

# Fromアドレスをホワイトリストで確認して結果をメールヘッダに含める

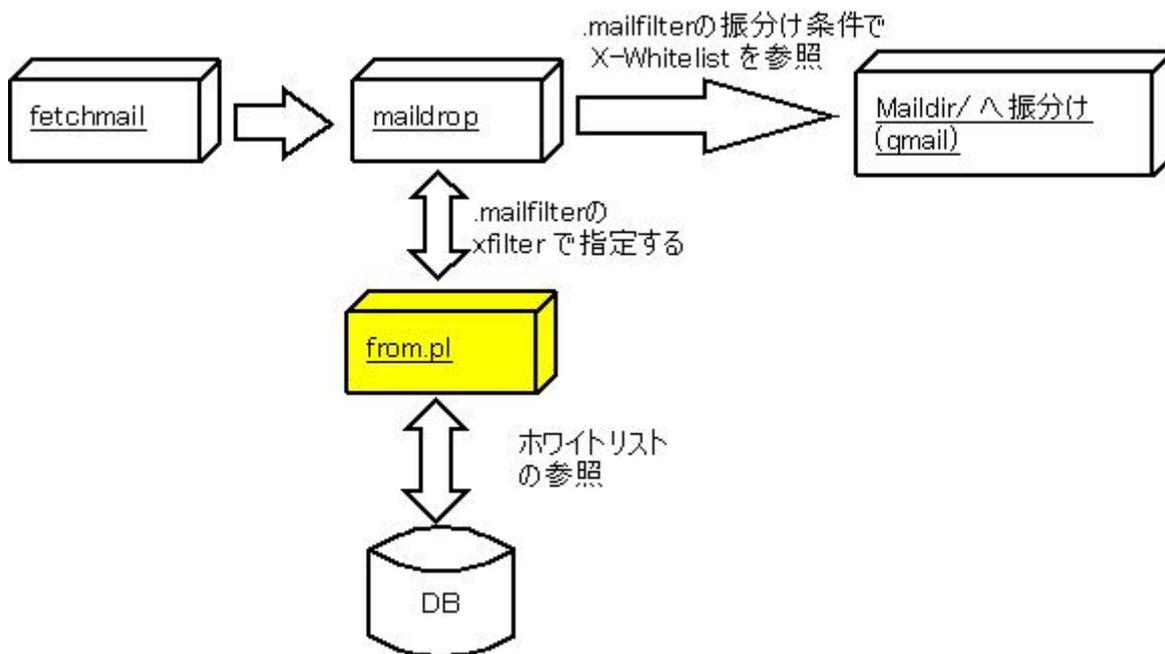
2009年07月15日 07時12分37秒

メールを読み込んでFromヘッダで示すメールアドレスが準備したホワイトリストに含まれているか調べる。調べた結果を X-Whitelist ヘッダで示す。

## 概要

maildrop用で使う事を想定している maildropのレシピ中で X-Whitelist に従った振り分けを実施すればOKなはず。

- ホワイトリストに存在する → X-Whitelist: Active
- ホワイトリストに存在しない → X-Whitelist: Non-Member
- ホワイトリストを読めなかった → X-Whitelist: Unknown



## ソース

from.pl

[from.pl](#)

```
#!/usr/bin/perl

use DBI;
use Encode;
use MIME::Parser;
```

```
my $dbhost="ホスト名 or IP";
my $dbname="データベース名";
my $dbuser="ログインユーザ";
my $dbpass="パスワード";
my $parseObj = MIME::Parser->new();
my $entity;
my $subject, $from, $flag;

### 標準入力からメールを読み込んで件名と送信者アドレス取得
    $parseObj->output_dir('./');
    $entity = $parseObj->parse(\*STDIN);
    $subject = encode("euc-jp", decode("MIME-Header",
$entity->head->get("Subject")) );
    $from = encode("euc-jp", decode("MIME-Header",
$entity->head->get("From")) );
    if ( $from =~ /^.+<([^\t]+@[^\t]+)>.*$/ ) {
        $from =~ s/^.+<([^\t]+@[^\t]+)>.*$/$1/;
    }
    else {
        $from =~ s/^(([^\t]+@[^\t]+)[\t].*)*/$1/;
    }
    $from =~ s/\r$//;
    $from =~ s/\n$//;

### ホワイトリストの確認(この例では□□を見に行ってる
    eval {
        $dbh =
DBI->connect("dbi:Pg:dbname=$dbname;host=$dbhost",$dbuser,$dbpass) or
die "cannot connect database.";
        $sth=$dbh->prepare( "select case exists(select * from
\"EmailWhiteList\" where email = ? ) when true then 'Active' ELSE 'Non-
Member' END" );
        $sth->bind_param(1,$from,{TYPE => SQL_VARCHAR});
        $sth->execute();
        $sth->bind_col(1, \$flag);
        $sth->fetch();
        $sth->finish();
        $dbh->disconnect();
    };
    $flag = "Unknown" if $@;

### ヘッダ X-Whitelist を付与して標準出力へ出力
    $entity->head->add("X-Whitelist",$flag);
    ###print $entity->print();
    $entity->print(\*STDOUT);
    $entity->purge();
```

## 説明

### メールの解析(パース)

MIME-Toolsの PerseParserを使っている。これでメールをパースし Entityオブジェクトにする。パース結果にヘッダ付与後、そのまま標準出力へ出す。

パースの際、

```
decode("MIME-Header", $entity->head->get("Subject"))
```

の部分は

```
$entity->head->decode->get("Subject")
```

でも良さ気だが、これを使うと\$entity内に格納されたMIMEエンコード部分(例えばFromヘッダの漢字部分)が全てデコードされてしまう。ヘッダの付与以外は書き換えをしたくないのでこれは困る。そのため Encode::decode() を使っている `decode()` で指定のエンコーディングに "MIME-Header" を指定するとどうやら

```
=?ISO-2022-JP?B?GyRCJSI\ZCVRJSQ\USVzGyhC?=
```

な部分をよしなにしてくれるようだ。

### Fromヘッダからメールアドレス抽出

<>で囲まれた場合とそうでない場合で切り出し方を変えている。

```
if ( $from =~ /^.+<([\t]+@[^ \t]+)>.*$/ ) {  
    $from =~ s/^.+<([\t]+@[^ \t]+)>.*$/$1/;  
}  
else {  
    $from =~ s/^([\t]+@[^ \t]+)[ \t].*$/$1/;  
}
```

- hoge hoge <hoge@fogo.com>
- hoge@fogo.com (Daemon send)

のようなパターンに対処する。いずれも hoge@fogo.com が切り出しされる。

### ホワイトリスト照合

今回はDBIを使ってPostgreSQLデータベースにあるテーブル EmailWhiteList で検索を行い切り出したメールアドレスがこのテーブルに含まれていない場合には Non-Member `含まれていれば Active` を返す。多分CSVとかのフラットファイルでも取り扱いできると思う。

## その他

ホワイトリストに有効期限も格納すれば、ある期間だけ有効なメールアドレス制御が可能になる。また Subjectの記述パターンを検出する事で、件名に配信番号等が含まれていない場合の処理も記述できるようになる。

[mail](#), [Perl](#), [技術資料](#)

From:  
<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:  
<https://wiki.hgotoh.jp/documents/mail/mail-004>

Last update: **2023/04/14 02:32**

