Математические структуры и моделирование 2019. № 4(52). С. 111–117

УДК 004.622:801.81
DOI 10.24147/2222-8772.2019.4.111-117

ОБФУСКАЦИЯ КОДА В ВЕБ-ПРЕДСТАВЛЕНИИ ДИАЛЕКТНОГО КОРПУСА НАРОДНОЙ РЕЧИ

Д.Н. Лавров
к.т.н., доцент, e-mail: lavrov@omsu.ru
А.П. Лапин
студент, e-mail: aleksandrlapinsanek@gmail.com

Омский государственный университет им. Ф.М. Достоевского

Аннотация. В работе предложен алгоритм обфускации тематической разметки диалектного корпуса, позволяющий защитить работу исследователей от неправомерного копирования данных. Предложенный подход основан на замене реальных названий тем на случайно сгенерированные идентификаторы. Таблица замен сохраняется на сервере и каждый раз при обращении к тексту корпуса создаётся для данного текста заново с генерацией новых идентификаторов. Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 18–012–00519.

Ключевые слова: обфускация кода, тематическая разметка, диалектный корпус.

Введение

В корпусе народной речи [1–4] для представления тематической разметки используются XML-теги. Процесс разметки текста является сложной трудоёмкой работой научного коллектива лингвистов Омского государственного университета им. Ф.М. Достоевского. Авторство этой работы требует защиты. Необходимо построить механизм, затрудняющий копирование размеченных данных корпуса, и в тоже время не затрудняющий работу исследователей.

Цель: защитить данные диалектного корпуса путём запутывания данных при скачивании таким способом, чтобы восстановление разметки было существенно затруднено или даже попросту невозможно.

1. Алгоритм обфускации

По определению: *обфускация* или запутывание кода — это приведение исходного текста или исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции.

Пример обфускации кода XML или HTML разметки:

```
mailto: <b>ivanov@mail.ru</b>
```

для запутывания и усложнения может быть приведён к

```
mailto: <b>i</b><b>v</b><b>a</b><b>n</b><b>o</b>
<b>v</b><b>&#64;</b><b>m</b><b>a</b><b>i</b><b>l</b>
<b>.</b><b>r</b><b>u</b>
```

Как видно из приведённого примера, интерпретация человеком преобразованного кода сильно затруднена, хоть и остаётся с тем же представлением в браузере.

В нашем случае нам также нужно запутать разметку кода тематической разметки. Делается это на этапе преобразования XML-кода в HTML-представление. В предыдущей работе [1] описан механизм создания HTML-представления в виде автомата с конечным числом состояний. Мы покажем, как, не нарушая работу описанных в [1] алгоритмов, модифицировать их так, чтобы добиться обфускации HTML-кода.

Тематическая разметка создаётся тегами:

```
<theme class="родительская_тема--дочерняя_тема">
```

Иерархия тем подразумевает, что у родительской темы может быть несколько подтем. Тема дочерняя вложена в родительскую.

Сформулируем требования к обфускации.

1. При каждом обновлении страницы класс одного и того же тега должен быть разный.
2. Идентификаторы классов генерируются случайным образом.

Действуем по следующему алгоритму:

1. Создаём пустой словарь тем для осуществления замен и стек для отслеживания вложенности тем.
2. В цикле последовательно просматриваем исходный текст.
3. В случае обнаружения открывающего тега `theme` находим его класс.
4. Если класс простой и встретился первый раз, то для него генерируется случайный идентификатор класса, этот идентификатор запоминается в стеке и словаре.
5. После этого класс тега заменяется на хранящееся в словаре значение, дополненное всеми текущими темами из стека.
6. Если класс сложный, то он разбирается на составные части — простые классы — и с ними поступают, как с простыми классами.
7. Если встречается закрывающий тег темы, то идентификатор темы извлекается из вершины стека.
8. После окончания цикла с помощью jquery на основе таблицы замен модифицируем внешний вид новых классов.

Рассмотрим обфускацию на примере следующего кода:

```
<theme class="Статья">
    Эта статья для примера.
    <theme class="Тег">
        В статье есть теги.
```

```
    </theme>
    <theme class="Тег--Сложный">
        Также есть сложные теги.
    </theme>
    <theme class="Обфускация">
        Преобразуя данные в html можно
        сделать обфускацию.
    </theme>
</theme>
```

Создаём пока ещё пустой словарь замен:

Класс	**Замена**

Начинаем последовательный разбор содержания файла в цикле.

Итерация 1. Обнаружен тег `<theme class="Статья">`. В словарь вносится замена для класса `Статья` на `tag_45`. В стек кладём сгенерированный идентификатор темы `tag_45`. Производится замена, XML-документ преобразуется в следующий вид:

```
<ins class="tag_45">
    Эта статья для примера.
    <theme class="Тег">
        В статье есть теги.
    </theme>
    <theme class="Тег--Сложный">
        Также есть сложные теги.
    </theme>
    <theme class="Обфускация">
        Преобразуя данные в html, можно
        сделать обфускацию.
    </theme>
</theme>
```

Словарь замен:

Класс	**Замена**
Статья	tag_45

Состояние стека: "tag_45"

Итерация 2. Найден тег `<theme class="Тег">`. Действуем аналогично предыдущей итерации, атрибут `class` формируется из содержимого стека. Приведём результат замен, состояния словаря и стека.

```
<ins class="tag_45">
    Эта статья для примера.
    <ins class="tag_45 tag_86">
        В статье есть теги.
    </theme>
    <theme class="Тег--Сложный">
```

```
        Также есть сложные теги.
    </theme>
    <theme class="Обфускация">
        Преобразуя данные в html, можно
        сделать обфускацию.
    </theme>
</theme>
```

Словарь замен:

Класс	**Замена**
Статья	tag_45
Тег	tag_86

Состояние стека: "tag_45 tag_86" (вершина стека в конце списка, разделитель элементов — пробел).

Итерация 3. Обнаружен закрывающий тег `</theme>`. Он заменяется на `</ins>`, извлекается из вершины стека идентификатор `tag_86`.
Состояние стека: "tag_45".

Итерация 4. Найден тег `<theme class="Тег--Сложный">`. Его атрибут `class` состоит из двух тем, одна из которых `Тег` — уже в словаре замен. Для новой темы `Сложный` генерируется идентификатор класса, и оба идентификатора складываются в стек в одной связке.

```
<ins class="tag_45">
    Эта статья для примера.
    <ins class="tag_45 tag_86">
        В статье есть теги.
    </ins>
    <ins class="tag_45 tag_86 tag_89">
        Также есть сложные теги.
    </theme>
    <theme class="Обфускация">
        Преобразуя данные в html, можно
        сделать обфускацию.
    </theme>
</theme>
```

Класс	**Замена**
Статья	tag_45
Тег	tag_86
Сложный	tag_89

Состояние стека: "tag_45 tag_86-tag_89".

Обратите внимание, что формирование атрибута `class` осуществляется по-прежнему за счёт содержания стека, но учитывается строение сложного значения атрибута "Тег–Сложный" (tag_86-tag_89).

Итерация 5. Обнаружен закрывающий тег `</theme>`. Он заменяется на `</ins>`, из вершины стека извлекается идентификатор tag_86-tag_89.
Состояние стека: "tag_45".

Итерация 6. Найден тег `<theme class="Обфускация">`.

```
<ins class="tag_45">
    Эта статья для примера.
    <ins class="tag_45 tag_86">
        В статье есть теги.
    </ins>
    <ins class="tag_45 tag_86 tag_89">
        Также есть сложные теги.
    </ins>
    <ins class="tag_45 tag_49">
        Преобразуя данные в html, можно
        сделать обфускацию.
    </theme>
</theme>
```

Класс	**Замена**
Статья	tag_45
Тег	tag_86
Сложный	tag_89
Обфускация	tag_49

Состояние стека: "tag_45 tag_49".

Итерация 7. Обнаружен закрывающий тег `</theme>`. Он заменяется на `</ins>`, из вершины стека извлекается идентификатор tag_49.
Состояние стека: "tag_45".

Итерация 8. Обнаружен закрывающий тег `</theme>`. Он заменяется на `</ins>`, из вершины стека извлекается идентификатор tag_45.
Состояние стека: "tag_45".

Работа алгоритма завершена. Ниже показан результат работы алгоритма.

Было:

```
<theme class="Статья">
  Эта статья для примера.
  <theme class="Тег">
     В статье есть теги.
  </theme>
  <theme class="Тег--Сложный">
     Также есть сложные теги.
  </theme>
  <theme class="Обфускация">
     Преобразуя данные в html,
     можно сделать обфускацию.
    </theme>
</theme>
```

Стало:

```
<ins class="tag_45">
  Эта статья для примера.
  <ins class="tag_45 tag_86">
     В статье есть теги.
  </ins>
  <ins class="tag_45 tag_86 tag_89">
     Также есть сложные теги.
  </ins>
  <ins class="tag_45 tag_49">
     Преобразуя данные в html,
     можно сделать обфускацию.
  </ins>
</ins>
```

Заключение

Теоретически, зная механизм обфускации, тематическую разметку можно восстановить, но это уже не получится сделать простым копированием. Восстановление теперь стало существенно затруднено и приближено к трудозатратам на выполнение самостоятельной тематической разметки.

Для ещё большего усиления защиты от анализа кода в «ручном режиме» генерация идентификаторов тем может быть сделана на основе хэширования названий тем с секретной «солью».

Благодарности

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 18–012–00519.

Литература

1. Лавров Д.Н., Лапин А.П., Харламова М.А., Черкащенко И.А. Обработка текста с фонетической и тематической разметками для отображения текстов диалектного корпуса // Математические структуры и моделирование. 2019. № 3(51). С. 135–144.
2. Лавров Д.Н., Харламова М.А., Костюшина Е.А. Представление разметки корпуса народной речи среднего Прииртышья // Математические структуры и моделирование. 2018. № 4(48). С. 85–91.
3. Харламова М.А., Лавров Д.Н. История полиэтнического региона в «зеркале» народной речи: о проекте регионального диалектного корпуса. Current trends and Future Perspectives in Russian Studies // Proceedings of the International Conference on Russian Studies at the University of Barcelona, MKR-Barcelona. Barselona : Trialba Ediciones 2018. P. 1564–1572.
4. Лавров Д.Н., Харламова М.А., Костюшина Е.А. Модель представления экстралингвистической и тематической разметки в корпусе народной речи // Математическое и компьютерное моделирование : сборник материалов VI Международной научной конференции, посвящённой памяти Б.А. Рогозина (Омск, 23 ноября 2018 г.). Омск : Изд-во Ом. гос. ун-та, 2018. С. 115–118.

CODE OBFUSCATION IN WEB-PRESENTATION OF DIALECT CORPUS OF FOLK SPEECH

D.N. Lavrov
Ph.D.(Eng.), Associate Professor, e-mail: lavrov@omsu.ru
A.P. Lapin
Student, e-mail: aleksandrlapinsanek@gmail.com

Dostoevsky Omsk State University

Abstract. The paper proposes an algorithm for obfuscation of thematic marking of the dialect corps, which allows to protect the work of researchers from illegal copying

of data. The proposed approach is based on replacing real topic names with randomly generated identifiers. The substitution table is stored on the server and each time you access the body text, it is created for this text again with the generation of new identifiers. This work was supported by the Russian Foundation for Basic Research as part of a scientific project No. 18–012–00519.

Keywords: code obfuscation, thematic markup, dialect case.

References

1. Lavrov D.N., Lapin A.P., Kharlamova M.A., and Cherkashchenko I.A. Obrabotka teksta s foneticheskoi i tematicheskoi razmetkami dlya otobrazheniya tekstov dialektnogo korpusa, Matematicheskie struktury i modelirovanie, Omsk, Om. Gos. Un-t. Publ., 2019, no. 3(51), pp. 135–144. (in Russian)
2. Lavrov D.N., Kharlamova M.A., and Kostyushina E.A. Predstavlenie razmetki korpusa narodnoi rechi srednego Priirtysh'ya. Matematicheskie struktury i modelirovanie, Omsk, Om. Gos. Un-t. Publ., 2018, no. 4(48), pp. 85–91. (in Russian)
3. Kharlamova M.A. and Lavrov D.N. Istoriya polietnicheskogo regiona v ≪zerkale≫ narodnoi rechi: o proekte regional'nogo dialektnogo korpusa. Current trends and Future Perspectives in Russian Studies. Proceedings of the International Conference on Russian Studies at the University of Barcelona, MKR-Barcelona, Barselona, Trialba Ediciones 2018, pp. 1564–1572. (in Russian)
4. Lavrov D.N., Kharlamova M.A. and Kostyushina E.A. Model' predstavleniya ekstralingvisticheskoi i tematicheskoi razmetki v korpuse narodnoi rechi. Matematicheskoe i komp'yuternoe modelirovanie : sbornik materialov VI Mezhdunarodnoi nauchnoi konferentsii, posvyashchennoi pamyati B.A. Rogozina (Omsk, 23 noyabrya 2018 g.), Omsk, Izd-vo Om. gos. un-ta, 2018, pp. 115–118. (in Russian)

Дата поступления в редакцию: 29.05.2019