УДК 004.8 004:658.562

ОЦЕНКА ЗАЩИЩЕННОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ ИНТЕЛЛЕКТУАЛЬНЫХ МЕТОДОВ

Кабак И.С., Суханова Н.В., Гейдаров Э.

МГТУ «СТАНКИН»

ikabak@mail.ru

Защищенность программного обеспечения – это безопасность функционирования комплекса программ и возможный риск от его применения. Защищенность является показателем качества программного обеспечения, характеризующим его функциональные возможности.

В данной статье рассмотрен метод оценки защищенности программного обеспечения, в основе которого лежит применение искусственного интеллекта, в частности, нечеткой логики и искусственных нейронных сетей.

Ключевые слова: качество программ, защищенность программ, нечеткая логика, искусственные нейронные сети.

SECURITY ASSESSMENT SOFTWARE WITH INTELLIGENT METHODS

Kabak I.S., Sukhanova N.V., Geidarov I.

MSTU «STANKIN»

Security of the software is safety of functioning of a complex of programs and possible risk from its application, Security is an indicator of quality of the software, characterizing its functionality.

In given article the method of an estimation of security of the software in which basis artificial intelligence application, in particular, the indistinct logic and artificial neural networks lies is offered.

Keywords: quality of programs, security of programs, the indistinct logic, artificial neural networks.

Под защищенностью понимают способность программного обеспечения (ПО) безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью [1]. Количественная оценка защищенности, одного из параметров качества, характеризуется вероятностью работы ПО без отказа в течение определенного периода времени. Ранее оценке защищенности был посвящен ряд работ, в частности, [2]. К сожалению, описанные методы и модели прогнозирования защищенности ПО не в полной мере пригодны для практического применения из-за излишней трудоемкости.

Отметим, что защищенность ПО является характеристикой его исполняемого кода. Две функционально идентичные программы, написанные на разных языках или подготовленные для разных типов машин (или для одной и той же машины, но с использованием разных компиляторов), с точки зрения защищенности следует считать разными.

Под уязвимостью понимается программный код, выполнение которого может при определенных условиях нарушить безопасность обрабатываемой информации. При этом наличие уязвимости может быть обусловлено как непреднамеренными ошибками разработчика, так и умышленными действиями.

Известны следующие методы оценки защищенности ПО [3]:

– сквозная инспекция исходных текстов;

– аудит исходных текстов программного обеспечения;

– динамическое тестирование нагрузкой.

В настоящее время наиболее часто применяется сквозная инспекция исходных текстов, которую проводит эксперт-аудитор. Аудитор строит ментальную модель работы программы и старается определить, какие внешние воздействия могут нарушить ход ее работы. При этом аудитор существенно упрощает систему, обращая внимание на структуру и поведение ПО в контексте обработки внешних данных. В общем виде аудитор выполняет следующую последовательность технологических операций:

– производит декомпозицию ПО, выделяя множество программных субъектов и связей между ними;

– определяет все интерфейсы ПО, через которые в систему попадают внешние входные данные;

– определяет логические тракты прохождения входной информации и оценивает зависимость от нее алгоритма работы ПО;

– восстанавливает модель функционирования ПО в части обработки входных данных;

– путем экспертной оценки определяет множество критических программных субъектов, нештатное функционирование которых может повлечь реализацию угрозы, а также условия, приводящие к такому функционированию.

Аудит безопасности

Аудит представляет собой процесс ручного или автоматического сканирования (при помощи специализированного программного средства) исходных текстов ПО с целью выявления потенциально опасных синтаксических конструкций. В частности, существуют конструктивные изъяны, которые связаны с нарушениями общих принципов безопасного программирования, а именно:

– некорректным использованием синтаксических конструкций языка программирования;

– несоблюдением интерфейсных соглашений и ограничений при использовании библиотечных или системных вызовов, а также при взаимодействии с операционной системой и аппаратной платформой;

– необоснованным доверием к входным данным и среде функционирования ПО.

При анализе применяются специализированные программы, ориентированные на выявление потенциально опасных синтаксических конструкций в исходных текстах ПО. Выявленные конструкции заносятся в отчет, который затем совместно с исходными текстами подвергается ручному анализу аудитором.

Сама проблема защищенности ПО имеет по крайней мере два аспекта: обеспечение и оценка (измерение) защищенности [3]. Обеспечению защищенности посвящено достаточное количество работ.

В патенте [4] приводится способ оценки защищенности программных систем на основе двумерной матрицы оценки защищенности программной системы. Первая размерность матрицы соответствует множеству параметров защищенности системы, а вторая размерность – численная оценка этого параметра. Формирование матрицы осуществляется в результате опроса экспертов и математической обработки этих результатов.

Предлагаемый способ оценки защищенности близок к описанному в [4], но имеет существенные отличия:

1. Опрос производится не среди экспертов, а среди пользователей системы в автоматизированном режиме.
2. При опросе используются специально подготовленные вопросы.
3. Данные могут быть представлены как в виде четких переменных, так и в виде нечетких значений.
4. Используются интеллектуальный способ обработки и получение результатов по матрице.

Защищенность ПО определяется степенью (полнотой) выполнения требований, предъявляемых к оценке защищенности ПО. В основу оценки защищенности положим данные опроса, представленные в виде матрицы знаний, заполняемой пользователями или экспертами. Заполнение матрицы знаний осуществляется на основе лингвистических (интервальных) оценок отдельных элементов. Особенностью частных показателей является то, что все они имеют качественный характер, т.е. не имеют точного количественного измерения. Поэтому при оценке одного и того же показателя несколькими пользователями он может существенно различаться. Кроме того, пользователь не всегда способен точно оценить каждый показатель, хотя интуитивно его ощущает.

Применение нечетких значений существенно упрощает для пользователя процедуру опроса. Показатели определяются как лингвистические переменные, заданные на едином универсальном множестве $U = \backslash u, u \backslash$, которым является шкала оценок.

Такой подход дает возможность использовать в качестве показателя оценки защищенности ПО аддитивный показатель Q , определяющий количество выполненных частных показателей:

$$Q = \frac{\sum_{k=1}^{5}\sum_{j=1}^{4}\sum_{i=1}^{7} z_{kji}}{140},$$

где $z_{kji} = \begin{cases} 1, \text{если } q_{kji} > q_{kji}^{T}, \\ 0, \text{если } q_{kji} < q_{kji}^{T}, \end{cases}$ q_{kji} и q_{kji}^{T}

$$Q = \sum_{i=1}^{m} \omega_i q_i .$$

Полученные в результате опроса данные являются весьма субъективными. Для повышения объективности данных в системе предусмотрено введение весовых коэффициентов объективности. Каждому пользователю соответствует один столбец таблицы. Первоначально все столбцы имеют одинаковые весовые коэффициенты. При функционировании системы будем оценивать степень соответствия оценок отдельных пользователей усредненным значениям (по всем пользователям).

Критерии оценки защищенности программной системы также выбирались с известной степенью субъективности. В результате получим также весовые коэффициенты для параметров оценки (опроса).

Для весовых коэффициентов пользователей и для параметров оценки обязательным является условие (суммирование ведется по всему множеству коэффициентов):

$$\Sigma \omega_i = 1 .$$

Определение принадлежности ОЗ ПО к конкретному классу проводится на основе функции принадлежности, заданной нечеткими термами классов. Результатом оценки будет вероятность принадлежности ОЗ ПО к конкретному классу.

Наиболее существенным отличием, на взгляд авторов, является использование методов искусственного интеллекта для обработки результатов опроса, сведенных в матрицу.

зификации информации. Для фазификации информации используется искусственная нейронная сеть особого вида, описанная в [5–13].

Обработка матрицы и получение окончательной оценки также производится с использованием искусственных нейронных сетей особого вида. Нечеткие оценки ответов подают на входы обученной искусственной нейронной сети. На выходе искусственной нейронной сети получают итоговую оценку защищенности ПО (см. рис. 1).

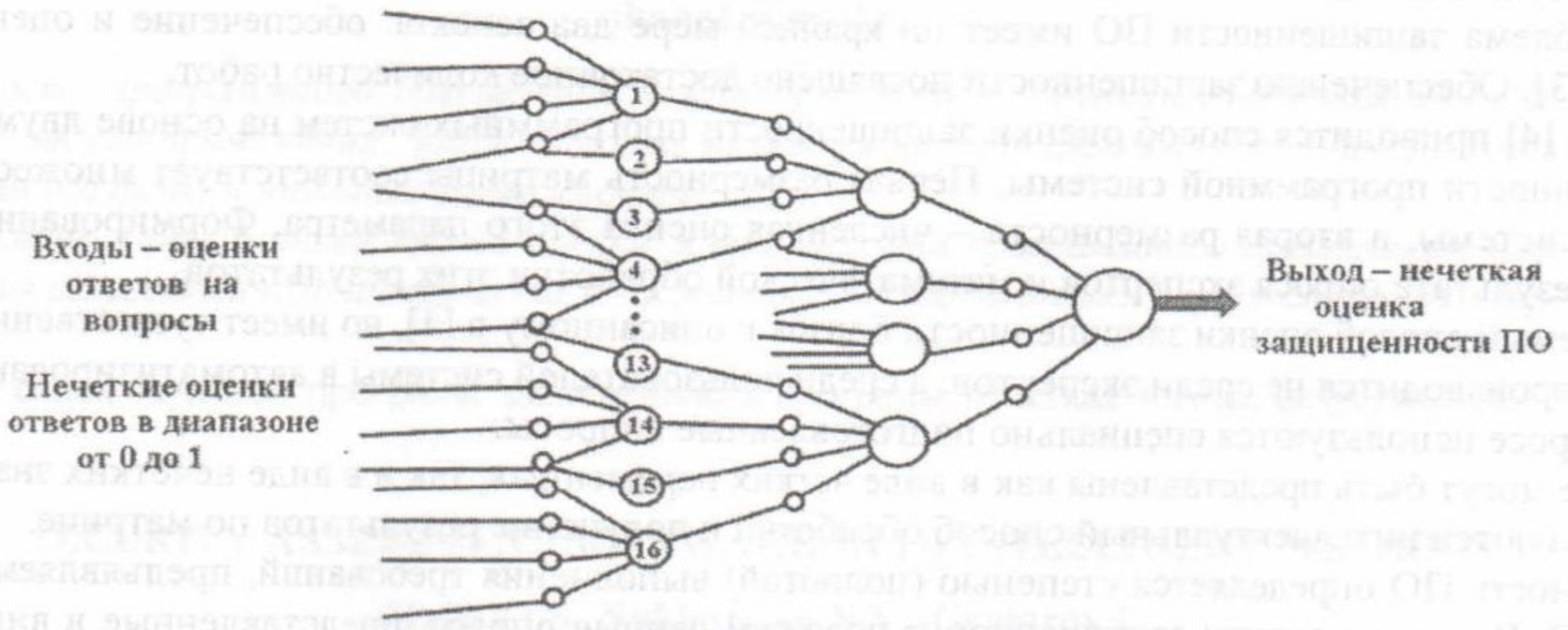

Рис. 1. Схема обработки результатов опроса

На рис. 2 представлена структура системы для оценки защищенности ПО.

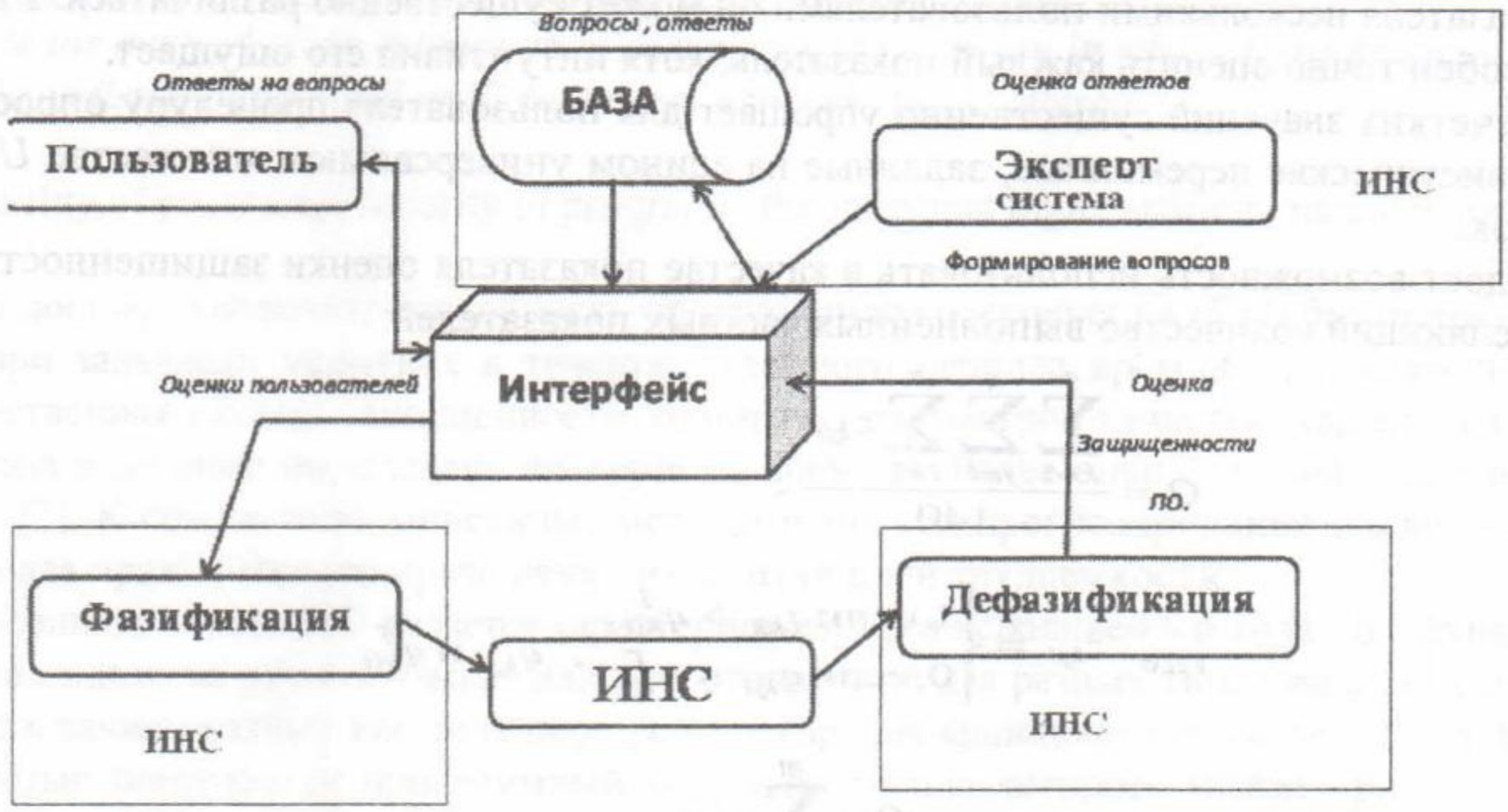

Рис. 2. Структура системы для оценки защищенности ПО

В работе системы многократно применяется один и тот же механизм – искусственные нейронные сети, обозначенные на рис. 2 ИНС. Они используются для фазификации четких параметров (приведение четких параметров к нечеткому виду), для дефазификации (приведение нечетких параметров к четкому виду), для работы базы знаний и модернизации системы – выявления наиболее значимых параметров оценки программной системы и оценки влияния мнения каждого пользователя на общий результат.

Выводы

1. Предлагаемый в статье подход позволяет создавать системы количественной оценки защищенности программного обеспечения.

2. Созданные системы оценки защищенности являются адаптивными, они приспосабливаются к конкретным пользователям и различным типам оцениваемого программного обеспечения.

3. Системы оценки обладают свойством самосовершенствования. В процессе работы качество оценки существенно улучшается.

4. В процессе проведения опроса возможно расширение-сокращение списка вопросов для пользователей и добавление-исключение пользователей, т.е. изменение количества пользователей.

5. Процедура оценки защищенности состоит из нескольких итераций. Каждая итерация используется для обучения ИНС на следующем шаге.

Библиография

1. ГОСТ 19781-90 Программное обеспечение.
2. ГОСТ Р ИСО/МЭК 9126-93 Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению.
3. Липаев В.В. Качество программных средств: методические рекомендации. – М., 2002. – 400 с.
4. Патент № 2004134589/09, 28.04.2003. Способ оценки совершенности защиты. Опубл. 27.09.2007 / К.Р. Бодуэн, К.Р. Элиот.
5. Патент на ПМ № 66831 приоритет 02.04.2007; Федеральная служба по интеллектуальной собственности. 2007 г. Нейронная сеть / И.С. Кабак, Н.В. Суханова.
6. Патент на ПМ №.72084 приоритет 03.12.2007; Федеральная служба по интеллектуальной собственности. 2007 г. Доменная нейронная сеть / И.С. Кабак, Н.В. Суханова.
7. Патент на ПМ №. 75247 приоритет 26.12.2008; Федеральная служба по интеллектуальной собственности. 2009 г. Модульная вычислительная система / И.С. Кабак, Н.В. Суханова.
8. Патент на изобретение № 2398281 приоритет 07.11.2008; Федеральная служба по интеллектуальной собственности. 2010 г. Многослойная модульная вычислительная система / Ю.М. Соломенцев, С.А. Шептунов, И.С. Кабак, Н.В. Суханова.
9. Патент на изобретение № 2417442 приоритет 19.12.2008; Федеральная служба по интеллектуальной собственности. 2011 г. Способ построения систем нечеткой логики и устройство для его реализации / Ю.М. Соломенцев, С.А. Шептунов, И.С. Кабак, Н.В. Суханова.
10. Степанов С.Ю., Кабак И.С. Алгоритм фрагментации больших нейронных сетей и исследование его сходимости // Информационные технологии. – № 7. – 2012. – С. 73–78.
11. Кабак И.С., Суханова Н.В. Моделирование надежности программного обеспечения систем управления автоматизированными технологическими комплексами на базе искусственного интеллекта // Вестник МГТУ «Станкин». – № 1 (19). – 2012. – С. 95–99.
12. Кабак И.С., Суханова Н.В. Технология реализации автоматизированных систем управления на базе больших искусственных нейронных сетей МОДУС-НС // Межотраслевая информационная служба. – 2012.
13. Кабак И.С. Создание больших аппаратно-программных нейронных сетей для систем управления // Авиационная промышленность. – № 4. – 2012.