УДК 004.7.056.53

МЕТОДЫ ЗАЩИТЫ МЕТАДАННЫХ В ФОРМАТЕ XML

В.И. Воробьев[1], Т.В. Монахова[2]

[1] *Санкт-Петербургский институт информатики и автоматизации Российской академии наук, Санкт-Петербург, Россия*
vvi@iias.spb.su

[2] *Центральный научно-исследовательский институт Минобороны РФ, Королев, Россия*
monakhova_t81@mail.ru

Аннотация

В статье рассматривается защита метаданных, представленных в формате XML и родственных языков. При этом данные о предметной области упорядочиваются с применением онтологических методов. Предложена трёхкомпонентная онтологическая модель системы защиты данных на основе онтологических представлений данных о предметной области в части защищаемых данных и потенциальных угроз. Разработан шаблон классификации данных, который позволяет детализировать соответствующие классы, вносить конкретные элементы данных и отсекать неиспользуемые классы или их подклассы. На основе онтологических представлений защищаемых данных и актуальных угроз в соответствии с политикой безопасности строится онтологическая модель средств защиты, реализуемых в разрабатываемой системе. Обсуждены языки описания метаданных. Для защиты XML-документа предлагается использовать методы обфускации и текстовой стеганографии. Предложен алгоритм модифицированного метода обфускации со случайной выборкой части кода. Построена блок-схема алгоритма, пригодного для проектирования средств защиты метаданных. Пояснен отказ от использования в данном случае документа XML в роли стегоконтейнера. Разработаны рекомендации по способу и последовательности применения онтологических методов защиты метаданных. Приводится описание особенностей применения методов обфускации и текстовой стеганографии.

Ключевые слова: метаданные, онтологическая модель, система защиты данных, XML, обфускация, стеганография, большие данные.

Цитирование: *Воробьев, В.И.* Методы защиты метаданных в формате XML / В.И. Воробьев, Т.В. Монахова // Онтология проектирования. – 2018. – Т. 8, №2 (28). - С.253-264. – DOI: 10.18287/2223-9537-2018-8-2-253-264.

Введение

Работа в любой из предметных областей (ПрО) в сфере науки, техники или коммерции связана с обработкой разнообразных данных. При этом значительное количество обрабатываемых данных составляют данные, требующие защиты. Это не только персональные данные, но и данные, являющиеся предметом интеллектуальной собственности, в том числе таких её направлений, как государственная и коммерческая. Риски, возникающие в связи с применением метаданных, можно условно разделить на две группы: внедрение кода и раскрытие ценной информации. Большинство таких данных имеют разные форматы, типы и относятся к разным ПрО. Данные также могут дублироваться в силу различных причин, к ним могут применяться различные процедуры обработки. Соответственно, обработка получаемого набора данных должна начинаться с систематизации и упорядочения с учётом форматов, происхождения и способов обработки. При этом выявляются обобщённые данные об отдельных подмножествах данных о ПрО, другими словами, метаданные. Защита метаданных является важным направлением исследований.

1 Онтологический подход к определению метаданных

В настоящее время в подавляющем большинстве ПрО используют методы больших данных. Количество таких ПрО и объёмы данных неуклонно возрастают, что делает обработку больших данных актуальным и приоритетным направлением исследований. Это привело к появлению таких технологий, как Data Mining, Big data, RDF, XML, Semantic Web [1]. При этом характерной особенностью работы с данными является работа со слабо структурированными данными, что привело к появлению технологий класса Semantic Web, RDF и XML [2]. К большим данным применяют определённый набор методов и техник анализа, среди которых: методы класса Data Mining, распознавание образов, прогнозная аналитика, визуализация аналитических данных и др. Примером технологий и инструментов работы с большими данными является Hadoop [3]. Успешность проекта связана с тем, что он разработан на языке Java в рамках вычислительной парадигмы MapReduce, согласно которой приложение разделяется на большое количество одинаковых элементарных заданий, выполнимых на узлах кластера и затем сводимых в конечный результат. Приведённые сведения о проекте Hadoop объясняют его ценность для обработки больших данных – кластерная обработка позволяет существенно сократить временные затраты на обработку больших объёмов данных, что особенно важно при условии высокой скорости изменения больших данных. Защита Big Data существенно отличается от защиты, рассчитанной на обработку только обычных данных (отличающихся набором признаков - volume, velocity, variety).

Безопасность больших данных имеет два направления: управление безопасностью больших данных; разработка и применение средств защиты больших данных. Разработка системы защиты больших данных должна производиться с использованием ряда методов и техник анализа, применимых к большим данным. Одним из таких методов является онтологический анализ и описание данных с выделением метаданных [4]. Преимущества онтологического подхода состоят в гибкости онтологии, т.е. возможности быстрого изменения, в том числе добавления новых элементов данных без кардинальной переработки уже созданной онтологии. Кроме того, онтологический анализ данных позволяет разделить их на некоторые классы, что, в свою очередь, даёт возможность разработки процедур обработки данных, принадлежащих к одному классу [5].

Модель ПрО можно представить в следующем виде [6]: $M_n = (F,T,U,I,R)$, где

$F = \{f_a \mid a = \overline{1, A}\}$ - множество функций системы;

$T = \{t_i \mid j = \overline{1, J}\}$ - множество задач обработки информации;

$U = \{u_k \mid k = \overline{1, K}\}$ - множество пользователей;

$I = I^{вх} \bigcup I^{вых}$ - множество данных ПрО;

$I^{вх} = \{i_x^{вх} \mid x \circ X^{вх}\}$ - множество данных, необходимых для обеспечения информационных потребностей системы;

$I^{вых} = \{i_x^{вых} \mid x \circ X^{вых}\}$ - множество данных, являющихся результатом взаимодействия пользователей и функций системы;

$R = \{r_l \mid l = \overline{1, L}\}$ - множество отношений между компонентами F, T, U, I.

На основе указанной модели осуществляем переход к описанию семантики онтологии:

$O = \langle F, V, S, H \rangle$, где

F - множество функций, выполняемых системой;

V - множество определений указанных функций;

S - множество отношений между функциями;

H - множество правил использования функций системы, что позволяет разделить её на составляющие элементы.

Онтологией называют явное описание множества объектов и связей между ними, т.е. структурированный словарь. Иными словами, онтология определяет множество сущностей, описывающих и представляющих ПрО, и логические выражения соотношений терминов друг с другом. Такое описание выглядит как четвёрка вида

O=(E, D, R, P), где

E - множество сущностей (термины, классы, объекты, отношения и функции);

D - множество определений сущностей;

R - множество отношений между сущностями;

P - множество правил использования сущностей.

Онтологический анализ представляет собой разделение данных на классы с последующим выделением подклассов и экземпляров данных классов, а также отношений между ними.

Существуют следующие типы онтологий: генеалогия, партономия, атрибутивная структура, таксономия и функциональности. Генеалогией называют онтологию, описывающую отношения типа «отец-сын», партономия рассматривает отношения «имеет-часть», таксономия – «род-вид». Что описывается при помощи онтологий других двух типов, очевидно [7].

Обычно онтологии используют язык, имеющий чёткие различия между классами, свойствами и отношениями. Некоторые инструментальные средства поддерживают автоматизированное использование онтологий, обеспечивая расширенные возможности в отношении интеллектуальных приложений. Кроме того, онтологии позволяют осуществлять описание и структуризацию метаданных [7, 8].

2 Онтологическая модель системы защиты данных

В данной статье при построении системы защиты данных использовалась онтологическая модель, состоящая из трёх компонент: онтологические представления защищаемых данных, актуальные угрозы, средства защиты. Такая модель позволяет разрабатывать системы защиты как больших, так и традиционных данных [9].

В первую очередь строится онтологическая модель защищаемых данных. Защищаемые данные можно условно разделить по предметной направленности, по виду данных и по процессу обработки, в котором они участвуют. В указанных классах выделяются подклассы и, возможно, сущности. Полученные подклассы также можно разделить на подклассы и т.д.

К примеру, при разделении данных по предметной направленности выделяют следующие классы: государственная тайна, коммерческая тайна, банковская тайна, профессиональная тайна, служебная тайна, персональные данные и интеллектуальная собственность.

Аналогичным образом составляется онтологическое представление угроз, актуальных для данной ПрО. При этом угрозы делят по преднамеренности, выделяя в них преднамеренные и непреднамеренные, по воздействию (нарушение физической целостности, несанкционированная модификация, несанкционированное получение и несанкционированное размножение), по дестабилизирующим факторам, по субъекту непосредственной реализации.

Каждая ПрО характеризуется своим набором защищаемых данных и индивидуальным набором актуальных угроз, и предсказать заранее модель, подходящую для конкретной ПрО, невозможно. Так же индивидуален для каждой ПрО и набор экземпляров защищаемых данных. Поэтому в статье предлагается некий шаблон, который для каждой ПрО позволяет детализировать соответствующие классы, вносить конкретные элементы данных и отсекать неиспользуемые в данной ПрО классы или их подклассы.

На основе онтологических представлений защищаемых данных и актуальных угроз в соответствии с политикой безопасности конкретной ПрО строится онтологическая модель средств защиты, реализуемых в разрабатываемой системе. В случае использования в ПрО технологий Big Data, выбираемые средства защиты несколько отличаются от традиционных, к примеру, они должны обладать функциями самообучения.

Таким образом, строится трёхкомпонентная онтологическая модель разрабатываемой системы защиты, на базе которой строится программный код системы защиты данных. Непосредственный переход от онтологической модели к написанию программного кода возможен в силу возможности задания в онтологической модели типов данных и отношений между этими данными, что позволяет описывать в коде конкретные функции и процедуры. Кроме того, онтологическая модель данных является объектно-ориентированной, что позволяет создать на её основе объектно-ориентированный код.

3 Специфика описания метаданных и языки их описания

При анализе данных о ПрО, и больших данных в том числе, выделяются метаданные, в соответствии с которыми данные группируются и делятся на классы. В случае повреждения метаданных (например, преднамеренное искажение форматов данных), собранные данные о ПрО вновь превращаются в хаотичный набор, который непригоден для обработки без повторного выделения метаданных. Поэтому защита метаданных является основным элементом процесса защиты данных. Наиболее часто метаданные представляются в форме онтологии. В свою очередь, метаданные также могут быть разделены на отдельные группы. Для описания метаданных обычно применяют структурированные языки: XML, OWL, RDF, RDFS и другие [1].

При использовании языка XML система описывается в виде тегов и их атрибутов. Такая организация позволяет создать модель системы любой степени сложности при одном ограничении: корневой элемент описываемой структуры должен быть только один.

Наиболее часто для описания онтологий применяется язык OWL (Ontology Web Language). Онтология, применяемая в OWL, может включать описания классов, свойств и методов. При этом формальная семантика OWL определяет способы получения её логических последствий – фактов, не присутствующих явным образом в онтологии, но вызванных семантикой. Эта возможность может базироваться на одном документе или множестве распределённых документов, объединённых с использованием определённых механизмов языка OWL.

Поскольку данные, для защиты которых разрабатывается система, имеют определённую значимость для ПрО, защите метаданных (включая используемые для построения трёхкомпонентной модели) также требуется уделить внимание.

Рассмотренные языки описания метаданных являются родственными по отношению друг к другу, поскольку представляют собой язык XML и его модификации. Следовательно, методы защиты, применимые для языка XML, работают и в отношении остальных языков описания метаданных [10-12].

4 Применение методов обфускации для защиты метаданных

Наиболее часто для документов XML применяются методы обфускации («запутывания») кода. На рисунке 1 представлена иерархическая схема классификации метаданных на примере разных типов ресурсов, на рисунке 2 - XML-представление примера классификации метаданных для разных типов ресурсов.

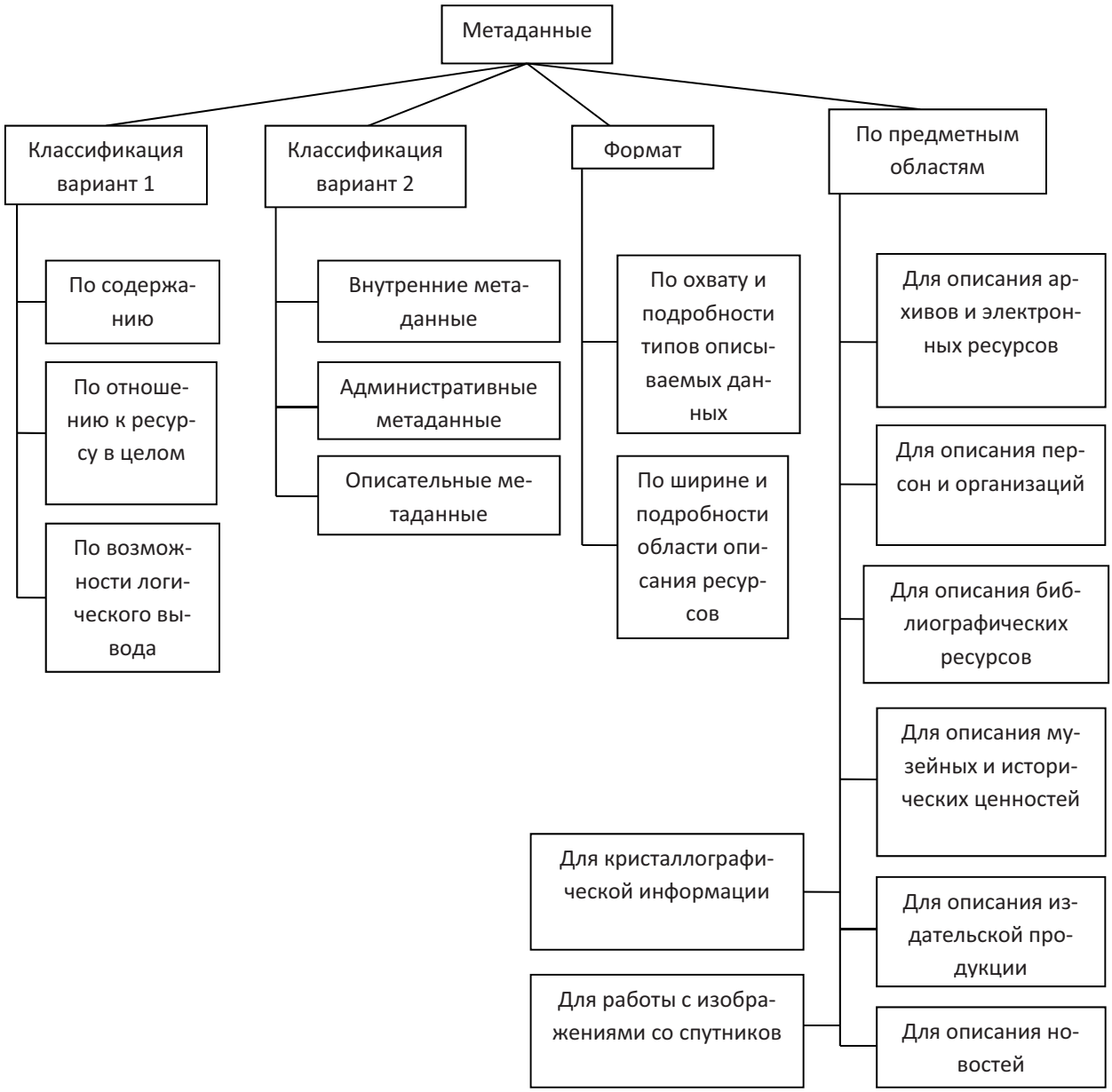

Рисунок 1 - Иерархическая схема классификации метаданных на примере разных типов ресурсов

Для «запутывания» XML- кода (рисунок 2) часто используется следующий метод: выбирается случайный фрагмент XML- кода, после чего в исходном тексте выбирается способ запутывания с сохранением исходной логической последовательности, после чего выбранный кусок кода заменяется полученным [13]. При этом следует выбирать новый способ представления таким образом, чтобы он не мог выражать почти ничего, кроме исходной логики. Приведём простейший пример такой операции. На рисунке 3 приведена схема выделенного фрагмента XML- кода из рисунка 1. Далее изменим структуру фрагмента XML- кода по схеме, представленной на рисунке 4, и сам XML- код на рисунке 5.

```
<?xml version="1.0" encoding="Windows-1251" ?>
<Метаданные>
  <Классификация_вариант_1>
    <По_содержанию/>
    <По_отношению_к_ресурсу_в_целом/>
    <По_возможности_логического_вывода/>
   </Классификация_вариант_1>
  <Классификация_вариант_2>
   <Внутренние_метаданные/>
   <Административные_метаданные/>
   <Описательные_метаданные/>
  </Классификация_вариант_2>
  <Формат>
   <По_охвату_и_подробности_типов_описываемых_ресурсов/>
   <По_ширине_и_подробности_области_описания_ресурсов/>
  </Формат>
  <По_предметным_областям>
   <Для_описания_архивов_и_электронных_ресурсов/>
   <Для_описания_персон_и_организаций/>
   <Для_описания_библиографических_ресурсов/>
   <Для_описания_музейных_и_исторических_ценностей/>
   <Для_описания_издательской_продукции/>
   <Для_кристаллографической_информации/>
   <Для_работы_с_изображениями_со_спутников/>
   <Для_описания_новостей/>
  </По_предметным_областям>
</Метаданные>
```

Рисунок 2 - XML-представление примера классификации метаданных для разных типов ресурсов

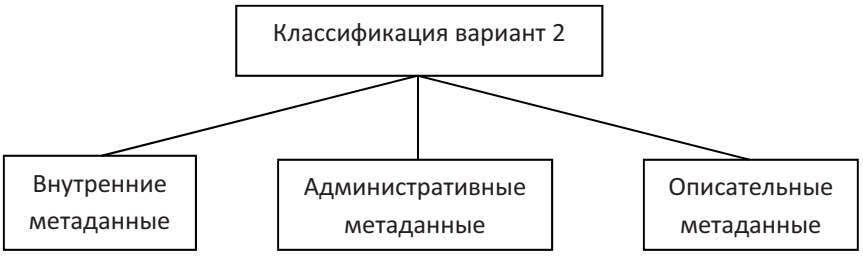

Рисунок 3 - Структура выделенного фрагмента XML-кода

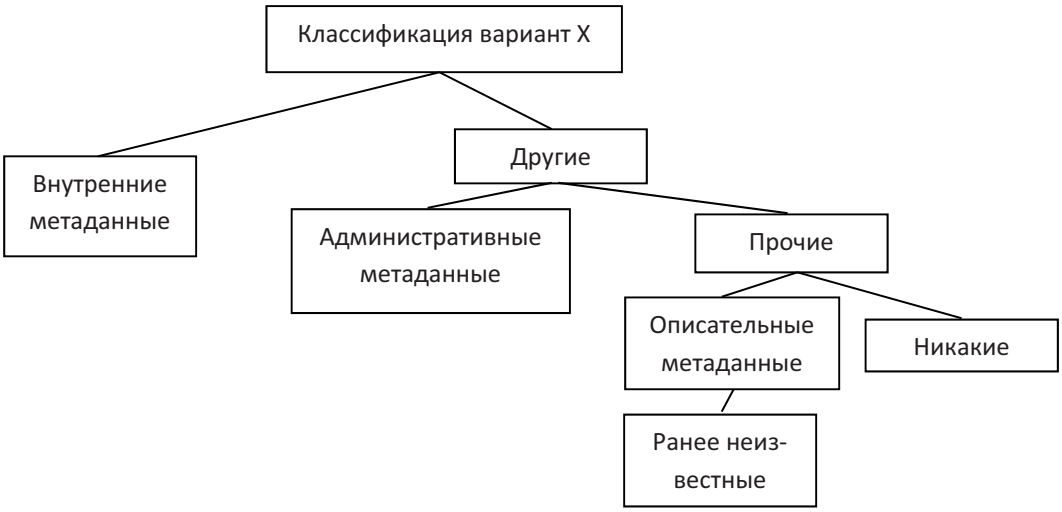

Рисунок 4 - Структура изменённого фрагмента кода, где «вариант X», «Другие», «Прочие» и «Никакие» добавлены для запутывания XML- кода

```
<Классификация_вариант_X>
 <Внутренние_метаданные/>
 <Другие>
  <Административные_метаданные/>
  <Прочие>
   <Описательные_метаданные>
    <Ранее_неизвестные/>
  </Описательные_метаданные>
   <Никакие/>
  </Прочие>
 </Другие>
</Классификация_вариант_X>
<X value=”2”>
 <XXX/>
</X>
```

Рисунок 5 - Изменённый фрагмент кода

Как видно из рисунков 2 и 5, в исходном XML-документе выбрана часть, описывающая разделение метаданных на внутренние, административные и описательные, и изменён этот участок кода, как показано на рисунке 6.

```
<?xml version="1.0" encoding="Windows-1251" ?>
<Метаданные>
  <Классификация_вариант_1>
    <По_содержанию/>
    <По_отношению_к_ресурсу_в_целом/>
    <По_возможности_логического_вывода/>
   </Классификация_вариант_1>
<Классификация_вариант_X>
   <Внутренние_метаданные/>
   <Другие>
    <Административные_метаданные/>
     <Прочие>
      <Описательные_метаданные>
      <Ранее_неизвестные/>
     </Описательные_метаданные>
     <Никакие/>
   </Прочие>
</Другие>
  </Классификация_вариант_X>
  <X value=”2”>
   <XXX/>
  </X>
  <Формат>
   <По_охвату_и_подробности_типов_описываемых_ресурсов/>
   <По_ширине_и_подробности_области_описания_ресурсов/>
  </Формат>
  <По_предметным_областям>
   <Для_описания_архивов_и_электронных_ресурсов/>
   <Для_описания_персон_и_организаций/>
   <Для_описания_библиографических_ресурсов/>
   <Для_описания_музейных_и_исторических_ценностей/>
   <Для_описания_издательской_продукции/>
   <Для_кристаллографической_информации/>
   <Для_работы_с_изображениями_со_спутников/>
   <Для_описания_новостей/>
  </По_предметным_областям>
</Метаданные>
```

Рисунок 6 - Результат обфускации

Из рисунка 6 видно, что полученный в результате документ стал менее компактным, чем исходный, и менее понятным. Однако, при желании выделить исходный код всё же можно. Представленный метод обфускации можно доработать, присвоив каждой части кода порядковый номер. Далее используется программный генератор случайных чисел, чтобы получить порядковый номер трансформируемого участка. Осуществляется замена выбранного куска кода на изменённый. Из списка возможных значений исключается порядковый номер изме-

нённой части и вновь запускается генератор случайных чисел, изменяется часть кода и так до тех пор, пока все участки кода не будут трансформированы. Блок-схема соответствующего алгоритма приведена на рисунке 7.

Рисунок 7 - Блок-схема изменённого алгоритма обфускации

Может быть выбран и менее распространённый метод обфускации, например, применение своеобразной «матрёшки», т.е. вставки не имеющих значения в контексте ПрО строк через строку кода с начала и конца документа к середине. Полученный текст с включением незначащих слов (например, Desyat_negrityat и т.д. - балласт) приведён на рисунке 8. Как видно из рисунков 2, 6 и 8, использование обфускации действительно затрудняет чтение и, следовательно, понимание кода, но имеет определённые недостатки, среди которых увеличение размера XML-кода и возможность при желании всё же определить исходный код [14].

5 Стеганография XML-кода

XML-документы часто применяются в связи со стеганографией, а именно – методом изменения порядка следования атрибутов в файлах с разметкой и рядом других алгоритмов. При этом XML-документ используется в качестве стегоконтейнера [15]. К примеру, скрываемая информация может быть встроена в зарезервированные поля, предназначенные для метаданных. Но в данном случае метаданные, записанные в формате XML, являются собственно стего. При этом логично воспользоваться методами текстовой стеганографии, когда контейнер представляет собой текстовый файл. Перед записью стего шифруется, а при чтении расшифровывается.

```
<?xml version="1.0" encoding="Windows-1251" ?>
<Метаданные>
<Desyat_negrityat/>
  <Классификация_вариант_1>
  <Odin_poperhnulsya/>
    <По_содержанию/>
   <Devyat_negrityat/>
    <По_отношению_к_ресурсу_в_целом/>
   <Odin_ne_smog_prosnutsya/>
    <По_возможности_логического_вывода/>
    <Vosem_negrityat/>
   </Классификация_вариант_1>
  <Odin_ne_vozvratilsya/>
  <Классификация_вариант_2>
  <Sem_negrityat/>
   <Внутренние_метаданные/>
<Zarubil_odin_sebya/>
   <Административные_метаданные/>
<Shest_negrityat/>
   <Описательные_метаданные/>
<Odnogo_uzhalil_shmel>
  </Классификация_вариант_2>
<Pyat_negrityat/>
  <Формат>
   <Zasudili_odnogo/>
   <По_охвату_и_подробности_типов_описываемых_ресурсов/>
  <Chetyre_negritenka/>
   <По_ширине_и_подробности_области_описания_ресурсов>
   <Poshli_kupatsya_v_more/>
  <Odin_popalsya_na_primanku/>
  <Ih_ostalos_troe/>
  <Troe_negrityat/>
  <V_zverince_okazalis/>
  <Odnogo_shvatil_medved/>
  <I_vdvoem_ostalis/>
  <Dvoe_negrityat/>
<Legli_na_solncepeke/>
<Odin_sgorel/>
<I_vot_odin/>
<Neschastnyi_odinokii/>
<Poslednii_negritenok/>
<Poglyadel_ustalo/>
<On_poshel_povesilsya/>
<I_nikogo_ne_stalo/>
  </По_ширине_и_подробности_описания_ресурсов>
  <Ostalos_ih_chetyre/>
  </Формат>
<Sudeistvo_uchinili/>
  <По_предметным_областям>
<Ih_ostalos_pyat/>
   <Для_описания_архивов_и_электронных_ресурсов/>
  <Poshli_na_paseku_gulyat/>
   <Для_описания_персон_и_организаций/>
   <I_ostalos_shest_ih/>
   <Для_описания_библиографических_ресурсов/>
  <Drova_rubili_vmeste/>
   <Для_описания_музейных_и_исторических_ценностей/>
   <Ostalis_vsemerom/>
   <Для_описания_издательской_продукции/>
  <V_Devon_ushli_potom/>
   <Для_кристаллографической_информации/>
  <Ih_ostalos_vosem/>
   <Для_работы_с_изображениями_со_спутников/>
  <Poev_klevali_nosom/>
   <Для_описания_новостей/>
  <Ih_ostalos_devyat/>
  </По_предметным_областям>
<Otpravilis_obedat/>
</Метаданные>
```

Рисунок 8 - Обфускация с использованием «матрёшки»

Часто используются методы изменения порядка следования маркеров конца строки, хвостовых пробелов, знаков одинакового начертания и двоичных нулей. Кроме того, зашифрованное стего может быть встроено в другой XML-файл. Хотя именно благодаря тому, что XML-документ легко использовать в качестве контейнера, этот метод применять весьма рискованно [16].

Заключение

Метаданные в виде онтологии записываются на XML и родственных ему языках. Эти данные требуют защиты, для чего предлагается использовать трёхкомпонентную модель при проектировании комплексных средств защиты информационных объектов. Поскольку эта модель также представляет собой онтологию, в статье рассмотрены применяемые методы защиты XML-структурированных данных, а именно обфускация и текстовая стеганография.

Список источников

[1] ***Tauberer, J.*** What is RDF and what is it good for? Last revised January 2008. – https://github.com/JoshData/rdfabout/blob/gh-pages/intro-to-rdf.md.

[2] ***Половикова, О.Н.*** Анализ XML-подхода для описания метаданных и онтологий в Semantic Web. 2015. – http://izvestia.asu.ru/media/files/issue/9/articles/ru/119-123.pdf.

[3] ***Лэм, Чак***. Hadoop в действии. — М.: ДМК Пресс, 2012. — 424 с.

[4] ***Когаловский, М.Р.*** Метаданные в компьютерных системах / М.Р. Когаловский // Программирование. - 2013. Т. 39, № 4. С. 28-46. - http://www.ipr-ras.ru/articles/kogalov13-03.pdf.

[5] ***Большаков, О.А.*** Метаданные и прикладное программирование / О.А. Большаков // Школа программирования Coding Craft – 2011. - https://codingcraft.ru/metadata.php.

[6] ***Воробьёв В.И.*** Проектирование систем защиты с применением онтологий / В.И. Воробьёв, Т.В. Монахова // Труды СПИИРАН. - 2004. Т.2, №2. – С.212-215.

[7] ***Гаврилова, Т.А.*** Онтологический подход к управлению знаниями при разработке корпоративных информационных систем / Т.А. Гаврилова // Новости искусственного интеллекта. - 2003. №2. - с.24-30.

[8] ***Боргест, Н.М***. Онтологии: современное состояние, краткий обзор / Н.М. Боргест, М.Д. Коровин // Онтология проектирования. 2013. №2(8). - С.49-55. - http://www.ontology-of-designing.ru/article/2013_2%288%29/7_Borgest.pdf

[9] Информационная безопасность социально-экономических систем: монография / Апатова Н.В., Акинина Л.Н., Бойченко О.В., Герасимова С.В. и др. Под ред. д.т.н. профессора О.В. Бойченко. – Симферополь: ИП Зуева Т.В., 2017. - 348 с.

[10] ***Монахова Т.В.*** Онтологическая модель описания экспериментальных данных / Т.В. Монахова // Труды СПИИРАН. – 2013. №1(24). – С.303-312.

[11] ***Монахова, Т.В.*** Онтологическая модель системы защиты данных / Т.В. Монахова // Сборник трудов секции «Информационная безопасность» Всероссийской конференции по вопросам баллистического обеспечения. - Королёв: 4 ЦНИИ МО РФ. – 2014.

[12] ***Монахова, Т.В.*** Защита XML-структурированных данных / Т.В. Монахова // Труды СПИИРАН. – 2013. №2(25). – С.182-189.

[13] ***Ализар, А.*** Математическая обфускация: криптографическая защита программного кода. 2014. - https://xakep.ru/2014/08/15/crypto-obfuscation/.

[14] ***Никольская, К.Ю.*** Обфускация и методы защиты программных продуктов / К.Ю. Никольская, А.Д. Хлестаков // Вестник УрФО. Безопасность в информационной сфере 2015; 2(16) с.7-10. - http://info-secur.ru/is_16/Nikolskaya.pdf.

[15] ***Текин, В.*** Текстовая стеганография / В. Текин // Мир ПК. - 2004. №11 - http://www.osp.ru/pcworld/2004/11/169154.

[16] ***Барильник, С.С.*** Применение алгоритмов стеганографии в современных информационных системах / С.С. Барильник, И.В. Минин, О.В. Минин // Материалы III Международной научно-практической конференции «Актуальные проблемы безопасности информационных технологий. - Красноярск. 2009. - https://window.edu.ru/resource/414/67414/files/AProBIT-2009.pdf.

PROTECTION OF METADATA IN XML FORMAT

V.I. Vorobjev[1], T.V. Monakhova [2]

[1]*Federal State Institution of Science St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russia*
vvi@iias.spb.su

[2] *Central Research and Development Institute of the Russian Defense Ministry, Korolev, Russia*
monakhova_t81@mail.ru

Abstract

The article discusses the protection of metadata presented in XML and related languages. At the same time data on the subject area are ordered using ontological methods. A three-component ontological model of the data protection system is proposed on the basis of ontological representations of domain data in terms of protected data and potential threats. A data classification template has been developed that allows you to detail the corresponding classes, enter specific data elements, and cut off unused classes or their subclasses. Based on ontological representations of protected data and actual threats, in accordance with the security policy, an ontological model of the protection tools implemented in the developed system is built. The languages for describing metadata are discussed. It is suggested to use the methods of obfuscation and text steganography to protect the XML document. An algorithm of the modified obfuscation method with random sampling of a part of the code is proposed. A block diagram of an algorithm suitable for the design of metadata protection facilities is constructed. Explanation of the decision to not use the XML document as a stegocontainer is made. Recommendations on the method and sequence of application of ontological metadata protection methods were developed. A description is also given of the specifics of the use of methods of obfuscation and text steganography.

Keywords: *metadata, ontology model, data protection system, XML, obfuscation, steganography, big data.*

Citation: *Vorobjev VI, Monakhova TV*. Protection of metadata in XML format [In Russian]. *Ontology of designing*. 2018; 8(2): 253-264. DOI: 10.18287/2223-9537-2018-8-2-253-264.

References

[1] ***Tauberer J.*** What is RDF and what is it good for? Last revised January 2008. – https://github.com/JoshData/rdfabout/blob/gh-pages/intro-to-rdf.md.

[2] ***Polovikova ON***. Analyze of XML approach for metadata and ontology description in Semantic Web [In Russian]. - 2015. – http://izvestia.asu.ru/media/files/issue/9/articles/ru/119-123.pdf.

[3] ***Chuck Lam.*** Hadoop in Action. — Manning Publications Co., Stanford. 2010. — 312 p.

[4] ***Kogalovskiy MR.*** [Metadata in computer systems] Metadannyye v komp'yuternykh sistemakh [In Russian]. Programmirovaniye. 2013; 39(4): 28-46.

[5] ***Bol'shakov O.*** Metadata and application programming [In Russian]. 2011. – http://codingcraft.ru/metadata.php.

[6] ***Vorobjev VI, Monakhova TV.*** Protection systems design with ontologies [In Russian]. Proceedings of SPIIRAS. – 2004; 2(2): 212-215.

[7] ***Gavrilova TA.*** Ontological approach to knowledge management in the development of corporate information systems [In Russian]. - J. News of Artificial Intelligence. - 2003; 2: 24-30.

[8] ***Borgest NM, Korovin MD.*** Ontologies: current state, short review [In Russian]. Ontology of Designing. – 2013; 2(8): 49-55. - http://www.ontology-of-designing.ru/article/2013_2%288%29/7_Borgest.pdf.

[9] Information security of socio-economic systems: monograph [Metamodel' zashchity metadannykh. Informatsionnaya bezopasnost' sotsi-al'no-ekonomicheskikh sistem: monografiya] [In Russian]. Apatova NV, Akinina LN, Boychenko OV, Gerasimova SV and etc. Ed. Doctor of technical sciences. professors O.V. Boychenko. – Simferopol': IP Zuyeva TV, 2017. - 348 p.

[10] ***Monakhova TV.*** Ontological model experimental data description [In Russian]. Proceedings of SPIIRAS. – 2013; 1(24): 303-312.

[11] ***Monakhova TV.*** Data protection system ontological model [In Russian]. Papers of «Information security» section of All-Russian conference on ballistic support. Koroljev: 4 CNII MO RF. – 2014.

[12] ***Monakhova TV.*** XML-structured data protection [In Russian]. Proceedings of SPIIRAS. – 2013; 2(25): 182-189.

[13] ***Alizar A.*** Math obfuscation: cryptography program code protection [In Russian]. 2014. - https://xakep.ru/2014/08/15/crypyo-obfuscation/.

[14] ***Nikolskaya KU, Khlestakov AD.*** Obfuscation and program products protection methods [In Russian]. UrFO messenger. Protection in informatics sphere 2015; 2(16) - https://info-secur.ru>is_16/Nikolskaya.pdf.
[15] ***Tekin V.*** Text steganography [In Russian]. 2004. - https://www.osp.ru/pcworld/2004/11/169154.
[16] ***Barilnik SS, Minin IV, Minin OV.*** An application of staganography algorithms in modern information systems [In Russian]. III International scientific and technical conference «Information technologies protection actual problems» papers, Krasnoyarsk. 2009. - https://window.edu.ru/resource/414/67414/files/AProBIT-2009.pdf.

Сведения об авторах

Воробьев Владимир Иванович, 1942 г. рождения. Окончил Ленинградский гидрометеорологический институт в 1965 г., д.т.н. (1994), профессор, Главный научный сотрудник Санкт-Петербургского института информатики и автоматизации Российской академии наук. В списке научных трудов более 115 работ в области математического моделирования и информатики.

Vladimir Ivanovich Vorobjev (b. 1942) graduated from Hydro-meteorological Institute (St-Petersburg) in 1965, PhD (1994), Professor, Chief Researcher Laboratory of Computing & Information Systems and Programming Technologies of Federal State Institution of Science St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences.

Монахова Татьяна Вячеславовна, 1981 г. рождения. Окончила Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина) в 2004 г., научный сотрудник 4-го Центрального Научно-исследовательского института Минобороны РФ (Королев). В списке научных трудов 7 работ в области моделирования систем защиты данных.

Tatjana Vjacteslavovna Monakhova (b. 1981) graduated St.-Petersburg State Electrotechnical University in 2004, Researcher 4th Central Research and Development Institute of the Russian Defense Ministry, Korolev. She is co-author of 7 publications in the field modeling of systems of data protection.