OcrPlugin

How it works

This plugin checks for specific keywords in image/gif attachments, using `gocr` (an *optical character recognition* program).

This plugin can be used to detect spam that puts all the real spam content in an attached image. The mail itself only random text and random html, without any URL's or identifiable information.

Requirements

You will need `convert` (imagemagick) and `gocr` installed.

Installation

Save the two files below in your local configuration directory, adjusting the score in `Ocr.cf` as you like, and the wordlist (`my @words =`) in `Ocr.pm` according to the spam you are receiving. You might want to run `gocr`
by hand on the image attachments to look for words that are correctly recognized.

Remarks

- Note that this is my first SA plugin, so any feedback is welcome
- The words checked for are specific for some spam I received a lot of recently.
- `gocr` can take up quite a bit of resources, so be careful. But it is only executed for messages that contain gif attachments.

ToDo

- Words are hardcoded. Should be a configuration parameter instead.
- Instead of checking for specific words, it might be better to "check if the image contains a certain amount of text", since it is not very likely that people send legitimate mail with text in images.

– Author: Maarten de Boer, mdeboer ~~at~~ iua ~~dot~~ upf ~~dot~~ edu

Changelog

Version 2:

- Use `convert` instead of `giftopnm`, because I received some mails with .gif's that were actually .jpg's. `convert` handles that ok.
- Some words added

Code

Ocr.cf

```
loadplugin Ocr Ocr.pm
body OCR eval:check_ocr()
describe OCR Check if text in attached images contains spam words
score OCR 3.0
```

Ocr.pm

```
# Ocr plugin, version 2
package Ocr;

use strict;
use Mail::SpamAssassin;
use Mail::SpamAssassin::Util;
use Mail::SpamAssassin::Plugin;

our @ISA = qw (Mail::SpamAssassin::Plugin);

# constructor: register the eval rule
sub new {
   my ( $class, $mailsa ) = @_;
   $class = ref($class) || $class;
   my $self = $class->SUPER::new($mailsa);
   bless( $self, $class );
   $self->register_eval_rule("check_ocr");
   return $self;
}

sub check_ocr {
   my ( $self, $pms ) = @_;
   my $cnt = 0;
   foreach my $p ( $pms->{msg}->find_parts("image") ) {
      my ( $ctype, $boundary, $charset, $name ) =
        Mail::SpamAssassin::Util::parse_content_type(
         $p->get_header('content-type') );
      if ( $ctype eq "image/gif" ) {
         open OCR, "|/usr/bin/convert -flatten - pnm:-|/usr/bin/gocr -i - > /tmp/spamassassin.ocr.$$";
         foreach $p ( $p->decode() ) {
            print OCR $p;
         }
         close OCR;
         open OCR, "/tmp/spamassassin.ocr.$$";
         my @words =
           ( 'company', 'money', 'stock', 'million', 'thousand', 'buy', 'price', 'don\'t' );
         while (<OCR>) {
            my $w;
            foreach $w (@words) {
               if (m/$w/i) {
                  $cnt++;
               }
            }
         }
         unlink "/tmp/spamassassin.ocr.$$";
      }
   }
   return ( $cnt > 1 );
}

1;
```