

¿Cómo escribo los filtros más eficientes del mensaje?

Contenido

[Pregunta](#)

Pregunta

¿Cómo escribo los filtros más eficientes del mensaje?

Mientras que los filtros del mensaje consiguen más de largo, pueden afectar a las características del rendimiento de su ESA. Para una pequeña cantidad de filtros o los filtros del cortocircuito, la eficacia no es una preocupación significativa. Sin embargo, al construir filtros más largos o si su implementación tiene muchos filtros, usted debe ser atento de la eficacia relativa de ciertas operaciones.

Al pasar los mensajes a través de la tubería del mensaje, todos los filtros del mensaje se combinan en una sola expresión que se evalúe de una manera atómica contra cada mensaje. Esto significa que el ordenar de los filtros es muy importante, y puede cortocircuitar la evaluación adicional de la expresión combinada. Por ejemplo, si usted tiene varios filtros que se apliquen a los mensajes, pero un filtro se aplicará muy con frecuencia y tendrá un `deliver()` de la última acción, un `bounce()`, o un `drop()` asociado a él, que el filtro se debe mover tan temprano en la lista como sea posible.

Aunque el ESA sea muy eficiente en su proceso de las expresiones normales, usted puede abusar del motor de la expresión normal a fin de causar el proceso adicional o innecesario. Cada evaluación de una expresión normal toma áspero la misma cantidad de recursos, así que significa que eso la reducción del número de expresiones que usted evalúa rendirá a mayor eficacia. Por ejemplo, en el filtro siguiente, las expresiones normales en cada “descenso-conexión-por-nombre” son todas evaluadas individualmente, significando que ocurre la evaluación de la expresión normal 7 veces al comparar el nombre de la conexión contra el modelo en el descenso-conexión-por-nombre:

```
strip_all_dangerous: si (verdad) {
¿descenso-conexión-por-nombre ("(? i) \ \ .pif$");
¿descenso-conexión-por-nombre ("(? i) \ \ .exe$");
¿descenso-conexión-por-nombre ("(? i) \ \ .scr$");
¿descenso-conexión-por-nombre ("(? i) \ \ .msi$");
¿descenso-conexión-por-nombre ("(? i) \ \ .java$");
¿descenso-conexión-por-nombre ("(? i) \ \ .dll$");
¿descenso-conexión-por-nombre ("(? i) \ \ .com$");
}
```

En el siguiente ejemplo, los resultados son equivalentes, pero el ejemplo es mucho más eficiente, causando solamente una sola evaluación de la expresión normal:

```
strip_all_dangerous: si (verdad) {
```

```
¿descenso-conexión-por-nombre ("(? i) \ \. (pif|exe|SCR|msi|Javas|DLL|COM) $");  
}
```

Aunque la segunda expresión normal sea más compleja que siete en el primer filtro, es mucho más eficiente evaluar una expresión normal compleja que siete unas simples.

Sin embargo, esta técnica debe ser equilibrada contra el coste de mantener tal filtro.