Short Primer #01 on Using Regular Expressions (RegEx) with MailWasherPro Filters and Rules – page 1

It is difficult to keep up with the ever-changing strategies and tactics of spammers. However, I believe that MailWasherPro and the use of RegularExpressions in filters give us **intended victims** the ability to stay in the fight against spammers.

Over the years, I've observed the patterns that spammers use and have tried to emulate their creativity in defining effective ways to trap bad guys.

I'd like to share my experiences with other "intended victims" of spammers.

- Spam filtering at the **mail server level** has its limitations because:
  - Spam filtering is based on the methodologies, tools and techniques that an ISP provides its customers – customers have to live with and use what is offered
  - I feel that ISP spam rules are too generalized to keep up with spammer creativity
  - The challenge at the mail server is to adjust the span detection sensitivity-levels so that:
    - fewer bad guys get a false positive (low score) and are sent to a user's Inbox
    - fewer good guys get a high score and are sent to a user's Spam Box
  - When you set the sensitivity low enough to allow more good guys to pass through this filter, TOO MANY bad guys also score low enough so that they also pass through this filter and end up in a user's Inbox.
- Spam filtering at the **email client level** (Thunderbird and Outlook) is inefficient and by then it's too late. At this point, one may have a large volume of unwanted messages to deal with and manual deletion is a laborious and time-wasting activity.
  - The tricks spammers use to create infinite numbers of spellings and string-combinations make it difficult to create the kinds of filters needed to block bad guys.
- MailWasherPro operates on messages **still on the mail server** after the ISP filters have completed but **before messages are sent to the email client**. This provides more sophisticated filtering options.

Short Primer #01 on Using Regular Expressions (RegEx) with MailWasherPro Filters and Rules – page 2

Simple to complex – using a filter to trap a bad guy

Writing regular expressions may seem daunting. However, you can learn to do this one step at a time – i.e. learn about how to create a specific filter and then expand your learning as you combine what you learn next with what you learned before.

1. My first use of MWP filters was rather crude in that I created a filter with a large number of rules with text strings to trap bad guys:

- Examples:

Subject	contains	Plain text	viagra
Subject	Contains	Plain text	cialis
Subject	contains	Plain text	levitra
etc.			

2. Spammers created too many strings for these text rules to be practical, so my first **simple use** of RegularExpressions in rules was to concatenate these large numbers of rules text into smaller groups of rules using regular expressions to trap bad guys.

- Example:
  - A single rule with a RegEx expression replaces three separate rules

Subject	contains	RegEx	(viagra|cialis|levitra)
etc.			

  - The () parentheses indicate a group of “things” to look for
  - The | (pipe character) separates different “things” to look for

3. Then spammers got clever by **sometimes** (but not always) doing character replacements so that regular spellings of “things” would not trap bad guys:

- Examples:
  - Spammers substitute 0 for o; 1 or l for I; 3 for e, etc.
  - Some variants of Cialis: Viagra, v1agra, vlagra, etc.
  - Some variants of Cialis: cial1s, c1alis, c1al1s, etc.
  - Some variants of Levitra: lev1tra, l3vitra, 1ev1tra, etc.

Short Primer #01 on Using Regular Expressions (RegEx) with MailWasherPro Filters and Rules – page 3

4. My next more complex use of RegularExpressions was to counteract multiple ways that spammers alternated the spelling of bad guys:

- Examples:
    - The | (pipe character) separates different "things" to look for
    - The [] brackets indicate a series of "characters" to look for

Subject	contains	RegEx	v[i1l]agra
Subject	Contains	RegEx	c[i1l]a[i1l][i1l]s
Subject	contains	RegEx	[i1l][e3]v[i1l]tra

- This checks for i 1 or l in any combination
- This checks for 3 or e in any combination

5. The combination of these three strings into a single rule may seem complex, but it's just a concatenation of all three strings enclosed in parentheses and separated by pipes
    - This traps all three "things" no matter how they are spelled or misspelled

Subject	contains	RegEx	(v[i1l]agra|c[i1l]a[i1l][i1l]s|[i1l][e3]v[i1l]tra)

6. Compare rule (#5) (above) with this simpler rule (#6) (below) (same as #2, above) that checked only for the correct spelling of the "things"

Subject	Contains	RegEx	(viagra|cialis|levitra)

Short Primer #01 on Using Regular Expressions (RegEx) with MailWasherPro Filters and Rules – page 4

7. Some General Comments & Guidelines on Filters and Rules

- Remember, a filter can have many rules – of different types.
  - a. The desired outcome of a filter is to determine whether the filter is TRUE
  - b. You must make sure set up the filter to execute correctly based on its rules
    - i. Should a message have to match **any rule** in the filter? – i.e. a message that **matches any rule** in a filter makes the filter TRUE
    - ii. Should a message have to match **all rules** in the filter? – i.e. a message **MUST match all rules** in a filter to make the filter TRUE

- If creating strings for specific "things" with no alternate spellings, then a larger (and more readable) rule would be look something like this:

<table>
<tr><td>Subject</td><td>Contains</td><td>RegEx</td><td>(Hire Offshore Developer|Red pottery pot terracotta|TransUnion, Equifax, and Experian|Compare Medicare Plans|Research studies may offer payment|New Fat Burner|CVS by Storm|Penny Pot Stock|Roof is covered|Revolutionary Non-Stick|Scratch Resistant Pan|Medicare Enrollment Period|Election Sale|Become a Wall Street Journal Member|Dear in Christ|Chronic Constipation|Best-boost for you|your loving gun|Borrow from a trusted|crafty psychological trick|3 Things Jesus Said|About How to Cure Disease)</td></tr>
</table>

- I have found no real limit to the number of "things" that can be concatenated into a single rule – as long as the Regular Expression syntax is correct. I have also not found that there is a limit to the number of rules that can be added to a filter.

- **Suggestion:** Keep things simple – i.e. design rules within filters **in ways that make sense to you**. If it seems like a specific rule should be in a separate filter, then create a separate filter for it.

- I would recommend that you create a separate rule (or filter) for more complex situations such as the rule (#5) above. **Reason – It will be easier to debug.**

  Example: If it makes sense to you to create filters for "Subject", "From" and "Body" separately, then create separate filters for them.

- If you create a rule to check the "Header", everything in a message header will be checked by that rule. If you create a rule to check the "From", only the "From" address in a message header will be checked by that rule. And so forth. NOTE that the "Body" is not a part of the "Header".

Short Primer #01 on Using Regular Expressions (RegEx) with MailWasherPro Filters and Rules – page 5

- **Caution:** Beware of the risk of having 2 pipes together "||" in a string.
    - This is syntactically correct, but such a rule would select everything (i.e. make everything TRUE) – and that might cause unexpected behaviors with a rule or a filter.

- I would recommend that you create a separate rule for more complex situations such as the rule (#5) above

----

8. **Recommendation:** Test your regular expressions as you develop them…

- There are several resources for testing Regular Expressions (RegEx)
    - http://www.regextester.com/ - online testing on this webpage
    - http://regexpstudio.com/TRegExpr/TRegExpr.html - downloadable .exe file
    - I have found that their performance is about 95% the similar - There are some slight differences:
        - The online testing program seems to have no size limit to the expression or the string to be tested – but the .exe program seems to have a size limit to both the expression and the string to be tested
        - The .exe program allows you to set the "/I" switch which ignores case of the text and treats all characters as lower case. The online program supposedly has the capability to set switches but I've not found a way to set the "/I" switch on the test string. The work-around for the upper/lower case issue with the online program is to convert the test data to all lower case before testing
        - I have found two test cases where the online tester finds a match but the .exe program does not – this is a very esoteric RegEx with a very complex test pattern that you probably won't encounter
        - I've found one test case where the online tester finds no error in the expression but where the .exe tester finds an error in the same expression – Not sure of the reason for this yet or whether this behavior is related to the match/non-match difference or the size limitations in the .exe version