PluginWritingTips

Plugin-Writing Tips

These tips may be useful to developers of SpamAssassin plugins.

Setting Output Headers and Custom Template Tags

If you want to add custom headers in the rewritten mail message, the `Mail::SpamAssassin::PerMsgStatus::set_tag()` API is the way to do it.

The way it works is, you add a line like this to the plugin configuration file, or the user's preferences file:

```
add_header all Test-Tag the tag says _MYTAG_
```

and in the plugin code, call set_tag() like so:

```
$permsgstatusobject->set_tag ("MYTAG", "hello world");
```

(obviously you want something interesting in there, like results of a scan etc.) The resulting output will look like:

```
X-Spam-Status: [usual stuff]
X-Spam-Test-Tag: the tag says hello world
```

You can also get the value of a tag using the `Mail::SpamAssassin::PerMsgStatus::get_tag()` API from inside a plugin, although note that some of the tags will not be filled in until after the scan completes, and may not be available.

In SpamAssassin 3.1.0 onwards, you can use C<set_tag> and provide a subroutine reference, like so:

```
my $data = "hello world";
$permsgstatusobject->set_tag ("MYTAG", sub {
      return $data;
    });
```

Writing plugins with dynamic score rules

Let's assume you are writing a plugin which for example searches for specific characteristics or occurrences in the mail, then you would probably want to make the score for a rule using this plugin dynamic, so more characteristics or occurrences found by your plugin raise the score. There are plenty of other situations where you would want to use this, one example where it is being used already is the AutoWhitelist Plugin.

Here's how it works:

Assuming you have written a plugin Example.pm which has a subroutine check_example() to check a message, you create the rule like this:

```
loadplugin Example example.pm

header EXAMPLE_RULE_NAME eval:check_example()
describe EXAMPLE_RULE_NAME This is just an example rule
```

Note that we do not need a line for the score, this is enough.

The important parts happen in check_example() now:

First of all, make sure you

```
return 0;
```

at the end of the rule, no matter if your test succeeded or not (you will see later why).

Then, when your tests have finished and the score was calculated, you'll have to make the actual call that does the magic:

```
sub check_example() {
    my ( $self, $pms ) = @_;
    ...
    #You do all your tests and whatever else here
    ...
    #Now we assume that the score you calculated is in $score


    #optional: change the description for report and summary
    my $description = $pms->{conf}->{descriptions}->{EXAMPLE_RULE_NAME};
    $description .= " -- with score $score";
    $pms->{conf}->{descriptions}->{EXAMPLE_RULE_NAME} = $description;

    #If the score is 0, then you don't need to do anything (obviously)
    if($score) {
        #The magic call
        $pms->got_hit("EXAMPLE_RULE_NAME", "HEADER: ", score => $score);

        #Yet another magic call
        for my $set (0..3) {
            $pms->{conf}->{scoreset}->[$set]->{"EXAMPLE_RULE_NAME"} =
                                                sprintf("%0.3f", $score);
        }
    }

    return 0;
}
```

And thats it. The

```
got_hit ($rule, $area, %params)
```

call will trigger a hit for the specified rule with the specified score. Area means for example BODY or HEADER, so this has to be changed according to what you are doing. (See Mail::SpamAssassin::PerMsgStatus for other possible params.)

The

```
return 0;
```

is important because that will make sure that the actual rule that we defined earlier with the

```
eval:check_example()
```

never evaluates to true so it doesn't score. It doesn't need to score because the actual scoring is completely being done inside the subroutine. So even if you defined a score in the configuration file earlier, it would never be used.

The rule description is accessible with

```
$pms->{conf}->{descriptions}->{EXAMPLE_RULE_NAME}
```

. It is usually set by

```
describe EXAMPLE_RULE_NAME
```

in the rules file but can also be changed at runtime. (actually, changing this at runtime is not recommended and may not be supported in future versions; I'm not sure it even works in SA 3.2.0. if you want to do something similar, it would be better to open an enhancement request on our bugzilla so we can come up with a proper API. --jm)

Setting

```
$pms->{conf}->{scoreset}->[$set]->{"EXAMPLE_RULE_NAME"}
```

lets the dynamic score appear in the template tag *TESTSSCORES*, which might be used for the X-Spam-Status line. (The for loop is necessary to set all 4 values, for their explanation see Mail::SpamAssassin::Conf item `score SYMBOLIC_TEST_NAME n.nn`) (again, changing the {scoreset} values is not recommended and probably will not work in future versions! please talk about this via the bugzilla. --jm (see http://issues.apache.org/SpamAssassin/show_bug.cgi?id=5463))