Содержание

Вопрос

Как я пишу более эффективные фильтры сообщения?

Поскольку фильтры сообщения становятся более длинными, они могут влиять на характеристики производительности вашего ESA. Для небольших чисел фильтров или коротких фильтров, эффективность не является значительным беспокойством. Однако при построении более длинных фильтров или если реализация имеет много фильтров, необходимо помнить относительную эффективность определенных операций.

При передаче сообщений через конвейер сообщения все фильтры сообщения объединены в отдельное выражение, которое оценено атомарным способом против каждого сообщения. Это означает, что заказ фильтров очень важен, и может закоротить дальнейшую оценку объединенного выражения. Например, если у вас есть много фильтров, которые применятся к сообщениям, но один фильтр будет применяться очень часто и иметь заключительное действие, поставляют (), сильный удар (), или отбрасывание () привязанный к нему, тот фильтр должен быть перемещен максимально рано в списке.

Несмотря на то, что ESA очень эффективен в своей обработке регулярных выражений, можно злоупотребить механизмом регулярного выражения таким способом как для порождения дополнительной или ненужной обработки. Каждая оценка регулярного выражения берет примерно одинаковую часть ресурсов, что означает, что, сокращая количество выражений вы оцениваете, приведет к большей эффективности. Например, в следующем фильтре, регулярные выражения в каждом "drop-attachments-by-name" все оценены индивидуально, означая, что оценка регулярного выражения происходит 7 раз при сравнении имени вложения с образцом в drop-attachments-by-name:

```
strip_all_dangerous: if (true) {
drop-attachments-by-name('(?i)\\.pif$');
drop-attachments-by-name('(?i)\\.exe$');
drop-attachments-by-name('(?i)\\.scr$');
drop-attachments-by-name('(?i)\\.msi$');
drop-attachments-by-name('(?i)\\.java$');
drop-attachments-by-name('(?i)\\.dll$');
drop-attachments-by-name('(?i)\\.com$');
}
```

В следующем примере результаты эквивалентны, но пример очень более эффективен, вызывая только одиночную оценку регулярного выражения:

```
strip_all_dangerous: if (true) {
drop-attachments-by-name('(?i)\\.(pif|exe|scr|msi|java|dll|com)$');
}
```

Несмотря на то, что второе регулярное выражение более сложно, чем эти семь в первом фильтре, это очень более эффективно для оценки одного сложного регулярного

выражения, чем семь простых.

Однако этот способ должен быть сбалансирован относительно стоимости поддержания такого фильтра.