УДК 004.056:378(06)

А.С. Марков, В.Л. Цирлов, С.А. Смолин, А.С. Корсунский

# СОВРЕМЕННЫЕ МЕТОДЫ ПРОВЕДЕНИЯ СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ ПРОГРАММНОГО КОДА ПРИ ОТСУТСТВИИ ИСХОДНЫХ ТЕКСТОВ

***Марков Алексей Сергеевич,*** *кандидат технических наук, доцент кафедры «Информационная безопасность» Московского государственного технического университета им. Н.Э. Баумана, CISSP, SBCI. Генеральный директор ЗАО «НПО «Эшелон». Научные интересы: тестирование и сертификация программного обеспечения по требованиям безопасности информации, вопросы обеспечения надежности программного обеспечения. [e-mail: mail@npo-echelon.ru].*

***Цирлов Валентин Леонидович,*** *кандидат технических наук, доцент кафедры «Информационная безопасность» МГТУ им. Н.Э. Баумана, CISSP, AMBCI. Исполнительный директор ЗАО «НПО «Эшелон». Научные интересы: аудит защищенности и формальная верификация программного обеспечения и автоматизированных систем, управление информационной безопасностью. [e-mail: mail@npo-echelon.ru].*

***Смолин Сергей Анатольевич,*** *окончил механико-математический факультет Ульяновского государственного университета. Начальник научно-исследовательской лаборатории ФНПЦ ОАО «НПО «Марс». Имеет статьи по сертификации в области защиты информации. [e-mail: mars@mv.ru]*

***Корсунский Андрей Сергеевич,*** *кандидат технических наук, окончил факультет радиосвязи Ульяновского филиала Военного университета связи, адъюнктуру Военной академии связи им. С.М. Буденного. Ведущий инженер-программист ФНПЦ ОАО «НПО «Марс». Имеет статьи и изобретения в области радиоэлектронной защиты, безопасности связи и информации. [e-mail: aksspb@mail.ru].*

Аннотация

В статье исследуются особенности современных систем разработки программного обеспечения (ПО) с точки зрения вопросов проведения аудита безопасности и сертификационных испытаний программных продуктов, не имеющих исходных текстов. Показана возможность выявления уязвимостей, закладок и ошибок, а также подготовки отчетов сертификационных испытаний для сред программирования JVM (Java) и .NET (C#).

Ключевые слова: программное обеспечение, уязвимость, недекларированные возможности, сертификация, информационная безопасность, аудит безопасности.

***Alexey Sergeevich Markov****, Candidate of Engineering, Associate Professor at the Chair 'Information Security' at Bauman Moscow State Technical University, CISSP, SBCI; Director General of 'Research-and-Production Association 'Eshelon', JSC; interested in testing and certifying of software as per requirements of information security, software reliability. e-mail: mail@npo-echelon.ru.*

***Valentin Leonidovich Tsirlov****, Candidate of Engineering, Associate Professor at the Chair 'Information Security' at Bauman Moscow State Technical University, CISSP, AMBCI; Executing Chief Manager of 'Research-and-Production Association 'Eshelon', JSC; interested in audit of security and formal verification of software and computer-aided systems, control of information security. e-mail: mail@npo-echelon.ru.*

***Sergey Anatolyevich Smolin****, graduated from the Faculty of Mechanics and Mathematics at Ulyanovsk State University; head of a research laboratory at FRPC OJSC 'RPA 'Mars'; author of articles in certification in the field of information security. e-mail: mars@mv.ru.*

***Andrey Sergeevich Korsunky****, Candidate of Engineering, graduated from the Faculty of Radio-Communications at Ulyanovsk branch of the Military Communications University, finished his post-graduate studies at the Military Communications Academy named after S. Budenny; lead programmer of FRPC OJSC 'RPA 'Mars'; author of articles and inventions in the field of radio-electronics protection, communications and information security. e-mail: aksspb@mail.ru.*

Abstract

The article researches features of state-of-the-art systems of software development from the point of view of audit of security and certification tests of software without source codes. It also shows possible detection of vulnerability, bookmarks and errors as well as development of certification-test reports for the programming environments JVM (Java) and .NET (C#).

Key words: software, vulnerability, not declared capabilities, certification, information security, security audit.

## Введение

В настоящее время проблема уязвимости программного кода является одной из самых важных в области информационной безопасности (ИБ) компьютерных систем. Это связано с тем, что именно из-за наличия уязвимостей в программном обеспечении возможно проведение обширного класса компьютерных атак и распространение вирусных «эпидемий» [1].

Проблема безопасности программного кода решается путем сертификации ПО на отсутствие недекларированных возможностей (НДВ) [2] либо, когда требования по сертификации не предъявляются, путем аудита безопасности кода [3, 4]. Данные подходы подразумевают предоставление разработчиками продуктов исходных текстов и спецификаций на ПО. На практике иногда возникают ситуации, когда разработчик не может предоставить исходный код, как правило, по причине его утери или некорректной модификации версий. В этих случаях предлагается использовать возможности функционального тестирования (по методу «черного ящика»). Однако такой подход не регламентирован нормативными документами и чрезвычайно трудоемок при выявлении программных закладок и некорректностей программирования, влияющих на безопасность. В данной статье рассмотрен класс систем программирования, позволяющих получить информацию о структуре ПО с целью проведения испытаний по требованиям безопасности.

## Подходы к проверке программ без исходных текстов

Необходимо понимать, что угрозу ИБ составляют только те уязвимости, которые действительно могут быть реализованы в той или иной среде с учетом существующих параметров окружения. Поэтому в процессе испытаний важно не только выявить уязвимость, но и оценить возможность ее реализуемости, а также определить пути снижения возможности ее реализации [5].

С учетом сказанного, для проведения анализа без исходных текстов программ можно предложить три класса процедур:

1. Оценка возможности декомпиляции продукта с целью проведения испытаний с учетом имеющейся нормативной базы.

2. Проведение проверок загрузочного кода с целью экспертной оценки степени безопасности кода и принятие решения о возможности снижения рисков, связанных с использованием рассматриваемого ПО.

3. Оценка возможности использования дополнительных средств защиты с целью снижения рисков, связанных с использованием рассматриваемого ПО.

Следует сказать, что до недавнего времени первый класс процедур связывали с понятием деассемблирования, не позволяющего решить поставленные задачи. Однако исследование показало, что ряд современных систем программирования позволяют провести высококачественную декомпиляцию таким образом, что возможно проведение испытаний по требованиям безопасности информации.

## Декомпиляция программных продуктов

Под высококачественной декомпиляцией мы понимаем возможность восстановить исходные тексты программы с полным сохранением иерархии функциональных и информационных объектов, их связей и управляющих структур с целью проведения испытаний по требованиям ИБ.

Проведенный анализ показал возможность высококачественной декомпиляции кода для ряда программных платформ. В первую очередь это касается байт-кода виртуальной машины Java (Java Virtual Machine, JVM), для которой известны реализации ряда языков программирования: Java, NetRexx, Ruby (JRuby), JavaScript (Rhino), Python (Jython), Groovy, PHP (Quercus), Clojure, Scala и др. В ряде случаев положительные результаты получены для платформы .NET, где применяется CLR (Common Language Runtime). В частности, в данной среде возможно исполнение кода, написанного на языках программирования: ASP.NET, C# , Visual Basic .NET, C++/CLI, F#, J#, JScript. NET, Windows PowerShell и др. Особенностью этих платформ, а также других, подобных им (например, ActionScipt Virtual Machine и Microsoft P-CODE Virtual Machine), является то, что исходные коды компилируются не в команды микропроцессора Intel, а в промежуточное бинарное представление, которое уже на этапе выполнения будет преобразовано в инструкции процессора. На рисунке показана схема сборки на примере среды CLR и промежуточного представления MSIL.

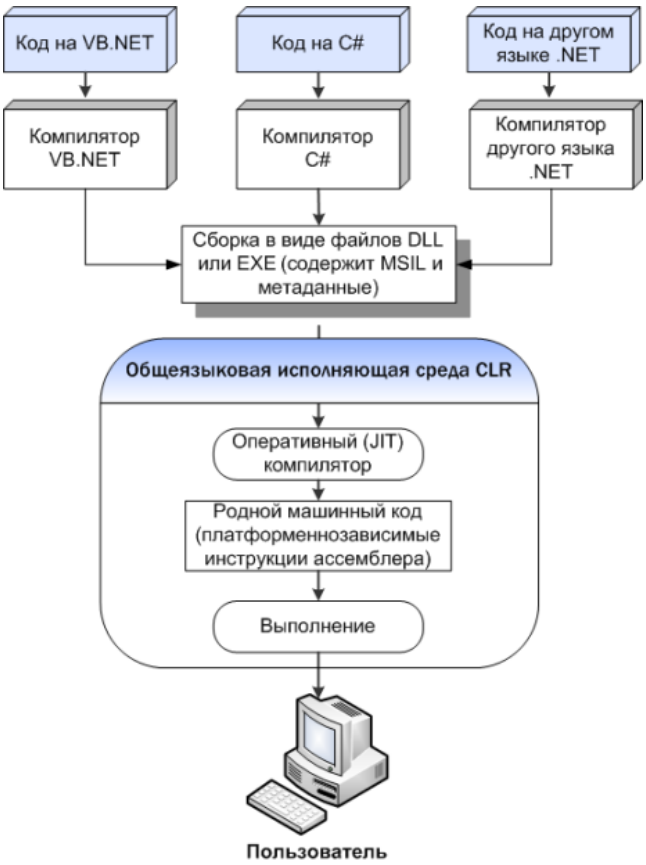


Рис. Схема сборки на примере среды CLR и промежуточного представления MSIL

### Методика проведения эксперимента

Для проведения эксперимента были выбраны программные продукты, проходящие тематические исследования или сертификационные испытания в аккредитованной испытательной лаборатории ЗАО «НПО «Эшелон». В качестве программных платформ были выбраны:

- виртуальная машина Java, язык программирования Java;
- среда CLR, язык программирования C#;
- система программирования C/C++ для архитектуры x86.

Согласно исследованию TIOBE Software, первые две платформы относят к наиболее популярным в мире.

На испытания представлялись исходные тексты и соответствующие декомпилированные тексты программ. Испытания проводились на предмет оценки возможности:

- проведения проверок с целью идентификации основных классов уязвимостей кода;
- проведения проверок (и построения отчетов), соответствующих сертификационным испытаниям на отсутствие НДВ.

В качестве инструментария по выявлению уязвимостей использовался сертифицированный в ФСТЭК России и Минобороны России анализатор безопасности программного кода АК-ВС, поддерживающий международную классификацию уязвимостей CWE (Common Weakness Enumerations).

Таблица

Результаты эксперимента по выявлению уязвимостей исходных текстов и текстов, полученных в результате декомпиляции

| Задача | Платформа (язык программирования) | | |
|---|---|---|---|
| | JVM (Java) | .NET (C#) | x86 (C/C++) |
| Возможность декомпиляции | + | + | + |
| Качество декомпиляции | 5 | 5 | 2 |
| Поиск уязвимостей в декомпилированном тексте | 5 | 5 | 2 |
| Компиляция исполняемых файлов из полученных исходных текстов | 5 | 4 | 3 |
| Список функциональных объектов | 5 | 5 | 3 |
| Список информационных объектов | 5 | 5 | – |
| Матрица связей по информации | 4 | 5 | – |
| Матрица связей по управлению | 5 | 5 | 3 |
| Трассы вызовов | 4 | 4 | – |

### Результаты эксперимента

Эксперимент показал высокую корреляцию результатов проверок исходных текстов и текстов, полученных в результате декомпиляции для платформ JVM (Java) и .NET (C#). В частности, в рамках эксперимента была доказана возможность выявления уязвимостей кода, а также проведения основных проверок (и формирования отчетов) в рамках статического анализа на отсутствие НДВ. Результаты исследования приведены в таблице. Экспертная оценка выполнялась по 5-бальной шкале.

### Заключение

Проведенное исследование показало потенциальную возможность проведения сертификационных испытаний для ПО без исходных кодов, разработанного с использованием современных сред программирования JVM (Java) и .NET (C#).

Доказано, что выявленные некорректности декомпиляции не влияют на возможность оценки наличия в ПО потенциально опасных фрагментов и зон рисков.

Проведенное исследование и опыт испытаний ПО (разработанного с применением указанных систем программирования) показали, что использование декомпилированного кода позволяет выполнить значимые процедуры, необходимые для проведения испытаний на соответствие требованиям руководящего документа «Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей» [6].

СПИСОК ЛИТЕРАТУРЫ

1. Марков А.С., Щербина С.А. Испытания и контроль программных ресурсов // Information Security. – 2003. – № 6. – С. 25–26.

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

3. Марков А.С., Цирлов В.Л. Аудит программного кода по требованиям безопасности. Ч. 1 // Information Security. – 2008. – № 2. – С. 56–57.

4. Марков А.С., Цирлов В.Л. Аудит программного кода по требованиям безопасности. Ч. 2 // Information Security. – 2008. – № 3. – С. 46–47.

5. Марков А.С., Фадин А.А. Разработка стандарта предприятия по выявлению недекларированных возможностей в программном обеспечении // Стандартизация информационных технологий и интероперабельность : тр. III-й всерос. конф. SITOP 2009, Москва, 27 октября 2009 г. – М. : РАН, 2009. – С. 78–80.

6. Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей: РД : утв. Гостехкомиссией при Президенте Российской Федерации 04.06.1999. – URL : www.fstec.ru/_docs/doc_3_3_010.htm.