Марков А.С., Ларионцева Е.А., Стельмашук Н.Н. Многофакторные модели планирования сертификационных испытаний программных средств защиты информации // Вопросы радиоэлектроники. 2013. №2. С. 76-83.

681.3.06

МНОГОФАКТОРНЫЕ МОДЕЛИ ПЛАНИРОВАНИЯ СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ ПРОГРАММНЫХ СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ

Марков А.С., Ларионцева Е.А., Стельмашук Н.Н.

Рассмотрены вопросы планирования сертификационных испытаний. Дан обзор моделей испытаний и метрик сложности программ. Предложены многофакторные модели планирования испытаний по требованиям защищенности и по контролю отсутствия недекларированных возможностей.
Ключевые слова: метрики сложности, модели сложности, модели планирования испытаний, сертификационные испытания, сертификация средств защиты информации, недекларированные возможности.

Введение

Динамизм и рост сложности современных защищенных информационных технологий обуславливает гиперсложность оценки соответствия ИТ-продукции требованиям по безопасности информации. Несмотря на усилия ведущих разработчиков, проблема безопасности программных систем не получила своего окончательного решения, причем реальным механизмом контроля и управления информационной безопасностью объектов информатизации остается сертификация средств защиты информации [1,9]. В то же время планирование работ в этой области, зачастую, основано на экспертной оценке специалистов испытательных лабораторий, в то время как имеются академические наработки в области теории надежности и безопасности программ. В контексте сертификации средств защиты представляет интерес исследование практической применимости математических моделей для повышения достоверности

результатов планирования процесса сертификации, что и составляет основное содержание работы.

Обзор моделей планирования испытаний программ

С практической точки зрения удобна следующая классификация моделей испытаний программного обеспечения (ПО) средств защиты информации (СЗИ), включающая четыре класса моделей [7]:

- отладочные модели, основанные на прогонах ПО;

- временные модели роста надежности;

- модели полноты тестирования;

- модели сложности ПО.

Первые три класса моделей подразумевают наличие статистики результатов испытаний, полученных до сертификации, например, на этапах предварительных и государственных испытаний. Однако сертификационные испытания проводятся третьей (независимой) стороной, которая, на практике, не имеет полного доступа к системе сбора статистики. Поэтому представляют интерес модели сложности программ, позволяющие оценить метрики сложности ПО и связанные с ними показатели трудоемкости испытаний по требованиям безопасности информации.

Модели сложности ПО основаны на гипотезе о том, что уровень безошибочности ПО может быть предсказан с помощью показателей (метрик) сложности ПО. Это справедливо для непреднамеренных уязвимостей, так как, чем сложнее и больше программа, тем выше вероятность того, что программист ошибется при ее написании и модификации, а также тем сложнее будет локализовать ошибку в ПО.

Допускаются различные подходы к оценке комплексных показателей испытаний, обычно, путем получения полиномиальной зависимости от частных показателей – метрик сложности ПО [7-9]:

$$P = \sum_{i=1}^{k} b_i p_i + \sum_{i \neq j}^{k} b_{ij} p_i p_j + \sum_{i=1}^{k} b_{ii} p_i^2 + \cdots,$$

где: b_i - коэффициент *i*-й метрики, *k* - число метрик.

Метрики сложности ПО являются измеримыми количественными характеристиками особенностей реализации программ и условно разделяются на следующие: меры длины («объема») ПО, метрики сложности текста программ, метрики сложности по управлению, метрики сложности по данным, объектно-ориентированные метрики, а также интегральные метрики, включающие вышеназванные [1-15]. Надо понимать, что, чем проще метрика, тем она доступнее для экспертов испытательных лабораторий, а значит, модели будут более реализуемыми.

Постановка задачи

В качестве основного показателя планирования испытаний предлагается выбрать время *T*, затрачиваемое на испытания одним экспертом лаборатории.

Для учета разного опыта и знаний сотрудников испытательной лаборатории вводится коэффициент квалификации эксперта k_i, также при планировании работ учитывается число экспертов *n*. Фактически, при ограничениях на время сертификации можно получить трудоемкость сертификационных испытаний:

$$W = T / \sum_{i=1}^{n} k_i.$$

Было принято ограничить исследование двумя задачами:

- оценка трудоемкости сертификационных испытаний на отсутствие недекларированных возможностей в ПО средств защиты информации, отнесенной к государственной тайне;

- оценка трудоемкости сертификационных испытаний на соответствие требованиям по защите от несанкционированного доступа к информации конфиденциального характера.

Такое деление связано с тем, что принципиально более трудоемкими испытаниями является контроль отсутствия недекларированных возможностей, включающий

проведение статического и динамического анализа программного кода, подразумевающего выполнение длительной и трудоемкой процедуры декомпозиции программ. Данный вид испытаний является обязательным при проведении сертификации средств защиты информации, отнесенной к гостайне, причем по трудоемкости существенно превышает трудоемкость других видов испытаний.

Для сбора статистики о ПО СЗИ использовался анализатор кода АК-ВС, а для получения функциональных зависимостей - математический пакет Mathcad 15.0.

Расчет времени контроля отсутствия недекларированных возможностей

В рамках работы была проанализирована статистика сертификационных испытаний 20 программных систем: МПО S3300, S5300, S9300, ОС Windows 7, МЭ "Рубикон", МВТС-1, МПО ЕС5580.05/ЕС5580.05.М1, vGate-S R2, ПК "Интеграция", ОС Astra Linux SE, SecretNet 6.5, «Витрина», «Конструктор», XSpider, «Криптон-Замок», «Набат», ТС-669, TrustAccess-S, «БюрократЪ».

Для оценки значимости факторов моделей были выбраны пять метрик: число языков программирования, объем исходных кодов, число функциональных объектов, число потенциально-опасных конструкций (метрика АК-ВС), количество файлов, метрики МакКейба.

Результаты исследования представлены в таблице 1.

Таблица 1.

Формулы расчета времени испытаний

Показатель	Формула времени испытаний
Число языков программирования, l	$T = 0.358l^2 + 0.992l + e$
Объем исходных текстов, v	$T = 1.876v + e$
Число функциональных объектов, o	$T = 0.003419o + e$
Число потенциально-опасных конструкций, c	$T = 0.011c + e$
Количество файлов, f	$T = 0.014f + e$
Цикломатическое число МакКейба, V	$T = 0.001809V + e$

Приближенная формула расчета времени испытаний ПО СЗИ от наиболее значимых параметров получила следующий вид:

$T \approx 2.3v + 0.36l^2 + l + e,$

где *e* – коэффициент лаборатории, в рамках данного исследования $e = \overline{34{,}39}$.

Расчет времени испытаний по требования защищенности информации

В рамках работы была проанализирована статистика сертификационных испытаний следующих систем: Symantec SEP 11.0.5, Symantec DLP 10м«Enterprise Security», McAfee Total Protection for Endpoint, McAfee Host Data Loss Prevention, CheckPoint Endpoint, Lumension Device Control, Windows 7, SAP ERP, СЗИ для базисной системы SAP NetWaver Application Server, Базисная система SAP NetWaver Application Server, Check Point VPN-1 версии NGX, R65, Check Point VPN-1 версии NGX, R651, ASTARO Security Gateway Software Network Appliance 8, Check Point UTM-1, ПК "IBM DB2 v.9.7", Check Point Endpoint Security V R73, Symantec Control Compliance Suity версия 10, Windows Server 2008, Microsoft SQL Server 2008 R2, КП ArcGIS ИТБВ.00028-01, CheckPoint UTM-1 Edge X, CheckPoint UTM-1 Edge-N. Symantec DLP версия 11, Symantec DLP версия 10.

Для оценки значимости были выбраны две метрики: количество операционных систем и количество классов требований. Результаты исследования представлены в таблице 2.

Таблица 2.

Формулы расчета времени испытаний

Показатель	Формула времени испытаний
Количество операционных систем, *s*	$T = 0.361s + e$
Количество классов предъявляемых требований, *g*	$T = 1.36g + e$

Приближенная формула от наиболее значимых параметров получила следующий вид:

$$T \approx 0.8s + 1.6g + e,$$

где e – коэффициент лаборатории, на практике $e \approx 59$.

Заключение

Исследование показало, что полученные наиболее значимые факторы моделей (метрики) совпали с интуитивно используемыми экспертами, что соответствует опыту сотрудников аккредитованных испытательных лабораторий. При этом характеристики общего плана (объем, используемые языки) оказались более значимые по сравнению с теоретическими метриками (число МакКейба и др.), представленными в академической литературе [14]. Последнее объясняется тем, что сертификационные испытания являются достаточно творческим процессом, зачастую мало формализуемым по причинам социального характера и чрезвычайной сложностью и динамичностью программных систем.

С другой стороны, полученные результаты исследования подтверждают возможность получения научно-обоснованной оценки планируемых затрат на проведение испытаний. Можно рекомендовать предложенный подход для внедрения в системах менеджмента качества испытательных лабораторий.

Библиография

1. Александрович А.Е., Бородакий Ю.В., Чуканов В.О. Проектирование высоконадёжных информационно-вычислительных систем. М.: Радио и связь, 2004. 144 с.

2. Вареница В.В. Проблемы вычисления метрик сложности программного обеспечения при проведении аудита безопасности кода методом ручного рецензирования // Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия: Приборостроение. 2011. Спец.вып. Технические средства. С. 79-84.

3. Горюнов М.Н., Юдичев Р.М. Модель процесса проведения тематических исследований программного обеспечения // Проблемы развития технологических систем государственной охраны, специальной связи и специального информационного обеспечения : Восьмая научно-практическая конференция (Орёл, 12-14 марта 2013 г.): сборник материалов. - Орёл: Академия ФСО России. 2013. 2 с.

4. Гуров В.В. Оценка надежности программного обеспечения на начальных этапах его проектирования // Вестник Национального исследовательского ядерного университета МИФИ. 2012. Т. 1. № 2. С. 245.

5. Звездин С.В. Метрики как средство управления качеством // Открытые системы. СУБД. 2009. № 8. С. 51-54.

6. Маевский Д.А., Яремчук С.А. Оценка количества дефектов программного обеспечения на основе метрик сложности. // Электротехнические и компьютерные системы. 2012. № 7. С. 113-120.

7. Марков А.С. Модели оценки и планирования испытаний программных средств по требованиям безопасности информации // Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение. 2011. Спец.вып. Технические средства. С. 90-103.

8. Марков А.С. Оценка динамической сложности программного обеспечения на ПЭВМ // Методы и средства совершенствования сложных управляющих систем и комплексов: Учебное пособие. СПб: Мех. ин-т им. Д.Ф.Устинова (Военмех), 1992. С. 35-47.

9. Методы оценки несоответствия средств защиты информации / А.С.Марков, В.Л.Цирлов, А.В.Барабанов; Под ред. А.С.Маркова. М.:Радио и связь , 2012. 192 с.

10. Минаков В.А., Мирошников В.В., Дьякова А.В. Система оценивания объёма работ по контролю отсутствия недекларированных возможностей в программном обеспечении. Пат. 2445684 Российская Федерация, МПК G06F 11/36.; № 2010122911/08, 04.06.2010; заявл. 04.06.2010; опубл. 20.03.2012, бюл. № 8.

11. Мусолин А.К., Сидоров М.В., Червяков Р.С. Статистическое исследование системы метрик сложности программного обеспечения. // Вестник Рязанского государственного радиотехнического университета. 2003. № 12. С. 31-37.

12. Смагин В.А. Основы теории надежности программного обеспечения. СПб.: ВКА им.А.Ф.Можайского, 2009. 336 с.

13. Тырва А.В., Хомоненко А.Д. Метод планирования тестирования сложных программных комплексов на этапах проектирования и разработки Научно-технические ведомости СПбГПУ. 2009. № 82. С. 125-131.

14. IEEE Std. 1061-1998 IEEE Computer Society: Standard for Software Quality Metrics Methodology, 1998. 20 p.

15. Ramos I.C. Statistical Procedures for Certification of Software Systems. TSIM, 2009. 195 p.