DOI: 10.24143/2072-9502-2020-1-50-56
УДК 004.056

ПОДХОД К ОЦЕНКЕ ЗАЩИЩЕННОСТИ ВСТРОЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УСЛОВИЯХ НЕЧЕТКОСТИ ВХОДНОЙ ИНФОРМАЦИИ

А. Н. Югансон, Д. А. Заколдаев

Университет ИТМО,
Санкт-Петербург, Российская Федерация

Вопросы защищенности и безопасности программного обеспечения оказываются второстепенными при проектировании и разработке программных средств в целях скорейшего вывода программного продукта на рынок. В связи с тем, что стоимость устранения дефектов безопасности выше на поздних этапах проектирования, рассмотрена научная задача оценки защищенности программного обеспечения в условиях высокой неопределенности. Приведены функциональные требования к защищенности встроенного программного обеспечения. Предложен новый подход для оценки защищенности программного обеспечения. Предметом исследования является встроенное программное обеспечение, предназначенное для управления различными устройствами и микроконтроллерами. На основе ГОСТ Р 56939-2016 «Защита информации. Разработка безопасного программного обеспечения. Общие требования» разработаны требования (качественные и количественные) защищенности к встроенному программному обеспечению, оценка выполнения которых позволяет определить уровень защищенности встроенного программного обеспечения в целом. Аппарат нечеткой логики был использован для оптимизации процесса оценивания в условиях возможной неопределенности, несогласованности, неполноты и качественного характера исходной информации. Предложенный метод поможет минимизировать экономические риски на этапах эксплуатации и технического обслуживания встроенных систем.

Ключевые слова: встроенное программное обеспечение, нечеткая логика, защищенность программных средств, уязвимость программного обеспечения.

Для цитирования: *Югансон А. Н., Заколдаев Д. А.* Подход к оценке защищенности встроенного программного обеспечения в условиях нечеткости входной информации // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2020. № 1. С. 50–56. DOI: 10.24143/2072-9502-2020-1-50-56.

Введение

Одними из ключевых решений в области автоматизации и управления технологическими процессами в рамках концепции Индустрии 4.0 являются встроенные вычислительные системы [1]. Сама встроенная система представляет собой специализированную информационно-управляющую систему для выполнения определенного набора функций, которая состоит из аппаратных и программных компонентов. Связь между программной частью системы, с одной стороны, и аппаратной, с другой, обуславливает наличие дополнительных ограничений, влияющих на процесс проектирования таких устройств. Встроенное программное обеспечение (ВПО) – это программное обеспечение (ПО), встроенное в аппаратные системы и предназначенное для управления устройствами. Поскольку ПО является основным компонентом встраиваемых систем, очень важно правильно и адекватно протестировать ВПО, особенно для критически важных систем. Из-за сложного системного контекста встроенных программных приложений дефекты в этих системах могут вызывать опасные для жизни ситуации, а задержки могут привести к огромным потерям в бизнесе [2].

Как известно, безопасность является основным требованием для любой информационной системы, и встроенные системы не являются исключением. Достижение данной цели возможно благодаря внедрению SDL (англ. Secure Development Lifecycle – жизненный цикл безопасной разработки) для решения проблем безопасности на всех уровнях [3].

Под защищенностью ВПО понимается вероятность, характеризующая способность программы сохранять заданный уровень пригодности в заданных условиях в течение заданного интервала времени, где в качестве ограничения уровня пригодности рассматриваются дефекты безопасности и уязвимости. Дефект также может быть определен как аномалия продукта [4]. Вероятность является функцией входных данных и использования системы, а также функцией

наличия неисправностей в ПО. Защищенность предназначена для количественной оценки вероятности сбоя ПО. Отказ определяется как прекращение способности функционального блока выполнять требуемую функцию. С другой стороны, отказ может быть определен как событие, в котором система или системный компонент не выполняет требуемую функцию в указанных пределах [5]. Как правило, защищенность объекта – это его защита от внешних источников опасности, в то время как безопасность объекта – это внутреннее свойство объекта не быть источником опасности для окружающей среды.

Во встроенных системах даже небольшие ошибки в ПО могут иметь серьезные последствия, поскольку они могут полностью нарушить взаимодействие системы с физическим миром. Стоимость устранения уязвимостей и дефектов безопасности выше на поздних стадиях проектирования.

Современные стандарты разработки промышленного ПО предполагают преобладание проблемы обеспечения защищенности работы ПО над его оценкой. Парадокс заключается в том, что для правильного обеспечения уровня защищенности разрабатываемого ПО необходимо однозначно определить этот уровень. На сегодняшний день в отрасли не принят единый стандарт для оценки защищенности ВПО. Таким образом, оценка защищенности ВПО в настоящее время является актуальной задачей при проектировании встраиваемых киберфизических систем.

Функциональные требования к защищенности встроенного программного обеспечения

Недостатком существующих подходов по оценке защищенности и безопасности ВПО является отсутствие методологической основы для интегрального анализа выполнения качественных и количественных требований к защищенности ВПО. Для повышения эффективности и оптимизации процесса оценивания предлагается использовать теорию нечетких множеств [6].

Большинство показателей ПО связаны с неопределенностью. Таким образом, необходимо агрегировать результаты экспертного опроса в системе нечеткого вывода [7]. Такой подход позволит получить неясные выводы о необходимом уровне защищенности ВПО на основе нечетких условий или предпосылок, которые представляют информацию о текущем состоянии объекта.

Рассмотрим оценку защищенности ВПО на этапе квалификационного тестирования. В качестве функциональных требований к защищенности ВПО были использованы результаты исследования [8], систематизирующие сведения о существующих способах разработки безопасного ПО в соответствии с этапами жизненного цикла изделия.

Оценка выполнения качественных требований к защищенности встроенного программного обеспечения

В первую очередь, необходимо разработать систему нечеткого логического вывода для качественных критериев, таких как функциональное тестирование и анализ уязвимостей. Они рассматриваются как входные переменные, тогда как качественная оценка защищенности считается выходной переменной.

В общем случае разработка и применение систем нечеткого вывода включают ряд этапов, реализация которых выполняется на основе положений нечеткой логики:

– определение перечня входных переменных;
– формирование базы правил;
– фаззификация (представление физического значения признака в лингвистическом виде);
– поиск правил в базе знаний;
– свертка простых высказываний в условной части правил и оценка ее истинности;
– формирование заключений;
– дефаззификация (представление получившихся в результате нечетких рассуждений лингвистического значения параметра в количественном виде).

Под базой правил в данном случае понимается формальное представление эмпирических знаний экспертов в форме нечетких продукционных правил. Нечеткое продукционное правило – это выражение вида

$$(i): Q; P; A \Rightarrow B; S, F, N,$$

где (i) – имя нечеткой продукции; Q – сфера применения нечеткой продукции; P – условие применимости ядра нечеткой продукции; $A \Rightarrow B$ – ядро нечеткой продукции, в котором A – условие ядра (или антецедент), B – заключение ядра (или консеквент), $\Rightarrow$ – знак логической секвенции или следования; S – метод или способ определения количественного значения степени истинности

заключения ядра; F – коэффициент определенности или уверенности нечеткой продукции; N – постусловия продукции.

Таким образом, система состоит из следующих лингвистических переменных:

– L_v^1 ATE_{FUN} = функциональное тестирование;

– L_v^2 = AVA_{VAN} = анализ уязвимостей;

– L_v^3 = $TEST_{QA}$ = качественная оценка защищенности на этапе квалификационного тестирования.

При задании входных и выходных лингвистических переменных, которые характеризуют показатели защищенности, необходимо задать функции принадлежности для нечетких множеств, которые характеризуют терм-множества лингвистических переменных. В работе предлагается использование типовых функций трапецеидального типа (табл. 1).

Таблица 1

Терм-множества лингвистических переменных

T_{ATE_FUN}	T_{AVA_VAN}	T_{TEST_QA}
Неудовлетворительно (1, 1, 5, 20)	Неудовлетворительно (1, 1, 5, 20)	Очень низкий (1, 1, 5, 20)
Плохо (5, 15, 30, 40)	Плохо (5, 15, 30, 40)	Низкий (5, 15, 30, 40)
Удовлетворительно (25, 40, 60, 75)	Удовлетворительно (25, 40, 60, 75)	Средний (25, 40, 60, 75)
Хорошо (60, 70, 85, 95)	Хорошо (60, 70, 85, 95)	Высокий (60, 70, 85, 95)
Отлично (80, 95, 100, 100)	Отлично (80, 95, 100, 100)	Очень высокий (80, 95, 100, 100)

После того как нечеткие входные и выходные множества и функции принадлежности созданы, формируется набор нечетких правила «ЕСЛИ–ТО», чтобы отразить отношения между любым возможным отношением входных переменных и выходной переменной. Уровни входных параметров, определенные на предыдущем шаге, являются антецедентами, а выходного параметра – консеквент. Нечеткое причинно-следственное отношение между антецедентом и консеквентом задается в виде нечеткой продукции. Нечеткие продукционные правила для качественной оценки защищенности на этапе квалификационного тестирования приведены в табл. 2.

Таблица 2

База нечетких продукционных правил

Правило	Антецедент	Консеквент
R_1	$((ATE_{FUN} = Н) \wedge (AVA_{VAN} = Н)) \vee ((ATE_{FUN} = П) \wedge (AVA_{VAN} = Н)) \vee$ $\vee ((ATE_{FUN} = Н) \wedge (AVA_{VAN} = П))$	$TEST_{QA} = ОН$
R_2	$((ATE_{FUN} = Н) \wedge (AVA_{VAN} = У)) \vee ((ATE_{FUN} = П) \wedge (AVA_{VAN} = П)) \vee$ $((ATE_{FUN} = П) \wedge (AVA_{VAN} = У)) \vee ((ATE_{FUN} = У) \wedge (AVA_{VAN} = Н)) \vee$ $((ATE_{FUN} = У) \wedge (AVA_{VAN} = П))$	$TEST_{QA} = Н$
R_3	$((ATE_{FUN} = Н) \wedge (AVA_{VAN} = Х)) \vee ((ATE_{FUN} = Н) \wedge (AVA_{VAN} = О)) \vee$ $((ATE_{FUN} = П) \wedge (AVA_{VAN} = Х)) \vee ((ATE_{FUN} = П) \wedge (AVA_{VAN} = О)) \vee$ $((ATE_{FUN} = У) \wedge (AVA_{VAN} = У)) \vee ((ATE_{FUN} = Х) \wedge (AVA_{VAN} = П)) \vee$ $((ATE_{FUN} = Х) \wedge (AVA_{VAN} = Н)) \vee ((ATE_{FUN} = О) \wedge (AVA_{VAN} = П)) \vee$ $((ATE_{FUN} = О) \wedge (AVA_{VAN} = Н))$	$TEST_{QA} = С$
R_4	$((ATE_{FUN} = У) \wedge (AVA_{VAN} = Х)) \vee ((ATE_{FUN} = У) \wedge (AVA_{VAN} = О)) \vee$ $((ATE_{FUN} = Х) \wedge (AVA_{VAN} = У)) \vee ((ATE_{FUN} = Х) \wedge (AVA_{VAN} = Х)) \vee$ $((ATE_{FUN} = О) \wedge (AVA_{VAN} = У))$	$TEST_{QA} = В$
R_5	$((ATE_{FUN} = О) \wedge (AVA_{VAN} = О)) \vee ((ATE_{FUN} = Х) \wedge (AVA_{VAN} = О))$ $\vee ((ATE_{FUN} = О) \wedge (AVA_{VAN} = Х))$	$TEST_{QA} = ОВ$

В предлагаемом методе нечеткие правила определяются с помощью эксперта по предметной области [9].

При качественной оценке защищенности ВПО на этапе квалификационного тестирования в качестве правила вычисления нечеткой импликации применяется классическая нечеткая импликация Л. Заде [10]:

$$\mu_R(x,y) = \max\left\{\min\left[\mu_A(x),\mu_B(y)\right]\left[1-\mu_A(x)\right]\right\}.$$

В результате дефаззификации происходит преобразование нечеткого множества в четкое число. Для дефаззификации выходной переменной используется метод центра площади:

$$\int_{\min}^{\mu}\mu(x)\,dx = \int_{\mu}^{\max}\mu(x)\,dx,$$

где x – переменная, соответствующая выходной лингвистической переменной и принимающая значения от x = min до x = max; min и max – левая и правая точки интервала носителя нечеткого множества; $\mu(x)$ – функция принадлежности нечеткого множества.

Результатом нечеткого моделирования является качественная и количественная оценки защищенности ВПО на заданном этапе жизненного цикла изделия. Данная оценка интерпретируется как оценка возможности нейтрализации угроз безопасности ВПО.

Данный подход к оценке качественных требований к защищенности позволяет снять ограничения на количество входных переменных, которые необходимо учитывать.

Оценка выполнения количественных требований к защищенности встроенного программного обеспечения

К количественным функциональным требованиям к защищенности ВПО на этапе квалификационного тестирования относятся глубина тестирования и тестовое покрытие.

Анализ тестового покрытия (ATE_{COV}) выражается в процентном эквиваленте и оценивает исполнимость написанного программного кода. Покрытие кода учитывается в блоках. Блок – это единица кода, которая представляет собой последовательность инструкций с ровно одной точкой входа и выхода. Если поток управления программы проходит через блок во время тестового прогона, этот блок считается закрытым. Количество раз использования блока не влияет на результат.

Одним из преимуществ оценки тестового покрытия на основе оценки покрытия базовых блоков (конструкций кода) состоит в том, что данная методика может быть также применена к объектному (бинарному) коду, что в случае ВПО является критически важным.

В общем случае тестовое покрытие можно вычислить следующим образом:

$$ATE_{COV} = B_{COV} / B,$$

где B_{COV} – количество проверенных блоков кода; B – общее количество блоков кода в программе.

Анализ глубины тестирования (ATE_{DPT}) выражается в процентном эквиваленте и оценивает как содержание тестовых процедур, с одной стороны, так и уровень завершенности тестирования – с другой.

При проведении тестирования объектного кода ВПО предлагается оценивать глубину покрытия требований следующим образом:

$$ATE_{DPT} = R_{DPT} / R,$$

где R_{DPT} – количество проверенных требований, предъявляемых к программе; R – общее количество требований. Полный перечень требований формируется на начальном этапе жизненного цикла программы – в процессе анализа требований к ПО, – где определяются требования в части разработки безопасного ПО, предъявляемые к разрабатываемому ПО.

Интегральная оценка защищенности встроенного программного обеспечения

Оценка защищенности ВПО – это величина, заданная на множестве функций оценки защищенности ВПО на каждом этапе жизненного цикла программы:

$$S_{sec} = F\left(f_{Req}, f_D, f_I, f_V, f_{Rel}, f_{Sup}\right),$$

где f_{Req} – функция оценки защищенности на этапе анализа требований к архитектуре; f_D – функция оценки защищенности на этапе проектирования архитектуры; f_I – функция оценки защищенности на этапе разработки; f_V – функция оценки защищенности на этапе квалификационного тестирования; f_{Rel} – функция оценки защищенности на этапе инсталляции и приемки; f_{Sup} – функция оценки защищенности на этапе поддержки.

Функцию оценки защищенности ВПО на этапе квалификационного тестирования можно представить следующим образом:

$$f_V = TEST_{QA} \cdot \omega_{QA} + ATE_{COV} \cdot \omega_{COV} + ATE_{DPT} \cdot \omega_{DPT},$$

где $\omega_{QA}, \omega_{COV}, \omega_{DPT}$ – весовые коэффициенты свойств защищенности на этапе квалификационного тестирования, при условии $\omega_{QA} + \omega_{COV} + \omega_{DPT} = 1$ и $\omega_{QA} = \omega_{COV} = \omega_{DPT}$.

Таким образом, f_V интерпретируется как вероятность ВПО сохранять заданный уровень пригодности на этапе квалификационного тестирования, где в качестве ограничения уровня пригодности рассматриваются дефекты безопасности и уязвимости.

Заключение

Встроенное программное обеспечение должно быть спроектировано таким образом, чтобы вероятность его отказа была минимальной. Существующие системы оценки защищенности ПО основаны на бинарной системе, в которой, согласно экспертному опросу, имеется либо безопасное, либо небезопасное состояние. Предложенный в работе подход базируется на использовании аппарата нечеткой логики для получения промежуточных оценок экспертного опроса и оптимизации процесса оценивания в условиях возможной неопределенности, несогласованности, неполноты и качественного характера исходной информации от экспертов.

Преимущества предлагаемого подхода заключаются в том, что, во-первых, он позволяет получать численные оценки защищенности ВПО с учетом качественных и количественных показателей защищенности на различных этапах жизненного цикла программы. Во-вторых, экспертное заключение основано на критериях оценки безопасности ПО, сформированных в ГОСТ Р 56939-2016 «Защита информации. Разработка безопасного программного обеспечения. Общие требования». В-третьих, этот метод позволит учитывать как качество вводимой информации, так и достоверность информации от экспертов. В-четвертых, этот подход имеет большой потенциал и позволит адаптировать его к существующим моделям оценки качества ПО, а также изменить его в соответствии с различными типами ПО.

СПИСОК ЛИТЕРАТУРЫ

1. *Зегжда Д. П., Васильев Ю. С., Полтавцева М. А., Кефели И. Ф., Боровков А. И.* Кибербезопасность прогрессивных производственных технологий в эпоху цифровой трансформации // Вопр. кибербезопасности. 2018. № 2 (26). С. 2–15.

2. *Ebert C., Jones C.* Embedded Software: Facts, Figures, and Future // IEEE Computer. 2009. V. 42. P. 42–52.

3. *Howard M., Lipner S.* The security development lifecycle. Redmond: Microsoft Press, 2006. V. 8. 352 p.

4. *IEEE* Guide for the use of IEEE standard dictionary of measures to produce reliable software. IEEE, New York, IEEE Std. 982.2–1988. URL: https://standards.ieee.org/standard/982_2-1988.html (дата обращения: 20.07.2019).

5. *IEEE* Standard glossary of software engineering terminology. IEEE, New York, IEEE Std. 610.12–1990, pp. 1–84. URL: https://ieeexplore.ieee.org/document/159342 (дата обращения: 20.07.2019).

6. *Рыжов А. П.* Элементы теории нечетких множеств и ее приложений. М.: Диалог-МГУ, 1998. 81 с.

7. *Югансон А. Н., Заколдаев Д. А.* Разработка методики для расчета оценки технологической безопасности программных средств // Вестн. УрФО. Безопасность в информационной сфере. 2017. № 1 (23). С. 20–23.

8. *Барабанов А. В., Марков А. С., Цирлов В. Л.* 28 магических мер разработки безопасного программного обеспечения // Вопр. кибербезопасности. 2015. № 5 (13). С. 2–10.

9. *Югансон А. Н., Заколдаев Д. А., Римша А. С.* Применение теории нечетких множеств для оценки технологической безопасности программных средств // Математические методы в технике и технологиях – ММТТ. 2019. Т. 7. С. 20–24.

10. *Zadeh L. A.* Fuzzy Logic // Computer. 1988. V. 21. N. 4. P. 83–93.

Статья поступила в редакцию 18.12.2019

ИНФОРМАЦИЯ ОБ АВТОРАХ

Югансон Андрей Николаевич – Россия, 197101, Санкт-Петербург; Университет ИТМО; ассистент факультета безопасности информационных технологий; a_yougunson@itmo.ru.

Заколдаев Данил Анатольевич – Россия, 197101, Санкт-Петербург; Университет ИТМО; канд. техн. наук, доцент; декан факультета безопасности информационных технологий; d.zakoldaev@mail.ru.

APPROACH TO ASSESSMENT OF FIRMWARE SECURITY UNDER FUZZY INPUT DATA

A. N. Iuganson, D. A. Zakoldaev

ITMO University,
Saint-Petersburg, Russian Federation

Abstract. The article highlights the issues of security and software security, which turn to be secondary in the design and development of software tools in order to please the speedy launch of the software product on the market. Due to the fact that the cost of eliminating security defects is higher in the late stages of design, the scientific problem of assessing software security under high uncertainty has been considered. The functional requirements for security of the firmware are given. A new approach is proposed for assessing the firmware security. The subject of research is a firmware designed to control various devices and microcontrollers. Based on GOST R 56939-2016 “Information security. Secure software development. General requirements” there have been developed the security requirements (qualitative and quantitative) for the embedded software, the assessment of which allows determining the level of security of the firmware as a whole. The fuzzy logic apparatus was used to optimize the assessment process in conditions of possible uncertainty, inconsistency, incompleteness and qualitative nature of the input data. The proposed method will help minimize the economic risks at the stages of operation and maintenance of embedded systems.

Key words: firmware, fuzzy logic, software security, software vulnerability.

For citation: Iuganson A. N., Zakoldaev D. A. Approach to assessment of firmware security under fuzzy input data. *Vestnik of Astrakhan State Technical University. Series: Management, Computer Science and Informatics*. 2020;1:50-56. (In Russ.) DOI: 10.24143/2072-9502-2020-1-50-56.

REFERENCES

1. Zegzhda D. P., Vasil'ev Iu. S., Poltavtseva M. A., Kefeli I. F., Borovkov A. I. Kiberbezopasnost' progressivnykh proizvodstvennykh tekhnologiĭ v epokhu tsifrovoi transformatsii [Cybersecurity of advanced manufacturing technologies in the era of digital transformation]. *Voprosy kiberbezopasnosti*, 2018, no. 2 (26), pp. 2-15.

2. Ebert C., Jones C. Embedded Software: Facts, Figures, and Future. *IEEE Computer*, 2009, vol. 42, pp. 42-52.

3. Howard M., Lipner S. *The security development lifecycle*. Redmond: Microsoft Press, 2006. Vol. 8. 352 p.

4. *IEEE Guide for the use of IEEE standard dictionary of measures to produce reliable software*. IEEE, New York, IEEE Std. 982.2–1988. Available at: https://standards.ieee.org/standard/982_2-1988.html (accessed: 20.07.2019).

5. *IEEE Standard glossary of software engineering terminology*. IEEE, New York, IEEE Std. 610.12–1990, pp. 1–84. Available at: https://ieeexplore.ieee.org/document/159342 (accessed: 20.07.2019).

6. Ryzhov A. P. *Elementy teorii nechetkikh mnozhestv i ee prilozhenii* [Elements of theory of fuzzy sets and its applications]. Moscow, Dialog-MGU Publ., 1998. 81 p.

7. Iuganson A. N., Zakoldaev D. A. Razrabotka metodiki dlia rascheta otsenki tekhnologicheskoi bezopasnosti programmnykh sredstv [Development of methodology for calculating technological safety assessment of software]. *Vestnik UrFO. Bezopasnost' v informatsionnoi sfere*, 2017, no. 1 (23), pp. 20-23.

8. Barabanov A. V., Markov A. S., Tsirlov V. L. 28 magicheskikh mer razrabotki bezopasnogo programmnogo obespecheniia [28 magic measures for developing safe software]. *Voprosy kiberbezopasnosti*, 2015, no. 5 (13), pp. 2-10.

9. Iuganson A. N., Zakoldaev D. A., Rimsha A. S. Primenenie teorii nechetkikh mnozhestv dlia otsenki tekhnologicheskoĭ bezopasnosti programmnykh sredstv [Using theory of fuzzy sets to assess technological security of software]. *Matematicheskie metody v tekhnike i tekhnologiiakh – MMTT*, 2019, vol. 7, pp. 20-24.

10. Zadeh L. A. Fuzzy Logic. *Computer*, 1988, vol. 21, no. 4, pp. 83-93.

The article submitted to the editors 18.12.2019

INFORMATION ABOUT THE AUTHORS

Iuganson Andrei Nikolaevich – Russia, 197101, Saint-Petersburg; ITMO University; Assistant of the Department of Secure Information Technologies; a_yougunson@itmo.ru.

Zakoldaev Danil Anatolievich – Russia, 197101, Saint-Petersburg; ITMO University; Candidate of Technical Sciences, Assistant Professor; Decan Faculty of Secure Information Technologies; d.zakoldaev@mail.ru.