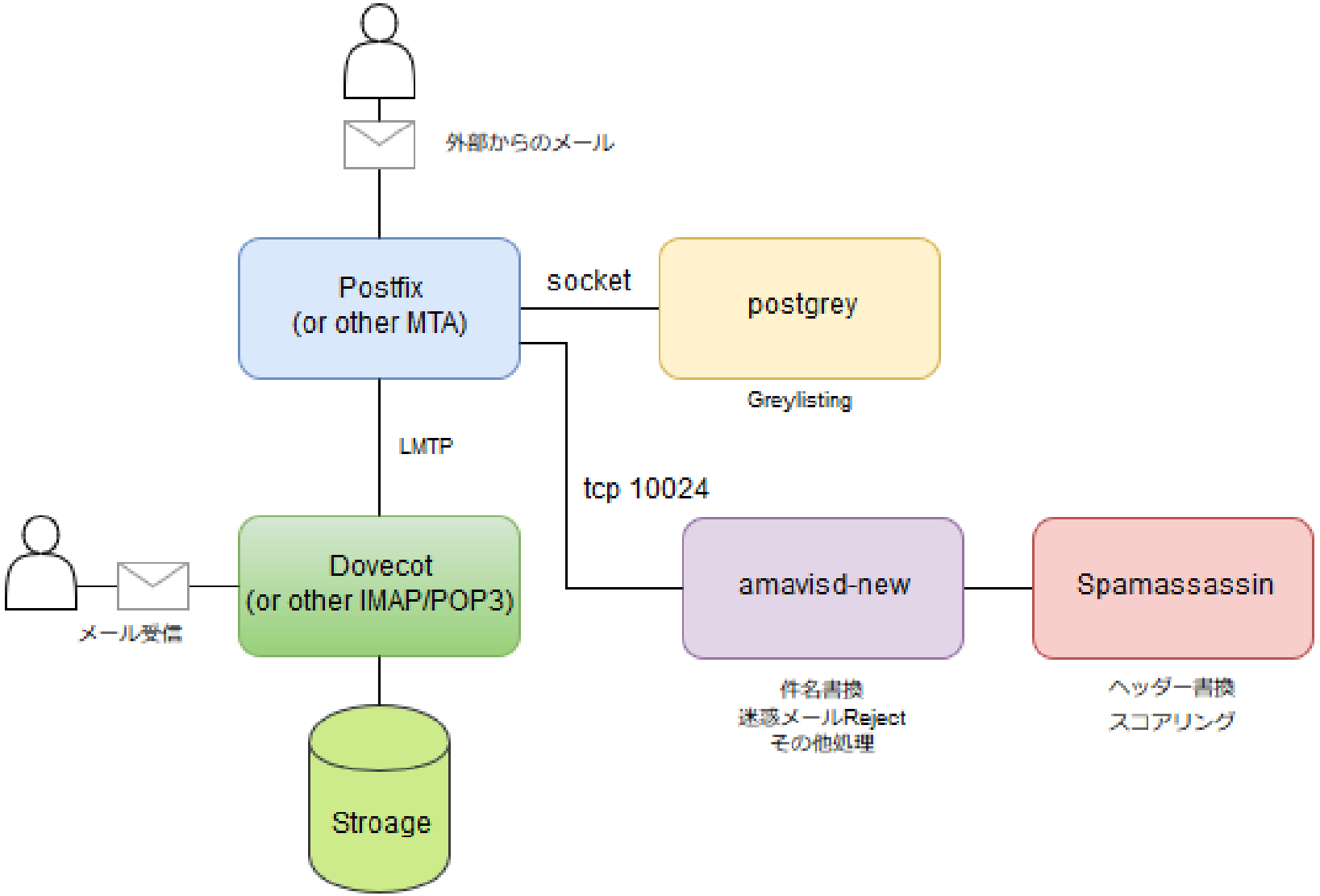
- OSSの迷惑メールフィルタリングシステム
- 既存のMTA（Postfix, etc)のプラグインとして稼働
- 最新は 2018/7/13 ver1.7.8
- 最近少し名前を聞くようになってきた（個人的感想です）
- 開発は活発
  - ✓ 2018/7/2　ver1.7.7
  - ✓ 2018/6/15 ver1.7.6
  - ✓ 2018/5/18 ver1.7.5
  - :
  - ✓ 2018/3/12 ver1.7.0
  - :
  - ✓ 2017/6/2　ver1.6.0

SAと比べたRspamdの特徴

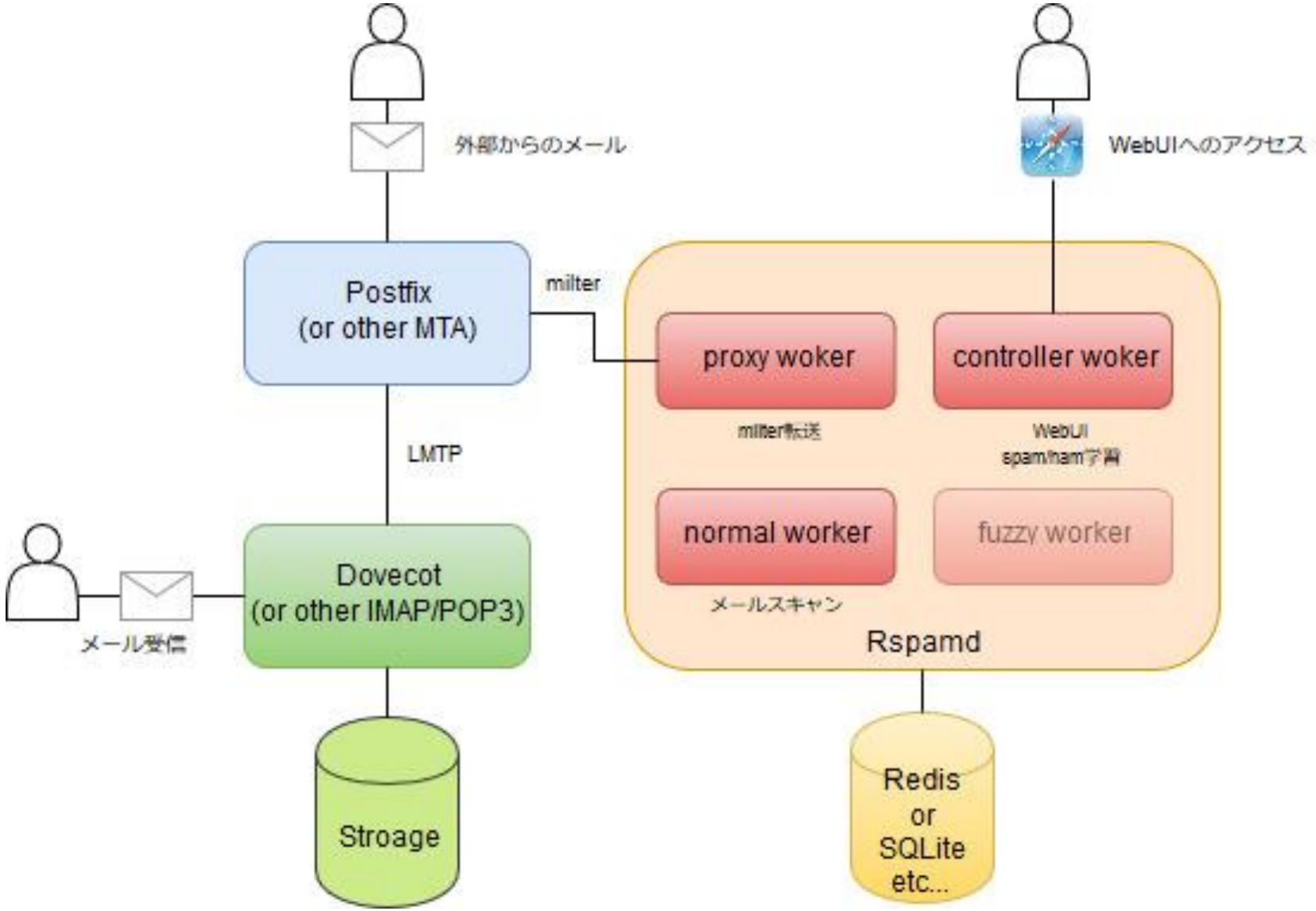
- 複合的な条件でメールを迷惑メールかスコアすることは一緒
- 実装は異なる
- 提供する機能範囲がちょっと違う

  Rspamd ≒ SA+Amavisd-new+Postgrey
- よりパフォーマンスの期待できる実装（C, Lua, etc）
- 大規模構成を想定した設計もされている
- Neural networkやfuzzy storageなどSAにない実装もある

SA+amavisd-new+postgrey



Rspamdの基本的な構成



	RSPAMD	SpamAssassin	DSPAM Statistical Spam Protection
General			
Written in	C	Perl	C
Process model	event driven	pre-forked pool	LDA and pre-forked
MTA integration	milter, LDA, custom	milter, custom (Amavis)	LDA
Web interface	✔ embedded	? 3rd party	✖
Languages support	✔ full, UTF-8 conversion, lemmatization	✖ naïve (ASCII lowercase)	✖ naïve
Scripting support	✔ Lua API	✔ Perl plugins	✖
Licence	Apache 2	Apache 2	GPL
Development status	✔ very active	✔ active	✖ abandoned
Mail filtering features			
Greylisting	✔	✖	✖
Ratelimit	✔	✖	✖
Replies whitelisting	✔	✖	✖
Rules composition	✔	✔	✖

https://rspamd.com/comparison.html

	RSPAMD	SpamAssassin	DSPAM Statistical Spam Protection
Filtering methods			
Regular expressions filtering	✔	✔	✖
DKIM	✔	✔	✖
SPF	✔	✔	✖
DMARC	✔ with reports support	✖	✖
Policies white and blacklists	✔	✔	✖
DNS lists	✔	✔	✖
URL DNS lists	✔	✔	✖
Phishing checks	✔ advanced with external resources	✔ very basic	✖
Custom lists	✔ with dynamic reload	✔	✖
Pyzor/Razor	✖	✔	✖
Own fuzzy storage	✔	✖	✖
DCC	✔	✔	✖
HTML rules	✔ own parser	✔ many regexp rules	✖
PDF filtering	✖	✔	✖

	RSPAMD	SpamAssassin	DSPAM Statistical Spam Protection
Statistical methods			
Bayes classifier	✔ hidden Markov	✔ naïve	✔ hidden Markov
Bayes autolearn	✔ with custom Lua rules	✔ by threshold	✖
Bayes window	5 words	1 word	2 words (5 words in SBPH/OSB mode)
Plain files backend	✔	✔	✔
SQLite3 backend	✔	✔	✔
MySQL backend	✖	✔	✔
Redis backend	✔	✔	✖
Neural networks support	✔ via libfann	✖	✖

※hidden Markovについて
作者のslideshareに上がっているRspamd紹介の資料
https://www.slideshare.net/VsevolodStakhov/rspamdfosdem-57772063
に「 Uses Markov chains for statistic tokens (using sparse bi-gramms model) 」とあり
下記のサイト「【技術解説】bi-gramマルコフモデル」に説明があります
https://mieruca-ai.com/ai/bi-gram_markov_model/

Rspamdを始めるには

1. 前準備
2. インストール
3. 設定
4. 動作確認
5. WebUIの設定

前準備

- MTAのインストールと初期設定
- （ほぼ必須） Redis Serverのインストールと設定
- （推奨）キャッシュDNSのインストールと設定
- （推奨）仮想マシンで動かす場合はSSSE3以上のSIMD拡張命令セットがCPUで利用可能か確認（Hyperscanのため）

【余談】Hyperscan

- オープンソースのパターンマッチングライブラリ
- SSSE3以上のSIMD拡張命令セットを必要とする
- Intel CPUを使ったIPS,DPIの高速スキャン用途に開発

※Hyperscan 参考URL:
https://github.com/intel/hyperscan/
https://www.slideshare.net/VsevolodStakhov/rspamdhyperscan
https://www.intel.co.jp/content/www/jp/ja/communications/hyperscan.html

MTAとの連携

- Postfixの場合

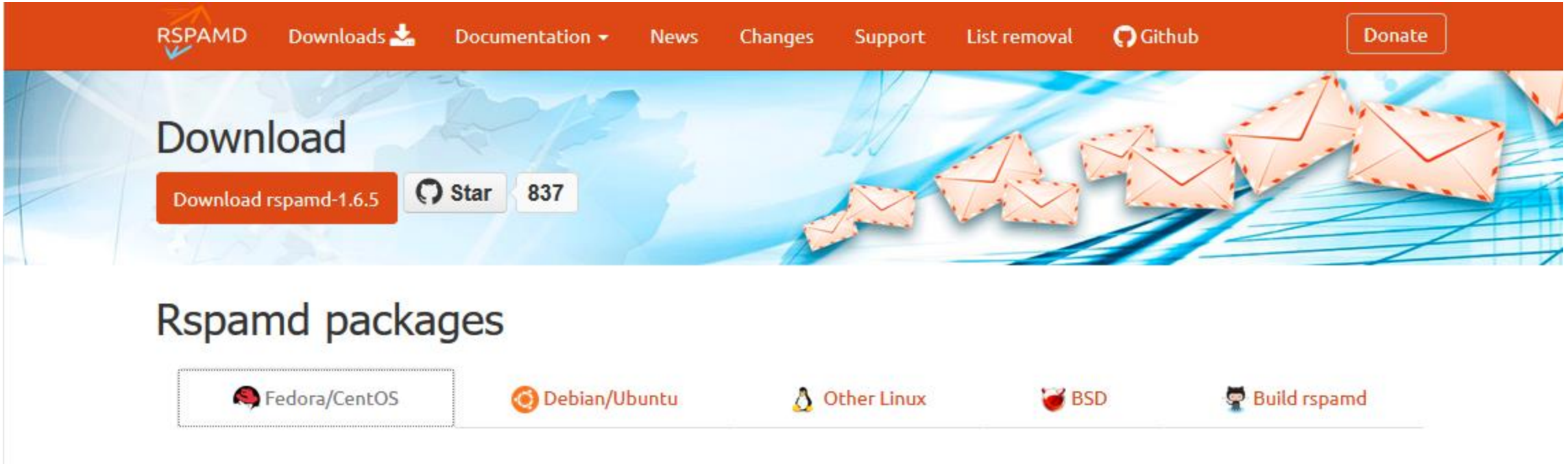
```
# vi /etc/postfix/main.cf

#--以下のを追加--
smtpd_milters = inet:localhost:11332
milter_default_action = accept
milter_protocol = 6
#-----------------
```

※Postfix 2.6以前のデフォルトMilter Protocolバージョンは2、その場合はmilter_protocol = 2を設定

インストール

- Rspamdにて公式パッケージがあります
- OSによってはOSオフィシャルのパッケージがあります



インストール

- CentOS6, CentOS7の場合 

```
# yum install epel-release
# curl https://rspamd.com/rpm-stable/${YOUR_DISTRO}/rspamd.repo >
/etc/yum.repos.d/rspamd.repo
# rpm --import https://rspamd.com/rpm-stable/gpg.key
# yum update
# yum install rspamd
```

※CentOS7の場合 ${YOUR_DISTRO}=centos-7

インストール

- Debian/Ubuntuの場合  

```
# apt-get install -y lsb-release wget # optional
# CODENAME=`lsb_release -c -s`
# wget -O- https://rspamd.com/apt-stable/gpg.key | apt-key add -
# echo "deb [arch=amd64] http://rspamd.com/apt-stable/ $CODENAME main" >
/etc/apt/sources.list.d/rspamd.list
# echo "deb-src [arch=amd64] http://rspamd.com/apt-stable/ $CODENAME main" >>
/etc/apt/sources.list.d/rspamd.list
# apt-get update
# apt-get --no-install-recommends install rspamd
```

※Debian official reposのRspamdはメンテナンスされていないらしいので
RspamdではRspamd公式のパッケージを使用することを勧めています

インストール

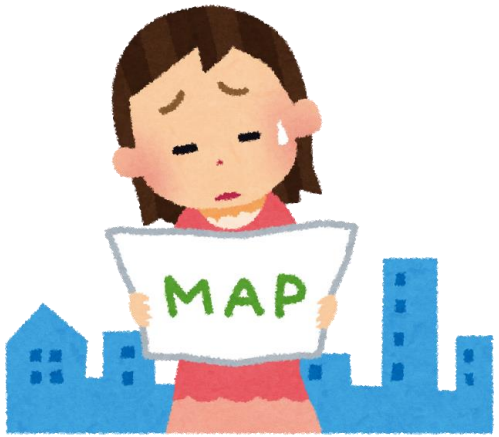
- その他Linux、FreeBSD  

RspamdのDownloadサイトを参考に

https://rspamd.com/downloads.html

初期設定する前に

- 独特？の作法があるので設定は最初戸惑います。



インストール直後の/etc/rspamd

```
$ ls /etc/rspamd

2tld.inc                groups.conf    mime_types.inc   rspamd.conf              worker-controller.inc
actions.conf            local.d/       modules.conf     scores.d/                worker-fuzzy.inc
cgp.inc                 logging.inc    modules.d/       spf_dkim_whitelist.inc   worker-normal.inc
common.conf             maillist.inc   options.inc      statistic.conf           worker-proxy.inc
composites.conf         metrics.conf   override.d/      surbl-whitelist.inc
dmarc_whitelist.inc     mid.inc        redirectors.inc
```

結論から

- /etc/rpsamd以下の○○.confや○○.incは直接編集は 
- 同様にmodules.d/およびscores.d/以下の直接編集は 
- ○○.confや○○.incのパラメータを変えたい場合は、
  local.d/ 以下に同名の○○.confや○○.incを作って編集 
- /etc/rspamd/rspamd.confを変更する場合は、
  /etc/rspamd/rspamd.conf.localを作って編集 

例えば

/etc/rspamd/modules.d/rbl.conf の設定値を変えたい場合

↓

/etc/rspamd/local.d/rbl.confというファイルを作成して差分を記述

設定変更の流れ

1. 設定を変えたい○○.incや○○.confを確認
2. 同じ名前の○○.incや○○.confを local.d/に作成する
3. 作成した○○.incや○○.confに変更したいパラメータや追加したいパラメータを記述する
4. rspamdadm configtest で Syntax checkをする
5. rspamdadm configdump で 変更結果を確認
6. service rspamd reload でリロードして設定を反映

local.dとoverride.dの違い

- local.d/はデフォルト設定と設定をマージされる
- override.d/はデフォルト設定を破棄して上書きで置き換わる

設定ファイル適用の優先度

1. デフォルトのconfをロード
2. /etc/rpsamd/local.d/ 以下の設定をマージ
3. /var/lib/rspamd/dynamic/（WebUI経由での変更?）以下の設定で上書き
4. /etc/rpsamd/override.d/ 以下の設定で上書き

の順番でconfigが反映される

modules.d/dmarc.confの内容

```
dmarc {
    .include(try=true,priority=5) "${DBDIR}/dynamic/dmarc.conf"
    .include(try=true,priority=1,duplicate=merge) "$LOCAL_CONFDIR/local.d/dmarc.conf"
    .include(try=true,priority=10) "$LOCAL_CONFDIR/override.d/dmarc.conf"
}
```

マッチしたルールのスコア値の変更

- ルールのスコア値の一部は、scores.d/以下の記述されている
- スコア値を変える場合もlocal.d/以下に○○_group.confを作成して、変更したいパラメータを記述する

※/usr/share/rspamd/rules/にもルールとそのスコア値の一部を見つけることができる

初期設定

1. rspamadm configwizardでRedisの設定
2. action設定を行い、rejectやgreylistの閾値を設定
3. メールヘッダーに付与する情報の設定
4. bayesの初期学習

初期設定

- rspamadm configwizardを実行

　RedisやDKIM署名、WebUIのパスワード設定を対話式で行うことがきます

rspamadm configwizardを実行

#rspamadm configwizard

Welcome to the configuration tool
We use /etc/rspamd/rspamd.conf configuration file, writing results to /etc/rspamd
Modules enabled: fuzzy_check, hfilter, phishing, dkim_signing, asn, settings, regexp, arc, trie, bayes_expiry, elastic, rspamd_update, ip_score, metadata_exporter, rbl, chartable, mid, multimap, dkim, surbl, mime_types, maillist, emails, once_received, dmarc, forged_recipients, milter_headers, whitelist, force_actions, spf
Modules disabled (explicitly): spamtrap, url_tags, mx_check, url_reputation
Modules disabled (unconfigured): spamassassin, reputation, metric_exporter, dynamic_conf, antivirus, fuzzy_collect, dcc, maps_stats, clickhouse
Modules disabled (no Redis): greylist, url_redirector, replies, neural, ratelimit, history_redis
Modules disabled (experimental):
Modules disabled (failed):
Do you wish to continue?[Y/n]: **(Enterキー)**

各機能の有効無効を示す

rspamadm configwizardを実行

Setup WebUI and controller worker:
Controller password is not set, do you want to set one?[Y/n]: **N**
Redis servers are not set:
The following modules will be enabled if you add Redis servers:
* greylist
* url_redirector
* replies
* neural
* ratelimit
* history_redis
Do you wish to set Redis servers?[Y/n]: **(Enterキー)**
Input read only servers separated by `,` [default: localhost]: **(Enterキー)**
Input write only servers separated by `,` [default: localhost]: **(Enterキー)**
Do you have any password set for your Redis?[y/N]: **(Enterキー)**
Do you have any specific database for your Redis?[y/N]: **(Enterキー)**

rspamadm configwizardを実行

Do you want to setup dkim signing feature?[y/N]: **(Enterキー)**
You have 1 sqlite classifiers
Expire time for new tokens [100d]: **(Enterキー)**
Reset previous data?[y/N]: **(Enterキー)**
Do you wish to convert them to Redis?[Y/n]: **(Enterキー)**
Convert spam tokens

Convert ham tokens

Migrated 0 tokens for 2 users for symbols (BAYES_SPAM, BAYES_HAM)
Converted classifier to the from sqlite to redis
File: /etc/rspamd/local.d/classifier-bayes.conf, changes list:
backend => redis
new_schema => true
expire => 8640000

rspamadm configwizardを実行

```
File: /etc/rspamd/local.d/redis.conf, changes list:
write_servers => localhost
read_servers => localhost

Apply changes?[Y/n]: (Enterキー)
Create file /etc/rspamd/local.d/classifier-bayes.conf
Create file /etc/rspamd/local.d/redis.conf
2 changes applied, the wizard is finished now
*** Please reload the Rspamd configuration ***
```

actionの設定

- local.d/actions.confを作成し、actionの閾値を変更

```
# vi /etc/rspamd/local.d/actions.conf

#--ファイルに追加--
reject = 16;
add_header = 5;
greylist = 4;
#--------------------
```

※閾値は適宜変更、特定のactionを無効にしたい場合は、reject = null;のようにnullを設定

https://rspamd.com/doc/tutorials/writing_rules.html

メールヘッダーに付与する情報の設定

- local.d/milter_headers.confを作成

```
# vi /etc/rspamd/local.d/actions.conf

#--ファイルに追加--
extended_spam_headers = true;
#または
#use = ["x-spamd-result","x-rspamd-server","x-rspamd-queue-id"];
#--------------------
```

X-Rspamd-Queue-Id: E3FEFC1483
X-Spamd-Result: default: False [15.31 / 6.00];
FORWARDED(0.00)[xxxx@example.com];
R_SPF_SOFTFAIL(0.00)[~all];
SPF_FAIL_FORWARDING(0.00)[];
FORWARDED(0.00)[xxxx@example.com];
:
NEURAL_SPAM_SHORT(2.00)[1.000,0];
RCVD_NO_TLS_LAST(0.00)[]
X-Rspamd-Server: rspamd01.example.net
X-Spam: Yes

extended_spam_headers=true
で追加されるメールヘッダ

actons.confのadd_headerの閾値
を超えると付与される

BAYESの初回学習（手動学習）

- rspamc learn_spam ディレクトリ or ファイル

```
# rspamc learn_spam ~/spam/
```

- rspamc learn_ham ディレクトリ or ファイル

```
# rspamc learn_ham ~/ham/
```

※デフォルト設定では200件以上学習させる必要があります

- rspamc statで学習した件数を確認

```
# rspamc stat
 :
Statfile: BAYES_SPAM type: redis; length: 0; free blocks: 0; total blocks: 0; free: 0.00%; learned: 1300; users: 1; languages: 0
Statfile: BAYES_HAM type: redis; length: 0; free blocks: 0; total blocks: 0; free: 0.00%; learned: 165; users: 1; languages: 0
Total learns: 1465
```

BAYESで学習した件数

WebUIの設定

- WebUIはworker-controllerが担当
- worker-controllerはlocalhostでLISTENしているのを変更してネットワーク経由でアクセスできるIPでLISTENする必要ある
- ログインのためのパスワードを設定する必要

worker-controllerの設定

- ログイン用パスワードハッシュを生成

```
# rspamadm pw

Enter passphrase: (パスワード入力)
$2$w94uctgb8yusmhp7kpy59rz48wp8q5p4$si67egqcx95cnj31jzd8eptkzekdh4fkannng1fuz3fbu36nia6rb
```

パスワードハッシュ

worker-controllerの設定

- worker-controller.incを編集

```
# vi /etc/rspamd/local.d/worker-controller.inc

#--ファイルに追加--
password = “(rspamadm pwで生成したパスワードハッシュ)";
enable_password = “(rspamadm pwで生成したパスワードハッシュ)”;
bind_socket = "*:11334";
#--------------------
```

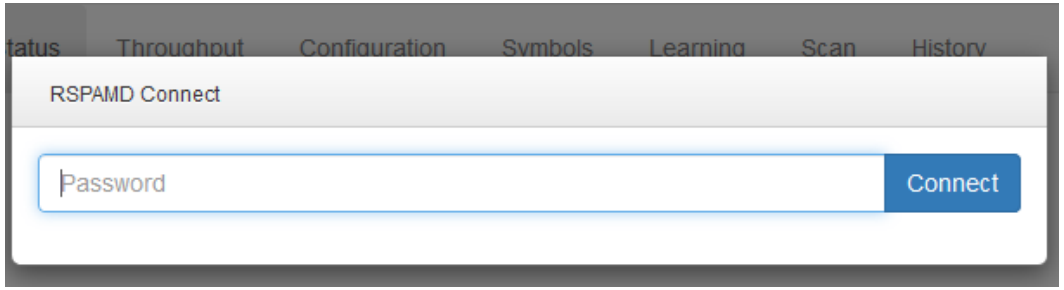
※passwordは閲覧のみ権限でのログイン用

※enable_passwordはルール等の編集可能な権限でのログイン用

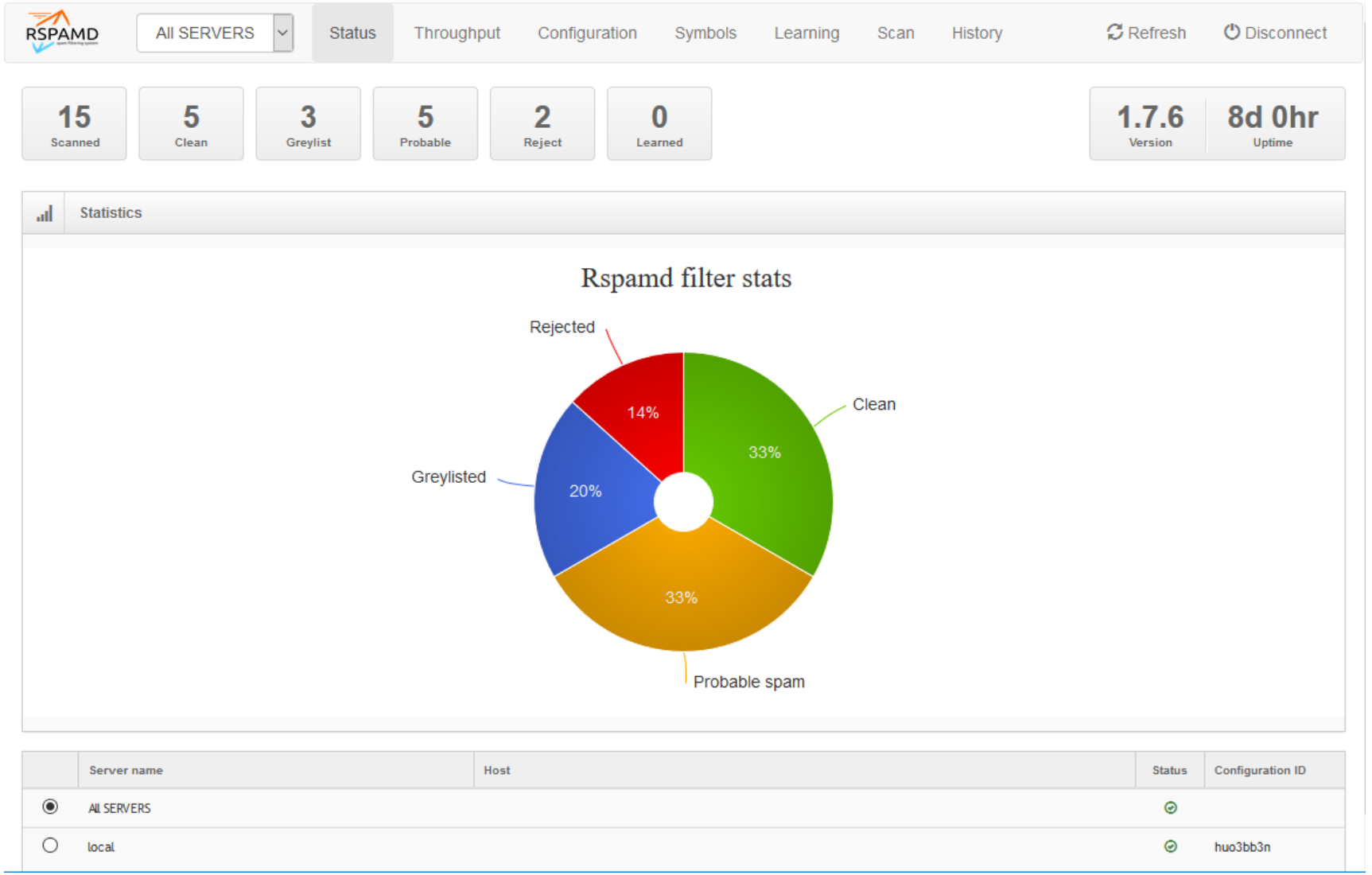
※別途、iptablesでアクセス出来るネットワークを限定

ログイン

- WebUIには http://IPアドレス:11334/ でアクセスします
パスワードを聞かれるのでパスワードを入力します。



ログイン後のトップページ



各種機能の紹介

- Rspamdは各種モジュールがあり、それによって複数の機能を有効または無効化しながら、必要な機能を利用することができる。
- 各種モジュールを一部紹介

※各種モジュールのドキュメント

https://rspamd.com/doc/modules/index.html

標準モジュール

- Cベースのモジュール
  コア機能、高速化を主眼としたモジュール
- Luaベースのモジュール
  コンパクトなコードを目的としたモジュール

Cモジュール

- chartable
- DKIM
- fuzzy_check
- SPF
- surbl
- regexp

chartable

- メールのメッセージに不自然に異なる言語の文字が混ざっていないかをチェックする
- どのくらい異なる文字（言語?）が含まれているかをカウント
- テストした範囲では、

  件名「【全品ポイント3倍】クリアランスSALE開催中！」

  件名「【先着５０名限定！】毎月１００万円をあなたにプレゼント！」

  のメールがスコアリング（弱めに）されていた

DKIM

- メールのDKIM署名を検証
- DKIM署名の追加
- DKIM署名検証をスルーできるホワイトリストの設定
- DKIM違反により厳しいスコアをつけるドメイン一覧を設定

fuzzy_check

- fuzzyストレージに格納されている特定のスパムメール（or ハム）ファジーハッシュとメッセージを比較して類似性があるかをチェック
- 添付ファイルと画像は現在ファジーハッシュと照合されないが、

blake2ダイジェストと完全一致するかチェックする

- デフォルトのfuzzy ストレージサーバは、fuzzy.rspamd.comを参照している
- ローカルにfuzzy ストレージサーバを立てることも可能

regexp

- 正規表現を使ってメッセージをフィルタリング
- メールヘッダー、MIMEヘッダー、MIMEパート、RAWメッセージ、URLと様々な部分をフィルタすることができる
- regexpを使って標準でどのようなフィルタをしているかは、
  /usr/share/rspamd/rules/regexp 以下を参考
- 独自のフィルタを書くことも可能
- フィルタの書き方は、
  https://rspamd.com/doc/modules/regexp.html
  https://rspamd.com/doc/tutorials/writing_rules.html
  を参考ください

SPF

- SPFレコードにより送信元メールサーバを検証

surbl

- メールのメッセージに含まれるURLが URL版RBLサービスのブラックリストに登録されているかを確認する
- デフォルトで参照に行っているURL RBL

✓multi.surbl.org
✓multi.uribl.com
✓dbl.spamhaus.org
✓sbl.spamhaus.org
✓uribl.spameatingmonkey.net
✓fresh15.spameatingmonkey.net
✓public.sarbl.org
✓uribl.rspamd.com

- サブドメインまでチェックするドメイン一覧の設定ができる
  ファイルは /etc/rspamd/2tld.inc

Luaモジュール

- コアモジュール以外のモジュール、新しい機能は大抵Luaモジュールで作られる
- 未設定だと無効なモジュールもあります

  （antivirus, spamassassin 等）
- 試験的な機能などにより、デフォルトで明示的に無効化されている機能もあります

  （mx_check,url_tag, url_reputation, spamtrap）

Luaモジュール（デフォルト有効）

- antivirus
- arc
- asn
- clickhouse
- bayes_expiry
- dcc
- dkim_signing
- dmarc
- elastic
- emails
- force_actions
- forged_recipients
- greylisting
- history redis
- hfilter
- ip_score
- maillist
- metadata_exporter
- metric_exporter
- mid
- milter_headers
- mime_types
- multimap
- neural networks
- once_received
- phishing
- ratelimit
- replies
- rbl
- rspamd_update
- spamassassin
- trie
- whitelist
- url_redirector

Luaモジュール（デフォルト無効）

- mx_check
- spamtrap
- url_reputation
- url_tags

antivirus

- メールを指定したアンチウイルスソフトでスキャンする
- ClamAV、F-Prot、Sophos、Avira に対応
- whitelistで特定のIPからのメールを除外設定可能
- ウイルス検知したメールをrejectさせる設定も可能
- 画像以外の添付付きのメールのみチェックも可能

ARC(Authenticated Received Chain)

- 中継したメールサーバがつけるReceivedヘッダに対するARC署名とシールチェックする
- ARC署名をする
- whitelistによる署名チェック除外ドメイン設定

ASN

- 送信元メールサーバのIPアドレスから、AS番号やサブネット、国コードを検索、それらの情報を他のプラグインに提供する。
- 検索先はans.rspamd.comで、RBLの様にDNSで検索する
- 前は検索にcymruを使用していたらしいが不安定を理由に独自に検索用サーバを立てたらしい

[Feature] Stop using cymru zone as it is unstable

https://github.com/vstakhov/rspamd/commit/c1c880f360600fc01b4c8ff449a3f4e6afaee19e#diff-72f19a68d4a5f04e85c8623791f7f870

Bayes expiry

- Redisに格納されているBAYESのトークンの有効期限を設定
- rspamadm configwizardでRedisを使用するように設定を進めると、下記の設定でlocal.d/classifier-bayes.confが生成されている

```
#------------------
backend = "redis";
new_schema = true;
expire = 8640000;
#------------------
```

- autolearn = true; を設定するとBAYESを自動学習するようになる

clickhouse

- Clickhouseという列指向データベースを使用する場合に使用
- 特定情報の参照性能の向上が期待できる
- IPv4, IPv6, URL, From:ヘッダー, ASN等
- デフォルトでは機能していない

DCC(Distributed Checksum Clearinghouses)

- DCCというNW協調型の迷惑メールデータベース
- DCCから迷惑メールのチェックサム（シグネチャ）を取得して、チェックサムに一致したかカウントして一定数になったらスパムと認定する
- デフォルトでは機能していない

DKIM signing

- 送信メールをスキャンする用途として使用する時に、送り出すメールにDKIM署名行う
- DKIMモジュールよりもより柔軟に署名をできるようになる

DMARC

- 送信者のドメインで公開されているDMARCのポリシーをチェックする
- DMARCのポリシーに従って、メールをrejectするなどアクションを強制する動作をさせることも可能

Elasticsearch

- 様々なメッセージ関連のメタデータをElasticsearchにプッシュする
- Elasticsearch 6.x以上
- ingest-geoipプラグイン
- kibana(オプション)
- デフォルトでは機能していない

Emails

- メールアドレスをハッシュ化した情報を元にRBLに問い合わせる、または静的リストに基づいてチェックする
- デフォルトで参照先するRBLとして
  - ✓email.rspamd.com
  - ✓ebl.msbl.org

  を参照するようになっている
- 静的リストでは、

  /^[^@]+@example.com$/I などの正規表現もサポート

Force Actions

- 特定のルール（Symbol）に該当または該当しない場合に強制的にアクション（何もしない、Reject、Rewrite）を適用
- オプションで指定されたSMTPメッセージを返すこともできる
- デフォルトでは特に設定はありません

Forged recipients

- 宛先や送信元が偽造の疑いがないかを確認
- From:ヘッダとMAIL FROM:のメールアドレス不一致を確認
- 宛先が RCPT TO:のメールアドレスと不一致を確認

※ドキュメントがないので詳しくは /usr/share/rspamd/lua/forged_recipients.luaを参照

Fuzzy collect

- 自身でFuzzyストレージを構築する際に必要
- 分離されたSpamtrapからファジーハッシュを収集し、レプリケーションプロトコルを使用してローカルのファジーストレージに書き込む役割を担当。
- デフォルトでは無効

Greylisting

- 閾値を超えるスコアを持つメールを一時受け取り拒否する
- 一時拒否時間（デフォルト5分）
- Greylistに登録される情報はRedisに保存
- ホワイトリストを設定可能
  (local.d/greylist-whitelist-domains.incにドメイン一覧を記述)

History Redis

- WebUIで参照できるhistoryデータの保存についての設定
- 最大保存履歴数、件名のハッシュ化によるプライバシー対策

```
# vi local.d/history_redis.conf

#--------------------
nrows = 2000; #履歴を2000に拡張
subject_privacy = true; #件名をプライバシー対応（ハッシュ化）
#--------------------
```

hfilter

- 接続元メールサーバのIPアドレスやホスト名が問題ないかをチェック（問題: ホスト名が動的IPに付くようなホスト名）
- メールのFROMアドレスのドメインが存在するかAレコードやMXレコードがあるかをチェック
- 等々

ドキュメントがないので詳しくは /usr/share/rspamd/lua/hfilter.lua を参照

```
local checks_hellohost = [[
/[0-9][.-]?nat/i 5
/homeuser[.-][0-9]/i 5
/[0-9][.-]?unused-addr/i 3
/[0-9][.-]?pppoe/i 5
/[0-9][.-]?dynamic/i 5
/[.-]catv[.-]/i 5
/unused-addr[.-][0-9]/i 3
/comcast[.-][0-9]/i 5
/[.-]broadband[.-]/i 5
/[0-9][.-]?fbx/i 4
/[.-]peer[.-]/i 1
/[.-]homeuser[.-]/i 5
/[0-9][.-]?catv/i 5
/customers?[.-][0-9]/i 1
/[.-]wifi[.-]/i 5
/[0-9][.-]?kabel/i 3
        :
```

IP Score

- IP ScoreはASNモジュールでの情報を元にスコアリングする
- 特定のIP/Subnet/AS番号/国から受信したメッセージの数を追跡して、合計スコアともに記録する
- 迷惑メールが多く届く同じIPであればスコアが上がり、正常なメールが多く届くIPであればスコアが下がる

Mailing list

- 一般的なメーリングリストのソフトウェアでメッセージが送信されてきたかどうかをチェック
- 一部のメーリングリストからメールが来た時、不必要にルールに触れないように、いくつかのルールを無効にするように動作
- 現在サポートされているメーリングリスト一覧
  - ✓Ezmlm
  - ✓Mailman
  - ✓Google groups
  - ✓Majordomo
  - ✓Communigate PRO mailing lists
  - ✓subscribe.ru mailing list

Metadata exporter

- メタデータを外部サービスにプッシュすることができる
- HTTP(POST), Redis Pub/Sub, SMTP経由
- Generalメタデータは以下の通り

action, from, header_date, header_from, header_subject, header_to, ip, mail_from (email_template only), mail_to (email_template only), message_id, our_message_id (email_template only), qid, rcpt, score, symbols, user

- デフォルトではどこかへexportする設定はされていない

Metric exporter

- Rspamdから統計情報を収集して、外部グラフ化ツールのGraphiteにエクスポートする
- エクスポートする統計情報(rspamc statで取れるような情報)

"actions.add header", "actions.greylist", "actions.no action", "actions.reject", "actions.rewrite subject", "actions.soft reject", "bytes_allocated", "chunks_allocated", "chunks_freed", "chunks_oversized", "connections", "control_connections", "ham_count", "learned", "pools_allocated", "pools_freed", "scanned", "shared_chunks_allocated", "spam_count“

- デフォルトでは機能していない

MID

- 特定のドメインでDKIM署名がなされたメッセージでINVALID_MSGID（不正なメッセージID）やMISSING_MID（メッセージIDがない）となるメッセージのスコア上昇を抑制
- /etc/rspamd/mid.inc に特定のドメインを記述されている

alibaba.com /^[a-f0-9]{8}(?:-[a-f0-9]{4}){3}-[a-f0-9]{12}-0$/
aliexpress.com /^(?:(?:[0-9]{14,15}|[a-z]{4}UTT_[0-9]{5,6}_(?:[0-9]{8}_)?)[$])?[a-f0-9]{32}$/
beeline.ru /^<[A-z0-9+]{18}>$/
noreply.esphere.ru
noreply.etprf.ru
rkn.gov.ru
garant.ru
is-zakupki.com
mirtesen.ru
fcod.nalog.ru
otc.ru
qiwi.ru
client.rostelecom.ru
sberbank-ast.ru
crm.taxcom.ru
wildberries.ru /^[a-f0-9]{8}(?:-[a-f0-9]{4}){3}-[a-f0-9]{12}$/
promo.wildberries.ru /^[A-F0-9]{8}(?:-[A-F0-9]{4}){3}-[A-F0-9]{12}$/
taxi.yandex.ru /^[a-f0-9]{32}$/
online.zcts.ru

Milter headers

- メールヘッダーに追加する情報または、削除する情報の設定
- SAのようなヘッダー情報をつけたり

Mime types

- MIMEタイプの健全性をチェック
- zipやrarで圧縮された添付ファイルもチェック可能
- 添付ファイルに特定の拡張子を見つけたらスコアを上げる
- デフォルトでどの拡張子がどのくらいのスコアをつけるかは、/usr/share/rspamd/lua/mime_types.lua を参照

Multimap

- ドメイン一覧、IPアドレス一覧などの各種動的リストを作ることができる
- 対象ファイルのmtimeを監視していて、更新があったらファイルを再ロードする
- メールアドレスからドメイン部を抽出して活用などのフィルタ機能もサポート
- ホワイトリスト、ブラックリスト、その他リストをファイルで整理するのに便利

Multimapデフォルト設定抜粋

```
multimap {
    # Freemail Addresses
    freemail_envfrom {
        type = "from";
        filter = "email:domain";
        map = "https://maps.rspamd.com/freemail/free.txt.zst";
        symbol = "FREEMAIL_ENVFROM";
        description = "Envelope From is a Freemail address";
        score = 0.0;
    }
:
```

Neural network

- Lua Torchというディープラーニングライブラリを使用して、迷惑メールと正常なメールを学習させ、迷惑メールか、否かを判定しスコアリングに利用する
- 複数の異なるニューラルネットワークを作ることができる

- Lua Torch自体を調べると理解が進みそうだけど・・・

```
rules {
  "LONG" {
    train {
      max_trains = 5000;
      max_usages = 200;
      max_iterations = 25;
      learning_rate = 0.01,
      spam_score = 8;
      ham_score = -2;
    }
    symbol_spam = "NEURAL_SPAM_LONG";
    symbol_ham = "NEURAL_HAM_LONG";
    ann_expire = 100d;
  }
  "SHORT" {
    train {
      max_trains = 100;
      max_usages = 2;
      max_iterations = 25;
      learning_rate = 0.01,
      spam_score = 8;
      ham_score = -2;
    }
    symbol_spam = "NEURAL_SPAM_SHORT";
    symbol_ham = "NEURAL_HAM_SHORT";
    ann_expire = 1d;
  }
}
```

Once received

- Receivedヘッダーに含まれるホスト名が正常なメールサーバのホスト名によく使われるパターンのホスト名(mail.example.com, mx1.example.com)か、悪いパターンのホスト名（dynamic-xxx.xxx～. static-xxx.xxx～)かで、スコアリングする
- パターンには Lua patternも使える
- hfilterで漏れた動的IPをスコアリングするときに使う？

Phishing

- フィッシングの可能性のあるURLがないかチェック
- フィッシングとみなすURL

http://**example.co.uk**

- フィッシングとみなさないURL

http://**example.com**/other

- OpenphishiおよびphishtankのフィードからフィッシングサイトへのURLリンクかを調査

openphish: https://openphish.com/

phishtank: https://www.phishtank.com/

Ratelimit

- 特定の送信者、特定のIPアドレス、特定の受信者を条件にレートリミットをかける（一時的拒否）
- デフォルトではレートリミットは設定されていません

RBL

- RecivedヘッダーのサーバIPアドレスや SMTP接続してきたサーバのIPアドレスで各種RBLに対して照合を行う
- どのRBLに対して、どの情報（Recivedヘッダー、FROMヘッダー、送信元SMTPサーバの逆引き等）で参照するか設定できる
- デフォルトで参照しているRBL
  - ✓zen.spamhaus.org
  - ✓rep.mailspike.net
  - ✓bl.score.senderscore.com
  - ✓spam.abuse.ch
  - ✓bl.spameatingmonkey.net
  - ✓bl.ipv6.spameatingmonkey.net
  - ✓list.dnswl.org

Replies

- 正常なメールへの返信メールを迷惑メールに誤判定されにくくする
- 認証されたユーザーから送信されたメッセージのmessage-idヘッダーを収集して対応するハッシュをRedisに保存、さらに、受信したのメッセージの「In-Reply-To:」ヘッダーをハッシュし、得点を調整したり、設定に従ってアクションを強制したりする。

Rspamd update

- Rspamd自体を更新せずに新しいルールとスコアのバックポートを提供
- SAにおける、sa-update?

Spamassassin rules

- SAルールセットを読み込んで利用できる
- SAユーザーがルールの再利用をしたい場合に便利
- ただしRspamのパフォーマンスを低下する
- 一部プラグインをサポートしていない
- 設定は有効になっているが、何も設定はされていない

trie matcher

- aho-corasicアルゴリズムを使用して高速にrawメッセージやテキストパーツ内の複数の文字列を検索するときに使用
- 単語境界は区別しないので、testという検索条件で、test, tests, 123testingもヒットする
- デフォルトでは特定の文字列を検査する設定は入っていません

※aho-corasic 参考

http://d.hatena.ne.jp/naoya/20090405/aho_corasick

http://tobya.hatenablog.com/entry/2017/12/20/223629

URL redirector

- URLリンクをSURBLモジュールがチェックしようとするときに転送URLが使われている時に、その先の真のURLをチェックさせることができる
- リダイレクト先を見に行く必要のあるドメインは、ファイルに転送URLサービスのドメイン一覧を記述する
- デフォルトでは redirectors.incというファイルに約1000件ほど転送URLのドメインが登録されている

Whitelist

- 指定されたファイル内のドメイン一覧をホワイトリストとして参照して、SPF, DKIM, SPFとDKIM両方,DMARCいずれかによる検証に通ったメールに対して、マイナススコアをつける
- ブラックリストを作ることも可能
- デフォルトで/etc/rspamd/以下にdmarc_whitelist.incと

spf_dkim_whitelist.inc が用意されていて、いくつかのドメインが登録されている

まとめ

- 各種参照しているRspamd頼みのリストは日本用に追加のリストを作ったほうがよさそう
- configファイルのお作法ははじめは戸惑うが、ドキュメントのあり、比較的難しくないのでは
- Fuzzyストレージを自前で構築してみてRspamd提供のFuzzyストレージとの結果の違いを見てみたい

その他参考

- Rspamd Documentation
  https://rspamd.com/doc/
- Rspamd google groups
  https://groups.google.com/forum/#!forum/rspamd
- github
  https://github.com/vstakhov/rspamd