Математические вопросы кибернетики и вычислительной техники 24, 2005, 120-124.

О генерации среды фильтрации избыточной информации с использованием шаблонов

А. Авагян, А. Милетбашян

Ереванский государственный университет,
Государственный инженерный университет Армении

Аннотация

Существующие методы проектирования систем фильтрации избыточности не всегда справляются с интенсивными изменениями форм избыточной информации (спама). Предлагаемая нами методология проектирования таких систем построена на инфраструктуре фильтрации (FI), основанной на обработке шаблонов. Эта инфраструктура позволяет гибко настраиваться на новые формы и тем самым эффективно фильтровать избыточную информацию.

1. Введение

Развитие интернета привело к появлению определенных негативных побочных явлений. Одно из них – это получение избыточной незапрашиваемой информация, которую часто называют спамом [1].

Сегодня доля спама составляет 30-80% трафика электронной почты. При сохранении современных тенденций, по данным [1,2], в ближайшем будущем каждое второе электронное письмо будет содержать спам. Учитывая сложившуюся ситуацию, актуальность защиты от непрошеных электронных рассылок приобретает особенно острое значение.

Спам постоянно меняет свои формы, отличается высокой изменчивостью внешних признаков и широким арсеналом лингвистических и технических уловок для обхода спам-фильтров [1,3]. Таким образом, необходимость непрерывной работы по созданию защиты от самых последних образцов ставит перед разработчиками антиспамовских систем уникальные задачи. В то же время, антиспам должен обеспечить пользователей эффективной системой противостояния всем существующим видам спама. Для этого необходимо постоянное обновление технологий распознавания и фильтрации спама в системах антиспама.

Высокая эффективность некоторых систем антиспама достигается благодаря регулярному пополнению базы контекстной фильтрации. Пополнение базы новыми компонентами для эвристического анализа может осуществляться как автоматически через интернет, так и вручную [4,5]. Ручное обновление позволяет администратору сети добавлять в базу свои собственные образцы нежелательной информации. Фильтры систем постоянно обновляются - оперативно реагируя на изменения технологий спамеров. Другие средства защиты от спама имеют встроенный механизм анализа сообщений основанный на оптимизированных нейро-сетях, способных обучаться отличать обычные письма от спама [4,5].

Весь процесс фильтрации может состоять из последовательности шагов, где выход

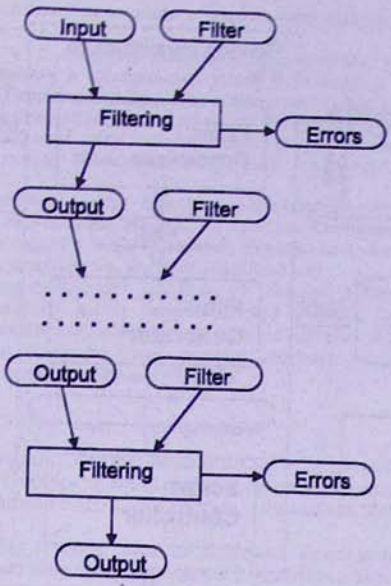

Рис. 1 Схема последовательной фильтрации

предыдущей фильтрации использован на входе следующей фильтрации. Данный процесс схематически представлен на Рис 1.

Доступные из литературы системы защиты от спама, на наш взгляд, основаны на функционально похожих механизмах фильтрации.

В данной работе предлагается методология проектирования антиспамовских систем, гибко настраиваемых на новые формы спама, и тем самым эффективно справляющие с задачей фильтрации.

2. Описание методологии

Предлагаемая нами методология проектирования унифицированной оболочки для генерации различных сред фильтрации избыточной информации, основана на следующей инфраструктуре фильтрации, далее называемой FI (Filtering Infrastructure): фильтр-процессора FP(Filter Processor), генератора фильтров FG(Filter Generator), системы интерпретации управляющих скриптов SC(Script Controller), системы генерации результатов, расширяемых наборов правил интерпретации и генерации результатов. Эта инфраструктура представлена на Рис 2.

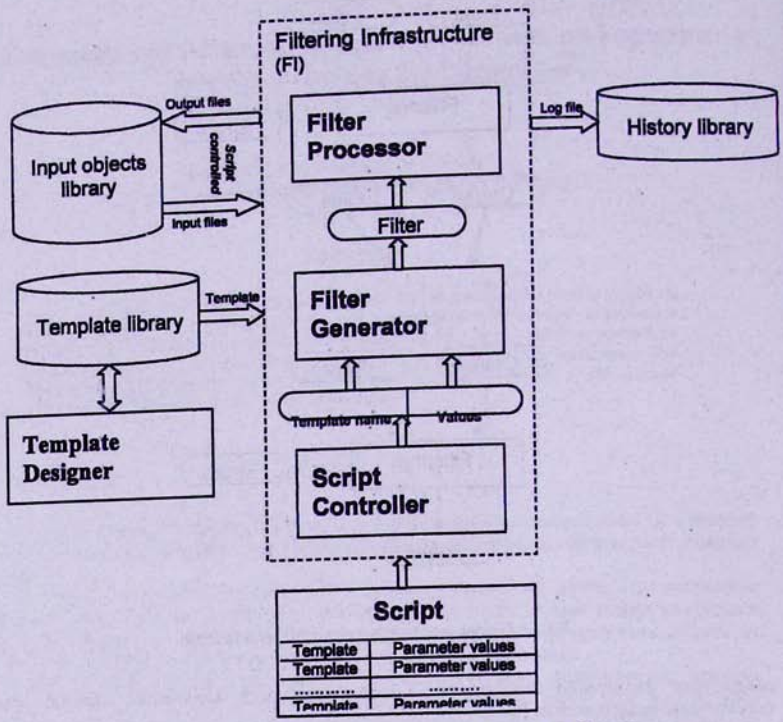

Рис. 2 Схема фильтрации с использованием шаблонов

Фильтр-процессор производит фильтрацию входной информации, принимая во внимание сгенерированные фильтры (правила фильтрации).

Управляющие скрипты, правила фильтрации и генерации результатов будут именоваться шаблонами.

Генератор фильтров создает фильтры из шаблонов и задаваемых параметров.

Система интерпретации управляющих скриптов и правил интерпретации управляет логикой применения шаблонов фильтрации для входной информации. Шаблоны фильтрации описывают различные ограничения, относительно формата и содержания входной информации.

На входе FI получает шаблоны фильтрации, управляющий скрипт и исходные объекты. На выходе FI получаются отфильтрованные и ошибочные объекты.

Для автоматизации очередности интерпретаций предлагается обработчик скриптов в виде надстройки над FI. Обработчик скриптов контролирует очередность интерпретаций, входы для каждой интерпретации и перенаправляет выход интерпретации либо в библиотеку данных, либо в библиотеку истории ошибок. Управление процессом последовательных интерпретаций определяется в скрипте, передаваемом на вход обработчику скриптов. Скрипт содержит информацию о последовательности интерпретации шаблонов и о параметрах фильтрации, а так же

определяет куда должен быть перенаправлен выход фильтрации: в библиотеку исходных данных или библиотеку истории ошибок.

Одни и те же шаблоны фильтрации и правила генерации результатов могут входить в состав разных шаблонов.

Конфигурируемость с помощью шаблонов обеспечивает необходимую гибкость и адаптируемость к изменениям в спецификации среды фильтрации с помощью языка описания шаблонов FTL(Filtering Template Language). При помощи различных значений параметров одни и те же шаблоны фильтрации могут быть применимы для разных видов спама.

Новые среды фильтрации могут генерироваться на базе уже имеющихся шаблонов или посредством создания новых. Изменение среды будет отражаться только в шаблонах, без изменения самого FI.

Использование шаблонов также позволяет разделить проектирование на максимально независимые части для параллельной реализации, позволив нескольким группам разработчиков независимо работать над одним и тем же проектом. Разработчики имеют возможность изменять среду, используя единую методологию, не делая изменений в FI.

Дизайнеры шаблонов фильтрации, управляющих скриптов и правил генерации результатов имеют возможность определять логику независимо друг от друга. Предлагаемый метод - при помощи параметров - дает возможность многоразового использования одних и те же шаблонов в разных контекстах. При создании среды имеется возможность определения стандартных шаблонов, используемых в многочисленных проектах.

3. Заключение

С точки зрения реализации, система фильтрации является настраиваемым интерпретатором, использующим правила проверки и интерпретации. Реализация системы обработки шаблонов основана на XML технологии [7-11]. Таким образом, описываемая методология будет обладать следующими свойствами:

Общее решение. Для создания оболочек различных видов избыточности разработчикам необходимо изучить только одну среду разработки. Разработчики имеют возможность изменять оболочку, используя единую методологию, не делая изменений в FI.

Разделение логики применения и интерпретации. Дизайнеры имеют возможность определять логику шаблонов и управляющих скриптов независимо друг от друга.

Разделение частей проекта. Предлагаемый метод позволяет разделить проект на отдельные части, что облегчает его поддержку и дает возможность многоразового использования одних и те же шаблонов в разных контекстах.

Глобальный контроль над оболочкой. При создании оболочки имеется возможность определения стандартных шаблонов, которые могут использоваться многократно.

Таким образом, изменения в подобном шаблоне в одном из проектов автоматически производятся и в остальных проектах, что сокращает время разработки и снижает вероятность ошибок при проектировании оболочки.

Повторное использование шаблонов. Правила интерпретации и генерации фильтров могут комбинироваться и использоваться разработчиками при создании разных фильтров.

Литература

[1] Alan Schwartz. SpamAssassin. - O'Reilly; 1 edition (July, 2004). – 207p.
[2] Paul Wolfe, Charlie Scott, Mike Erwin. Anti-Spam Tool Kit. - McGraw-Hill Osborne Media; 1 edition (March 17, 2004). – 400p.
[3] P. Lutus, The Anti-Spam http://www.arachnoid.com/lutusp/antispam.html
[4] SecurityLab, Обзор 11-ти антиспамовских продуктов, http://www.securitylab.ru/50190.html
[5] Алексей Тутубалин, Технология SPF, «Вебпланета». http://www.webplanet.ru/news/reading-room/2004/7/23/spf.html
[6] Gina M. LaPlante, How Private is Your Work Email?, November 14, 2000, http://www.tilj.com/content/ecomarticle11140001.htm

[7] Elliotte Rusty Harold, W. Scott Means. *XML in a Nutshell*. 2nd edition, O'Reilly & Associates, June 2002.
[8] Н. Питц-Моултис, Ч. Кирк. *XML в подлиннике*. - BHV-СПб, август 2000.
[9] Д. Мартин, М. Бирберк, Б. Лозген, Дж. Пиннок, С. Ливингстон, П. Старк, К. Уильямс, Р. Андерсен, С. Мор, Д. Балилес. *XML для профессионалов*. - Лори, апрель 2001.
[10] М. Даконта, А. Дж. Саганич. *XML и Java 2*. - Питер, май 2001.
[11] Brett McLaughlin. *Java & XML 2nd Edition: Solutions to Real-World Problems*. September 2001.

Շաբլոնների միջոցով ավելցուկային ինֆորմացիայի զտման միջավայրի գեներացիայի մասին

Ա. Ավագյան, Ա. Միլեթքաշյան

Ամփոփում

Զտման համակարգերի նախագծման գոյություն ունեցող մեթոդները միշտ չէ որ կարողանում են հաղթահարել ավելցուկային ինֆորմացիայի (spam) ձևերի իմտենսիվ փոփոխությունները: Նման համակարգերի մեր կողմից առաջարկվող մեթոդաբանության հիմքում ընկած է շաբլոնների մշակման վրա հիմնված զտման ենթակառուցվածքը (FI): Այս ենթակառուցվածքը թույլ է տալիս ճկում հարմարվել նոր ձևերին, այդպիսով թույլ տալով արդյունավետ զտել ավելցուկային ինֆորմացիան: