УДК 681.3

МЕТОДИКА ОЦЕНКИ ЗАЩИЩЕННОСТИ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ПРОВЕДЕНИИ ИСПЫТАНИЙ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Д. В. Колесников, А. Ю. Петров, В. Ю. Храмов

Федеральный государственный научно-исследовательский испытательный центр радиоэлектронной борьбы и оценки эффективности снижения заметности Министерства обороны Российской Федерации

Поступила в редакцию 01.03.2010 г.

Аннотация. Проведен анализ уязвимостей специального программного обеспечения, рассмотрены основные классы уязвимостей. Предложена методика оценки защищенности специального программного обеспечения при проведении испытаний автоматизированных систем, основанная на методах анализа уязвимостей.
Ключевые слова: оценка защищенности специального программного обеспечения, анализ уязвимостей, испытания автоматизированных систем.

Abstract. The vulnerability analysis the special software is lead, the basic classes vulnerability are considered. The technique of an estimation of security of the special software is offered at carrying out of tests of the automated systems based on methods of the vulnerability analysis.
Key words: estimation of security of the special software, vulnerability analysis, tests of the automated systems.

ВВЕДЕНИЕ

Одной из наиболее важных задач оценки качества специального программного обеспечения (СПО) современных автоматизированных систем (АС) при проведении испытаний является оценка его защищенности. Актуальность данной задачи обусловлена рядом причин, основными из которых являются:

— стремительный рост сложности программных средств приводит к тому, что задача анализа программного кода в условиях ограниченности ресурсов, выделяемых на проведение испытаний, становится неразрешимой;

— тесная взаимосвязь показателя защищенности с другими показателями качества СПО АС;

— влияние защищенности СПО на защищенность общего программного обеспечения (ОПО) и защищенность АС в целом.

Под защищенностью СПО будем понимать способность предотвращать случайный и умышленный несанкционированный доступ к своим функциям и данным, а также обнаруживать результаты такого доступа или действий по разрушению самого СПО и его данных [1]. Другими словами, защищенность СПО характеризуется наличием в нем уязвимостей.

Для выявления причин уязвимости программного обеспечения была проанализирована статистика компьютерных атак на АС. Результаты анализа показали, что большинство из них направлено на реализацию уязвимостей как ОПО, так и СПО, которые были внесены (возможно, преднамеренно) на этапе проектирования и разработки. Однако, в отличие от ОПО, выявлению уязвимостей СПО при проведении испытаний в настоящее время практически не уделяется внимания. Не выявленные на этом этапе уязвимости СПО АС могут явиться впоследствии плацдармом для реализации наиболее опасных угроз безопасности информации — скрытых.

Так одной из причин уязвимости программного обеспечения является несовершенство регламентированной нормативно-методической базы анализа и оценки защищенности программного обеспечения, вследствие чего проверка

защищенности СПО носит экспертный характер и зачастую не является объективной [2]. Для повышения объективности оценки защищенности СПО АС при проведении испытаний необходимо при подготовке испытаний анализировать существующие и уже выявленные уязвимости и эксплуатирующие их программы (exploit). На данный момент одним из наиболее информативных источников по уязвимым местам программного обеспечения можно считать список рассылки bugtraq, в котором были впервые публично рассмотрены многие программы атаки (http://www.bugtraq.com), базу данных уязвимых мест CVE (Common Vulnerabilities and Exposures — распространенные уязвимые места и ошибки), а также базу уязвимостей электронного ресурса Security Lab (http://www.securitylab.ru) [3].

1. ОСНОВНЫЕ УЯЗВИМОСТИ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ПЛАНИРОВАНИЕ ИСПЫТАНИЙ

Рассмотрим основные уязвимости, внесенные при проектировании и разработке, которые требуется оценить при проведении испытаний [4, 5]:

— переполнение буфера. В основе уязвимости лежит возможность переполнения буфера (стека, кучи) атакуемой программы, в результате чего нарушитель получает возможность выполнить любые команды на стороне узла, на котором запущенна эта программа. Основная причина — так называемый «плохой стиль» кодирования (особенно это касается C и C++). Также к этой категории относятся: ошибки индексации массива, ошибки в строках форматирования (formatstring), несовпадение размеров буфера при использовании Unicode и ANSI;

— SQL-injection. Данная уязвимость характерна для программ, которые в качестве входных данных получают параметры доступа к базе данных, после чего на их основе формируются SQL-запросы. Основная причина — отсутствие проверки корректности данных, поступающих на вход программы. Уязвимость позволяет нарушителю выполнить несанкционированные операции над содержимым баз данных путем вставки дополнительных команд в SQL-запросы;

— неправильная реализация обработки пакетов данных по протоколу TCP/IP. Причиной данной уязвимости, является неправильная реализация программных модулей, отвечающих за обработку входящих и исходящих пакетов данных. Уязвимость обусловлена отсутствием проверки корректности входящих и исходящих пакетов данных. В подавляющем большинстве атаки, реализующие данную уязвимость, приводят к отказу в обслуживании.

— уязвимости характерные для web-приложений (межсайтовое выполнение сценариев “cross site scripting (XSS)”, несанкционированный просмотр директорий “directory traversal” и др.) [3—5].

Защита от атак, направленных на уязвимости СПО, достигается благодаря правильному проектированию программ (чего добиться на практике очень сложно) и устранению распространенных ошибок (что является гораздо более простой задачей). Процесс управления качеством СПО АС согласно [4] должен включать следующие аспекты: управление рисками программного обеспечения, аудит программного обеспечения, тестирование системы безопасности (переполнения буфера, контроль доступа и проблемы выбора паролей, криптографические ошибки и т. д.), выполнение программного кода в замкнутом пространстве, защита от вредоносного кода, блокирование исполняемых файлов, отслеживание действий выполняемых программ, применение политик и расширяемых систем.

Чтобы оценить защищенность СПО АС приведенные методы обеспечения безопасности должны быть проанализированы и оценены на этапе проведения испытаний.

Результаты анализа существующих подходов к оценке защищенности программного обеспечения показали, что их применение при проведении испытаний СПО АС значительно затруднено, так как абсолютная проверка СПО на защищенность ни при одном из известных авторам подходов не осуществима [5—8]. Поэтому при планировании испытаний необходимо предварительно анализировать структуры испытываемых программ и входных данных. В частности, следует устанавливать те пути граф-схемы программы, использование которых при преобразовании данных наиболее вероятно. Для сложных программных комплексов она не имеет строго математического решения. Вместе с тем на практике нередко удается заранее установить наиболее вероятные ситуации, которые могут возникнуть в АС, а, следовательно, и наборы входных данных, описывающие эти ситуации [1].

Решение задачи планирования испытания включает в себя следующие этапы:

— нахождение всех путей реализации;

— выделение минимального подмножества путей, обеспечивающих проверку всех участков программы;

— разработка тестов для проверки выделенных путей.

Рассмотренный метод планирования на этапе автономных статистических испытаний компонентов СПО позволяет значительно уменьшить материальные и временные затраты на испытание программ.

2. МЕТОДИКА ОЦЕНКИ ЗАЩИЩЕННОСТИ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для повышения эффективности оценки защищенности, ее ускорения и удешевления необходимо разработать методику, позволяющую преодолеть недостатки подхода к оценке защищенности, недооценку ее роли в обеспечении требуемого уровня качества СПО, подмену тестирования уязвимостей процедурами типа проверки работоспособности на контрольном примере и т. п. Данная методика оценки защищенности, должна предусматривающей:

— знание назначения испытываемого СПО, условий его функционирования и требований к нему со стороны пользователей;

— автоматизацию всех наиболее трудоемких процессов и прежде всего моделирование среды функционирования, включая искажающие воздействия;

— ясное представление цели и последовательности испытания;

— целенаправленность и неизбыточность испытания, исключающие или минимизирующие повторение однородных процедур при одних и тех же условиях функционирования испытываемого СПО;

— систематический контроль за ходом, регулярное ведение протокола и журнала испытания;

— четкое, последовательное определение и исполнение плана испытания;

— четкое сопоставление имеющихся ресурсов с предполагаемым объемом испытания;

— возможность обеспечения, а также объективной количественной оценки полноты и достоверности результатов испытания на всех этапах.

Несмотря на высокую сложность, оценка защищенности СПО АС должна быть проведена в сжатые сроки, однако в настоящее время данная задача не является формализованной. С целью формализации и повышения достоверности оценки защищенности СПО при проведении испытаний АС авторами разработана методика оценки защищенности, основанная на методах анализа уязвимостей программного обеспечения [3, 4] и включает три этапа: подготовка, непосредственное тестирование и анализ результатов тестирования (рисунок 1).

Формально, предлагаемая методика представляет собой кортеж

$$\langle S, ID, M, C \rangle,$$

где S — испытываемое СПО;

$ID = \{ID_1, ID_2, ID_3\}$ — входные данные на трех этапах: подготовка, непосредственное тестирование и анализ результатов;

M — методы проведения испытаний;

C — средства проведения испытаний.

1. Подготовка

Подготовительный этап наиболее длительный и трудоемкий. Основными его задачами являются:

— планирование испытаний;

— разработка технологической схемы испытаний и испытательных средств;

— разработка тестовых вариантов;

— накопление предварительных статистических данных, характеризующих СПО.

Исходными данными (ID_1) на первом этапе являются: требования защищенности СПО согласно ТЗ, результаты оценки показателей правильности и устойчивости к ошибкам [9], результаты анализа и тестирования СПО разработчиком, классы уязвимостей СПО, СПО и ОПО, программная документация.

Подготовительный этап включает следующие процедуры:

— декомпозиция СПО на компоненты и интерфейсы:

$$f_1 : S \to K,$$

где $K = \{k_1, k_2, ..., k_j\}$ — множество компонент СПО.

$$f_2 : S \to I,$$

где $I = \{i_1, i_2, ..., i_j\}$ — множество интерфейсов СПО.

— выявление в компонентах предоставляемых интерфейсов:

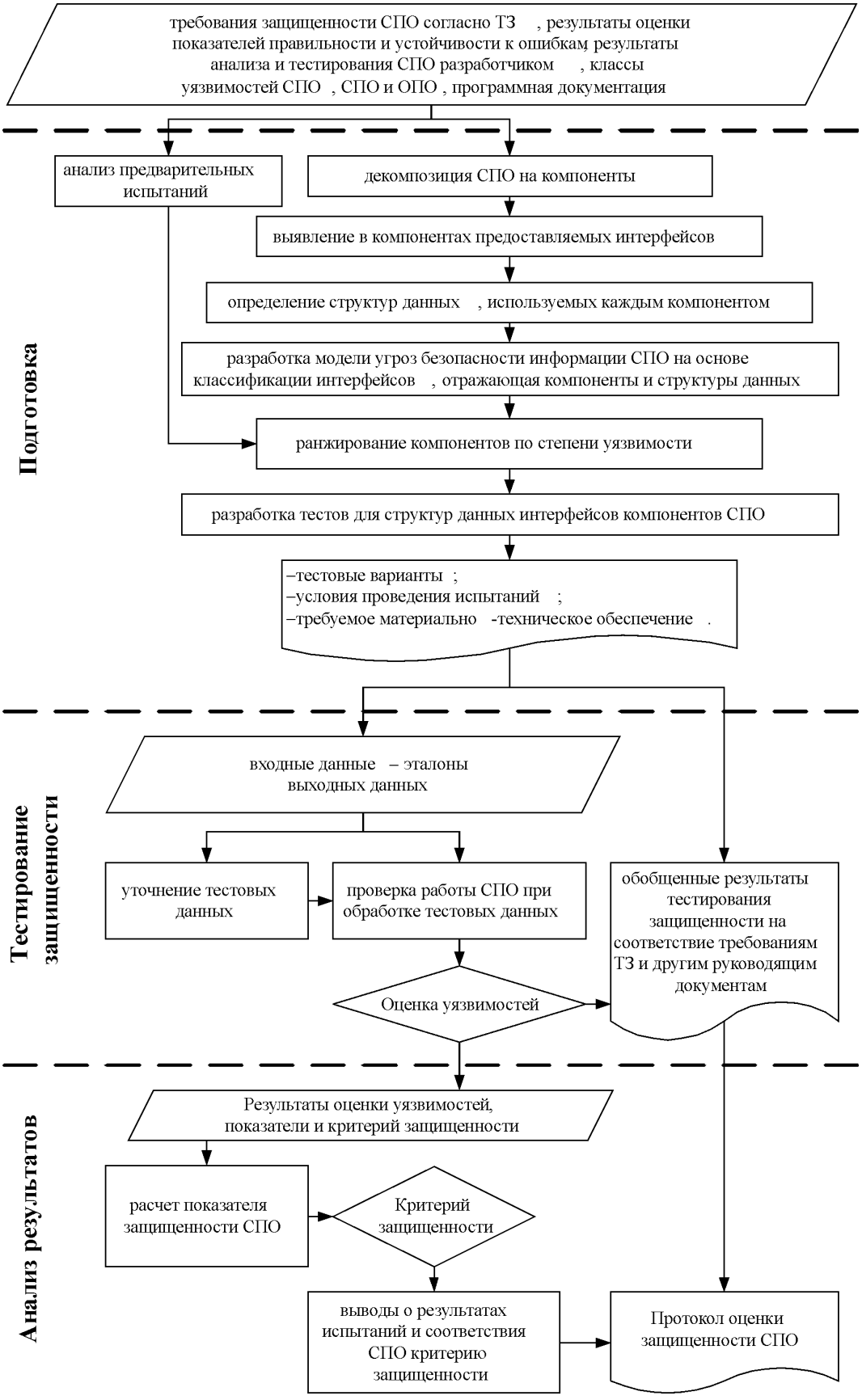

Рис. 1. Схема оценки защищенности СПО при проведении испытаний АС

$$f_3 : K \times I \to \{0,1\},$$

если $f_3(k_i, i_j) = 1$, то i-й компонент использует j-й интерфейс, в противном случае — $f_3(k_i, i_j) = 0$.

— определение структур данных, используемых j интерфейсом:

$$f_4 : I \to D,$$

где $D = \{d_1, d_2, \ldots d_m\}$ — множество структур данных.

— разработка модели угроз безопасности информации СПО на основе классификации интерфейсов, отражающая компоненты и структуры данных:

$$MU = \langle K, I, D, U, Y \rangle,$$

где U — множество угроз безопасности направленные на реализацию уязвимостей $Y = \{Y_l\}$, где $l = 1,2,3,4$:

1 — уязвимость переполнения буфера;

2 — уязвимость SQL-injection;

3 — неправильная обработка пакетов данных сетевых протоколов;

4 — уязвимости web-приложений.

— анализ предварительных испытаний;

— ранжирование интерфейсов по степени уязвимости:

$$f_5 : I \to R,$$

где $R = \{1,2,3,\ldots\}$ — множество рангов.

— разработка тестов для структур данных интерфейсов компонентов СПО:

$$\langle d^{l*}_{\text{вх}j}, d^{l*}_{\text{вых}j} \rangle.$$

где $d^{l*}_{\text{вх}j}, d^{l*}_{\text{вых}j} \in D$ — набор тестовых входных и выходных данных для j-го интерфейса при реализации l-й уязвимости.

Этап заканчивается оформлением документации, в которой отражаются:

— тестовые варианты (порядок действий, входные данные — эталоны выходных данных);

— условия проведения испытаний;

— требуемое материально-техническое обеспечение.

2. Тестирование защищенности

Исходные данные: тестовые данные (варианты).

Процедуры:

— уточнение тестовых данных;

— проверка работы СПО при обработке тестовых данных:

если $d_{\text{вых}j} \subseteq d^{l*}_{\text{вых}j}$, то j-й интерфейс имеет l-ю уязвимость,

где $d^{l*}_{\text{вых}j}$ — реакция интерфейса на тестовый вариант $\langle d^{l*}_{\text{вх}j}, d^{l*}_{\text{вых}j} \rangle$.

В документации по данному этапу отражаются обобщенные результаты тестирования защищенности на соответствие требованиям ТЗ и другим руководящим документам.

3. Анализ результатов тестирования

Исходные данные: описание выявленных в результате тестирования уязвимостей.

Процедуры:

— расчет показателя защищенности СПО.

Пусть g_j — коэффициент уязвимости j-го интерфейса, тогда показатель защищенности СПО (D) равен

$$D = \sum_{j=1}^{J} \frac{g_j}{r_j},$$

где r_j - ранг j-го интерфейса.

— выводы о результатах испытаний и соответствия СПО критерию защищенности:

$$D \le D_{mp},$$

где D_{mp} - требуемый уровень защищенности.

Результаты испытаний фиксируются в протоколах, которые обычно содержат следующие разделы:

— назначение тестирования и раздел требований технического задания, по которому проводится испытание;

— указание методик, в соответствии с которыми проводились испытания, обработка и оценка результатов;

— условия проведения тестирования и характеристика исходных данных;

— обобщенные результаты испытаний с оценкой их на соответствие требованиям технического задания и другим руководящим документам;

— выводы о результатах испытаний и степени соответствия оцениваемого СПО определенному разделу требований технического задания.

В дальнейшем по результатам испытаний согласно предложенной методике предполагается ведение базы данных по выявленным уязвимостям СПО АС и способам их реализации.

ВЫВОДЫ

Использование предложенной методики позволит сделать процесс оценки защищенности СПО АС «прозрачным» для специалистов испытательных лабораторий, что в условиях сжатых сроков поможет выполнить наиболее полную проверку. Методика согласуется с современными стандартами в области оценки безопасности (защищенности) программного

обеспечения АС и может быть использована для оценки программной продукции.

СПИСОК ЛИТЕРАТУРЫ

1. *Бойко А. А.* Технологическая схема оценки качества специального программного обеспечения при проведении государственных испытаний автоматизированных систем военного назначения / VII Всероссийская научно-техническая конференция «Актуальные вопросы разработки и внедрения информационных технологий двойного применения». — Ярославль: ЯВЗРУ ПВО (ВИ), 2006. — С. 133—137.

2. *Марков А. С., Миронов С. В., Цирлов В. Л.* Выявление уязвимостей в программном коде // Открытые системы, 2005. — № 12. С. 64—69.

3. *Хогланд Г., Мак Гроу Г.* Взлом программного обеспечения: анализ и использование кода.: Пер. с англ. — М.: Издательский дом "Вильяме", 2005.

4. *Ховард М., Лебланк Д.* Защищенный код: Пер. с англ. — 2-е изд., испр. М.: Издательско-торговый дом «Русская Редакция», 2004.

5. *Сердюк В. А.* Новое в защите от взлома корпоративных систем. М.: Техносфера, 2007. — 360 с.

6. *Маликов О. Р.* Автоматическое обнаружение уязвимостей в исходном коде программ // Известия ТРТУ. Тематический выпуск. Материалы VII Международной научно-практической конференции «Информационная безопасность». Таганрог: Издательство ТРТУ, 2005. № 4, С. 48—52.

7. *Хализев В. Н.* Алгебраическая модель анализа безопасности программных средств АС // Известия ТРТУ. Тематический выпуск. Материалы VII Международной научно-практической конференции «Информационная безопасность». Таганрог: Издательство ТРТУ, 2005. № 4, С. 60—64.

8. *Пальчун Б. П.* Дефектология интеллектуальных компьютерных программ // Известия ТРТУ. Тематический выпуск. Материалы VII Международной научно-практической конференции «Информационная безопасность». Таганрог: Издательство ТРТУ, 2005. № 4, С. 69—73.

9. *Бойко А. А.* Методика оценки правильности и устойчивости к ошибкам специального программного обеспечения автоматизированных систем военного обеспечения // Системный анализ и информационные технологии. Вестник Воронежского государственного обеспечения. — 2007. — № 1. — С. 106—115.

Колесников Даниил Владимирович — младший научный сотрудник Федерального государственного научно-исследовательского испытательного центра радиоэлектронной борьбы и оценки эффективности снижения заметности Министерства обороны Российской Федерации. Тел. 8-920-415-0828. E-mail: danil_k2@mail.ru.

Kolesnikov D. V. — researcher of Federal State Institution "Federal State Research Experimental Centre of Electronic Warfare and Estimation of Detection Effectiveness" of the Ministry of Defense of the Russian Federation. Tel. 8-920-415-0828. E-mail: danil_k2@mail.ru.

Петров Александр Юрьевич — адъюнкт очной адъюнктуры Федерального государственного научно-исследовательского испытательного центра радиоэлектронной борьбы и оценки эффективности снижения заметности Министерства обороны Российской Федерации. Тел. 8-915-580-2292.

Petrov A. U. — adjunct of Federal State Institution "Federal State Research Experimental Centre of Electronic Warfare and Estimation of Detection Effectiveness" of the Ministry of Defense of the Russian Federation. Tel. 8-915-580-2292.

Храмов Владимир Юрьевич — к.т.н., доцент, начальник отдела Федерального государственного научно-исследовательского испытательного центра радиоэлектронной борьбы и оценки эффективности снижения заметности Министерства обороны Российской Федерации. Тел. 8-903-030-9488.

Khramov V.U. — Ph. D of Engineering, assistant professor, The chief of a department of Federal State Institution "Federal State Research Experimental Centre of Electronic Warfare and Estimation of Detection Effectiveness" of the Ministry of Defense of the Russian Federation. Tel. 8-903-030-9488.