УДК 372.016:004

А.А. Козинский, В.И. Басин

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ВАЛИДАЦИИ HTML-КОДА В СООТВЕТСТВИИ С ЗАДАННЫМ НАБОРОМ ПРАВИЛ МОДУЛЯ BRAINTRAINING.SECURITY

Описаны особенности библиотеки BrainTraining.Security для валидацииHTMLи CSSкода на соответствие веб стандартам и выбранному типу документа. Применение библиотеки позволяет защитить приложение от несанкционированного доступа к учетным записям пользователей, которые открывают страницы с публикациями. Созданный модуль библиотеки предназначен для платформы .NET. Модуль позволяет выполнять валидациюв соответствии с набором заданных правил. Перечень правил содержится в xml-файле.Для унификации и обеспечения совместимости схемы описания правил выполнены по аналогии симеющимися подобными библиотеками. Приведен краткий перечень реализованных правил библиотекиBrainTraining.Security их блок-схемы и код.

Постановка проблемы

При загрузке веб-приложений обрабатывается html-код, который в современных приложениях может передаваться от клиента к серверу. Результаты интерпретации кода отображаются на страницах приложений сервера, доступных для изменения со стороны клиента. Например, при создании публикации блога, комментария или вопроса форума их содержание передается как блок html-кода для сохранения в базе данных. Для задач такого рода необходимо выполнять валидацию html-кода. Под валидацией будем понимать проверку HTML и CSS кода на соответствие веб стандартам и выбранному типу документа. Html-приложение без проведения валидации становится уязвимым для XSS-атак (CrossSiteScripting – «межсайтовый скриптинг»). Например, злоумышленник пересылает на сервер html-код, содержащий скрипт, который отправляет содержимое cookie-файлов браузера на компьютер клиента. Если в cookie-файлах хранится информация об авторизированном пользователе (сессия, пароль в зашифрованном виде), то злоумышленник получает несанкционированный доступ к учётным записям других пользователей, открывших страницу с публикацией, комментарием или вопросом форума. Несанкционированный доступ возможен, например, при создании копий cookie браузером клиента-злоумышленника.

Постановка задачи

Создать модуль (библиотеку) для платформы .NET, которая позволит производить валидациюhtml-кода в соответствии с заданными правилами. Правила описывают разрешённые теги, атрибуты и стили, их возможные значения и шаблоны значений, разрешенных к использованию. Дополнительно необходимо обеспечить «чистку» кода – удаление запрещённых правилами фрагментов html. Результатом работы модуля валидации должно быть безопасное для использования в сети содержимое.

Обзор существующих решений

Существует ряд библиотек, которые решают поставленную задачу. К ним относятся следующие:

1. MicrosoftWebProtectionLibrary [1]. Библиотека написана для платформы .NET. Html-код после проверки с помощью функций указанного модуля не содержит опасных блоков. В числе недостатков указанного решения также укажем, например, недопусти-

мость использования стилей CSS в атрибутах style тегов проверяемого исходного кода; отсутствие возможности определять списки разрешённых в коде приложения тегов, атрибутов и стилей. В библиотеке MicrosoftWebProtectionLibrary пользователь не имеет возможности самостоятельно определять перечень разрешенных тегов.

2. OWASPAntiSamy[2]. Библиотека написана на Java. Она обладает широким функционалом. К ее особенностям отнесем низкую скорость работы и отсутствие возможности использования функций библиотеки в приложениях .NET. Существует «порт» библиотеки для языка C# – AntiSamy.NET (перенос основного функционала с одной платформы или языка на другой). Однако использование порта затруднено из-за существующих проблем. Так, версия «порта» для .NET не поддерживается производителем, так как считается устаревшей. Кроме того, OWASPAntiSamy использует библиотеку vjslib.dll. Для использования последней необходимо устанавливать дополнительные компоненты (MicrosoftVisual J# RedistributableLibrary). Практика работы с указанным решением показала, что в большом числе различных приложений пользователя генерируются исключения при использовании методов библиотеки vjslib.dll.

Обоснование представленного решения

Библиотека BrainTraining.Security, созданная В.И. Басиным, реализует поставленную задачу без необходимости установки сторонних компонент (кроме самого .NET версии 4). Предлагаемое решение просто в использовании, легковесно, содержит подробную документацию и обладает высокой производительностью. Библиотека успешно используется на протяжении одного года на сайте BrainTraining [3] в блогах и форуме для обеспечения безопасности отображаемого содержимого приложений.

Краткое описание предлагаемого решения

В библиотеке BrainTraining.Security валидация текста проводится в соответствии с набором заданных правил. Перечень правил содержит xml-файл, схема которого представлена на рисунке 1.

Для унификации и обеспечения совместимости с библиотекой OWASPAntiSamy схемы описания правил BrainTraining.Security выполнены по аналогии. Аналогия означает, что правила разработанные для одной из указанных библиотек могут быть без модификации использованы в другой. Приведем краткий перечень реализованных правил библиотеки BrainTraining.Security (рисунок 1):

– directives – настройка проверки общих параметров кода (максимальный размер проверяемого кода, указание на разрешение импорта стилей и другие общие правила);

– common-regexp – набор регулярных выражений, которые используются для проверки;

– common-attributes – описание возможных атрибутов;

– global-attributes – набор атрибутов, которыми могут обладать элементы html-кода;

– tag-rules – правила обработки всех разрешённых тегов;

– css-rules – правила обработки стилей;

– allowed-empty-tags – список разрешённых тегов, для которых допускается отсутствие содержания.

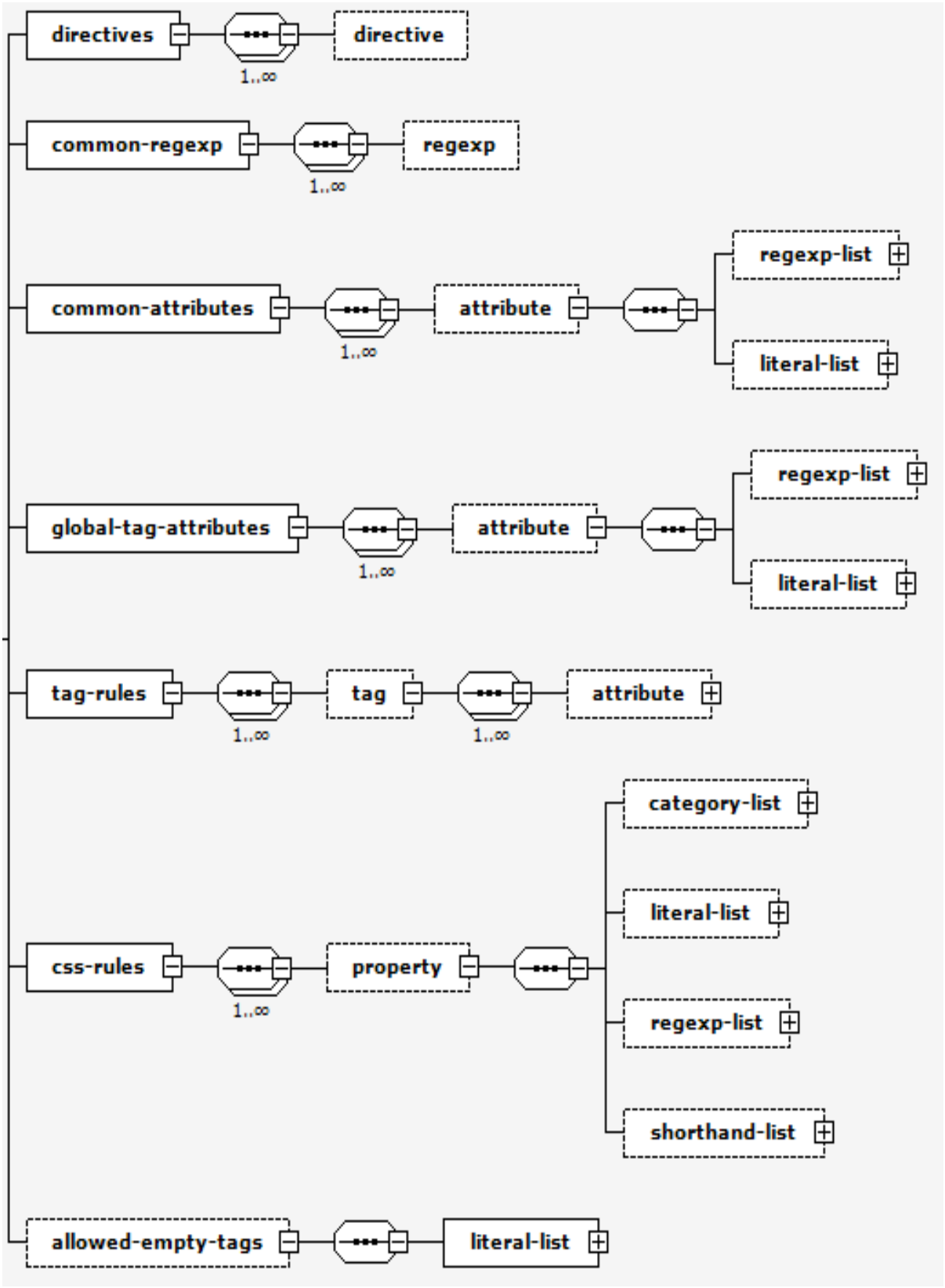

Рисунок 1 – Схема правил содержащихся в xml-файле

Фрагмент regexp-list (рисунок 1) задаёт множество регулярных выражений, которым должен удовлетворять атрибут или тег валидируемого кода. Список literal-list представляет перечень возможных значений атрибутов.

Библиотека BrainTraining.Security имеет следующие режимы проверки тегов, включаемые в описания правил валидации:

– Remove – удалять тег при встрече;

– Filter – удалить тег, оставив его содержимое;

– Validate – проверить на удовлетворение правилам, и удалить атрибуты и стили не прошедшие валидацию;

– Truncate – удалить все атрибуты и вложенные теги, оставив только содержимое текущего.

Приведём фрагмент кода чтения и анализа правил обработки атрибутов.

Ниже (Листинги 1 и 2) приведены методы, описанные в классе XmlHelper, которые проходят все элементы блоков xml с именем collectionName. Для каждого вложенного элемента (elementsName) запускается обработчик (XmlElementAnalyzer).

Листинг 1 – «Парсинг правил валидации файла конфигурации»

```
publicdelegatevoidXmlElementAnalyzer(XElementcurrentElement); //описание методов, которые могут быть обработчиками

public static void Aggregate(IEnumerable<XElement>itemCollections, string collectionName, string elementsName, XmlElementAnalyzer analyzer)
{
foreach (XElementitemCollection in itemCollections)
{ //для всехуказанных блоков
IEnumerable<XElement> items = itemCollection.Elements(elementsName); //получаем вложенные элементы
foreach (XElement item in items)
{ //для каждого вложенного элемента
analyzer.Invoke(item); //запускаем обработчик
}
   }
}
public static void Aggregate(XElement element, string collectionName, string elementsName, XmlElementAnalyzer analyzer)
{
IEnumerable<XElement>itemCollections = element.Elements(collectionName); //просматриваемые блоки
Aggregate(itemCollections, collectionName, elementsName, analyzer); //обработка каждого блока
}
```

Фрагмент блок-схемы проверки на корректность кода тега приведен на рисунке 2.

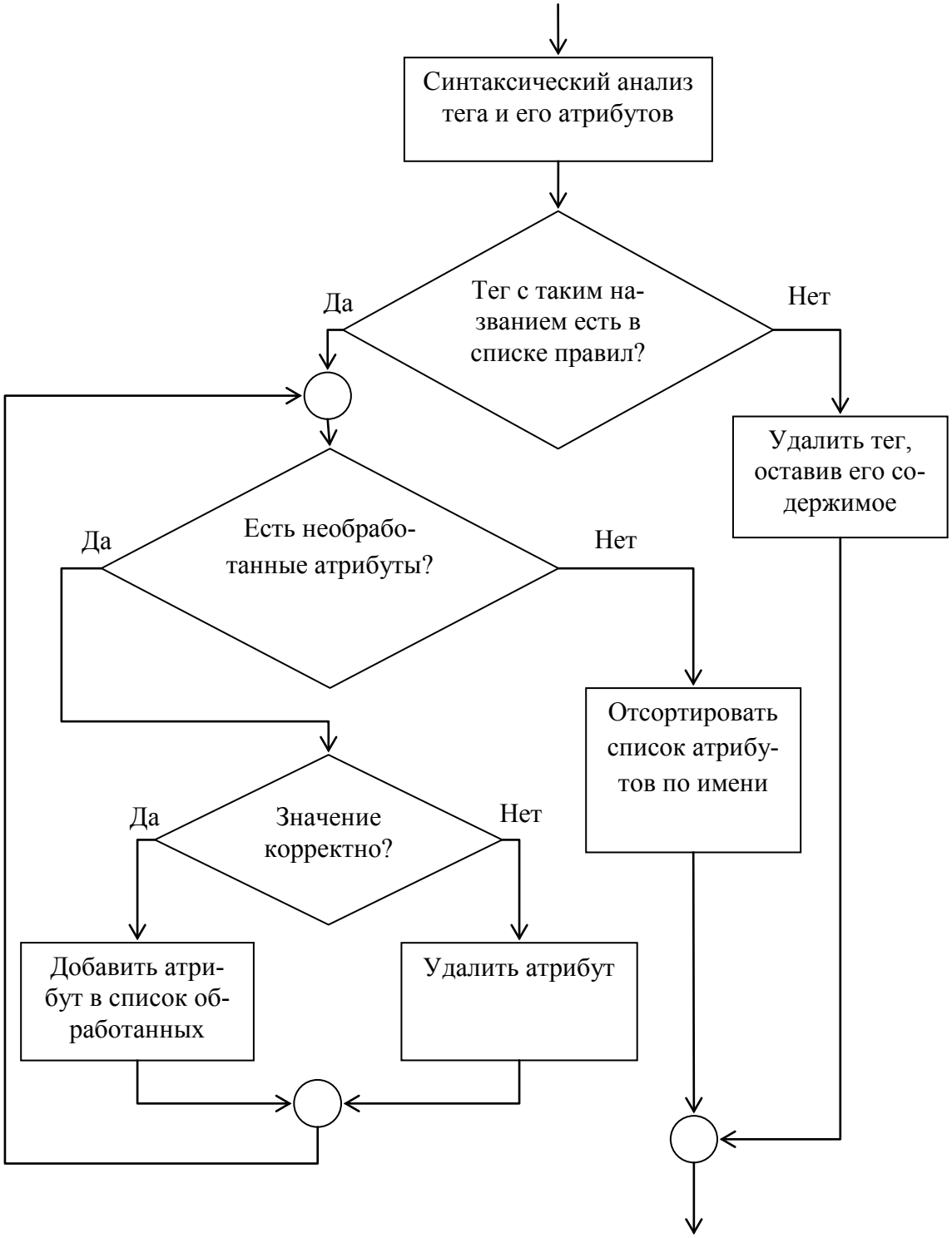

Рисунок 2 – Фрагмент блок-схемы обработки тега

Листинг 2 – «Обработка допустимых атрибутов правил валидации файла конфигурации»

```
private void ParseCommonAttributes(XDocument document)
{
XmlHelper.Aggregate(document, "common-attributes", "attribute", ProcessAttribute);
//пробегаем правила всевозможных атрибутов и вызываем для каждого обработчик
}
```

```
private void ProcessAttribute(XElement element) //обработчик элемента
{
stringnameAttr = element.Attribute("name").Value; //получаем имя

Model.Attribute attribute = new Model.Attribute(nameAttr); //создаём класс, представляющий атрибут

XmlHelper.Aggregate(element, "regexp-list", "regexp", ProcessRegexpValue); //обрабатываем правила проверки значений
XmlHelper.Aggregate(element, "literal-list", "literal", ProcessLiteralValue); //обрабатываем возможные значения

  _atrributesList.AddItem(attribute); //добавляем в коллекцию считанных атрибутов
}
```

Описанная модель и ее реализация прошли апробацию в ходе работы ресурса для программистов. Результаты работы над указанным ресурсом были описаны ранее в [4–6].

СПИСОК ЛИТЕРАТУРЫ

1. Официальный репозиторий библиотеки MicrosoftWebProtectionLibrary [Электронный ресурс] / Microsoft. – Режим доступа : http://wpl.codeplex.com,свободный. – Дата доступа : 24.09.2012.

2. Официальный репозиторий библиотеки OWASPAntiSamy [Электронный ресурс] / OWASP. – Режим доступа : http://code.google.com/p/owaspantisamy/, свободный. – Дата доступа : 24.09.201.

3. Басин, В.И. Сайт для программистов и дизайнеров пользовательского интерфейса BrainTraining [Электронный ресурс] / В.И. Басин. – Режим доступа : http://brtrg.com, свободный. – Дата доступа : 24.09.2012.

4. Басин, В.И. Brain Training – a new approach to testing your sports programming skills / В.И. Басин // Иностранные языки и современный мир :сб. материалов междунар. науч. конф. студентов, Брест, 14 апреля 2011 г : в 2 ч. – Брест : БрГУ, 2011.– Ч. 1. – С. 13–15.

5. Басин, В.И. Сайт для организации олимпиад по спортивному программированию / В.И. Басин // Современные проблемы математики и вычислительной техники : материалы VII Респ. науч. конф. молодых ученых и студентов, Брест, 26–28 ноября 2011 г. : в 2-х ч. – Брест : БрГТУ, 2011. – Ч. 2 – С. 4–6.

6. Басин, В.И. Соревнования по спортивному программированию в системе тестирования BrainTraining / В.И. Басин // XIV Респ. науч.-метод. конф. молодых учёных : сб. материалов : в 2 ч. – Брест : БрГУ, 2012. – Ч. 1. – С. 24–26.

A.A. Kazinski, V.I. Basin. **A Mathematical Model of the Validation of HTML-code in Accordance with a Given Set of Rules Module BrainTrainingSecurity**

This article contains the description of the possibilities of the BrainTraining.Security library for HTML and CSS code validation with given set of rules and document specifications. This library can provide protection for web applications against unauthorized access to the account of the user who is trying to view some page content with untrusted html code (e.g. publications in personal blog). This library is designed for applications under.NETplatform. The module allows you to perform validation in accordance with a set of predefined rules. The list of rules is described in xml configuration file. The scheme of this configuration file is similar existing libraries to provide a high level of compatibility. Also a short list of implemented validation rules of the library, block-diagrams and code snippets were described.

Рукапіс паступіў у рэдкалегію 10.10.2012