СТАТИСТИКА ВЫЯВЛЕНИЯ УЯЗВИМОСТЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ПРОВЕДЕНИИ СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ

Барабанов А.В.[1], Марков А.С.[2], Фадин А.А.[3], Цирлов В.Л.[4]

В работе рассмотрены практические аспекты внедрения методики выявления уязвимостей в программном обеспечении в повседневную детальность аккредитованной испытательной лаборатории. Представлены результаты апробации методики выявления уязвимостей в рамках исследования программного обеспечения с открытым исходным кодом и программного обеспечения, являющегося объектом исследований при проведении сертификационных испытаний по требованиям безопасности информации. Приведены результаты исследования, демонстрирующие распределение выявленных уязвимостей по типам атак, стране происхождения, используемым при разработке языкам программирования, методам выявления уязвимостей и пр. Проанализирован опыт зарубежных систем сертификации средств защиты информации в области выявления уязвимостей сертифицируемого программного обеспечения. Основным выводом, сформулированным по результатам проведенного исследования, является необходимость внедрения практик разработки безопасного программного обеспечения в процессы жизненного цикла разработки. Сформулированы выводы и рекомендации испытательным лабораториям по внедрению методики анализа уязвимостей.

***Ключевые слова**: сертификация программного обеспечения, защита информации, анализ уязвимостей.*

DOI: 10.21681/2311-3456-2017-2-2-8

Введение

Выполнение анализа уязвимостей программного обеспечения (ПО) в настоящее время является основным видом деятельности, выполняемым экспертами испытательных лабораторий (ИЛ) отечественных систем сертификации средств защиты информации [1, 2]. Данный вид работ выполняется как при сертификации на соответствие требованиям утвержденных ФСТЭК России профилей защиты, в которых в явном виде включены требования семейства доверия «Анализ уязвимостей» [3], так и при испытаниях на соответствие требованиям технических условий или классических руководящих документов [4]. Методология анализа уязвимостей, рекомендуемая в настоящее ФСТЭК России, заключается в совместном использовании подходов, изложенных в национальном стандарте ГОСТ Р ИСО/МЭК 18045 и международном стандарте ISO/IEC TR 20004. В общем случае методология предполагает выполнение следующих шагов [5].

1. Выявление известных (подтвержденных) уязвимостей объекта сертификации. При выполнении данного шага экспертами ИЛ осуществляется поиск известных (подтвержденных) уязвимостей в общедоступных источниках информации, например: в Банке данных угроз безопасности информации ФСТЭК России или ресурсе CVE.

2. Выявление ранее не опубликованных уязвимостей объекта сертификации. При выполнении данного шага экспертами ИЛ на основе анализа данных об объекте сертификации (исходный код, доступная документация, информация из открытых источников) определяется перечень потенциальных уязвимостей объекта сертификации и для каждой идентифицированной потенциальной уязвимости разрабатывается и выполняется тест на проникновение с целью определения верности сделанного предположения.

В связи с тем, что требования по проведению анализа уязвимостей является относительно новым для отечественных систем сертификации средств защиты информации, на текущий момент времени практически отсутствуют методические документы для ИЛ, которые могли бы использоваться для проведения эффективного анализа уязвимостей. Этим обусловлена актуальность задачи разработки и совершенствования методического обеспечения анализа уязвимостей при проведении сертификационных испытаний по тре-

1 Барабанов Александр Владимирович, кандидат технических наук, АО «НПО «Эшелон», г. Москва, a.barabanov@npo-echelon.ru

2 Марков Алексей Сергеевич, доктор технических наук, старший научный сотрудник, МГТУ им. Н.Э.Баумана, г. Москва, a.markov@bmstu.ru

3 Фадин Андрей Анатольевич, АО «НПО «Эшелон», г. Москва, a.fadin@npo-echelon.ru

4 Цирлов Валентин Леонидович, кандидат технических наук, МГТУ им. Н.Э.Баумана, г. Москва, v.tsirlov@bmstu.ru

бованиям безопасности информации. В рамках выполнения данного исследования была апробирована комбинированная методика анализа уязвимостей ПО, предложенная в работах [5, 6], и сформулированы рекомендации для экспертов аккредитованных ИЛ.

Адаптированная методика анализа уязвимостей программного обеспечения

В рамках проведения исследования указ анная комбинированная методика анализа уязвимостей была адаптирована с учётом особенностей работы аккредитованной ИЛ НПО «Эшелон» (рис. 1).

Краткое описание этапов и шагов адаптированной методики анализа уязвимостей ПО представлено далее по тексту.

Этап 1. Проведение статического анализа исходных текстов ПО.

Шаг 1. Идентификация множества исходных текстов, участвующих в компиляции объекта сертификации. На данном шаге экспертами ИЛ проводится контроль полноты и отсутствия избыточности представленных на испытание исходных текстов ПО с целью определения точного множества исходных текстов, участвующих в компиляции ПО. При выполнении этого шага экспертам ИЛ используется информация, генерируемая сборочной системой и различными инструментальными средствами (мониторы файловой системы, программы аудита файловой системы и пр.). Основная цель данного шага – зафиксировать перечень файлов исходных текстов, участвующих в компиляции объекта сертификации.

Шаг 2. Проведение статического сигнатурного анализа [7-9] в отношении зафиксированного на шаге 1 множества исходных текстов. Используемый статический анализатор должен обладать возможностью поиска потенциально опасных конструкций в исходных текстах и формирования данного перечня с присвоением каждой обнаруженной потенциально опасной конструкции идентификатора базы CWE.

Этап 2. Формирование перечня потенциальных уязвимостей объекта сертификации и шаблонов атак

Шаг 3. Обработка полученного перечня потенциально опасных конструкций с использованием критериев фильтрации, представленных в разделе 6.1.2.1 стандарта ISO/IEC TR 20004.

Шаг 4. Формирование перечня шаблонов атак, являющихся актуальными для исследуемого ПО, с использованием последовательности действий, представленной в разделе 6.1 стандарта ISO/IEC TR 20004. При выполнении данного шага, помимо информации, представленной в исходных текстах объекта сертификации, эксперты ИЛ используют представленную для проведения испытаний документацию (техническую, программную, эксплуатационную), информацию об известных уязвимостях ПО, схожих с объектом сертификации.

Этап 3. Формирование пар «потенциальная уязвимость – шаблон атаки».

Шаг 5. Обработка перечней потенциальных уязвимостей и шаблонов атак, полученных на этапе 2, с использованием последовательности действий, представленных в разделе 6.1.2.2 стандарта ISO/IEC TR 20004.

Этап 5. Проведение тестирования на проникновение.

Шаг 6. Разработка тестов на проникновение на основе сформированного перечня потенциальных уязвимостей и шаблонов атак.

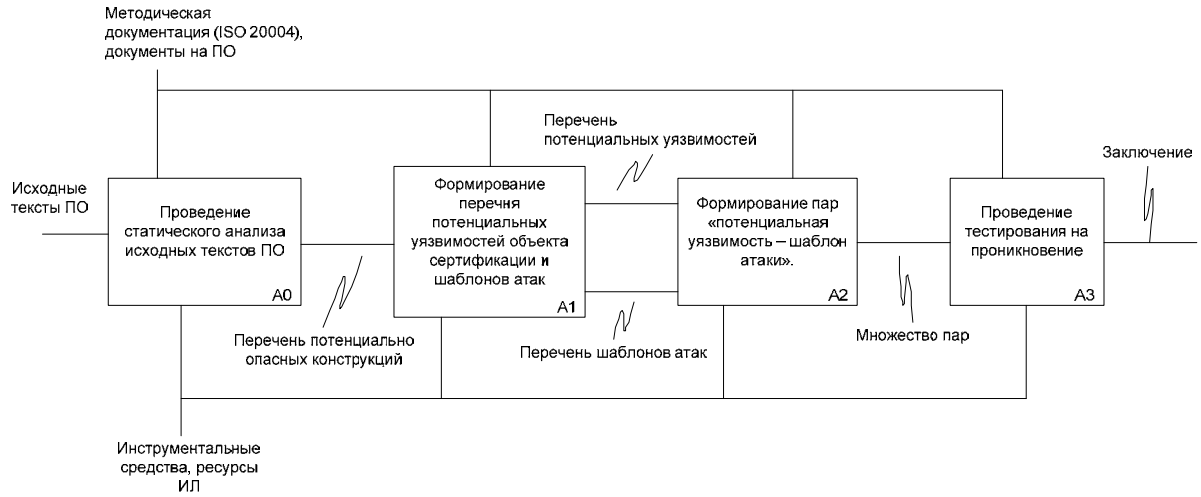

***Рис.1**. IDEF0-диаграмма адаптированной методики анализа уязвимостей ПО*

Шаг 7. Монтаж испытательного стенда и проведение тестов на проникновение с использованием разработанных текстов.

Шаг 8. Определение актуальных уязвимостей ПО по результатам выполнения тестов проникновения и оформление отчетных материалов.

Постановка эксперимента

Экспериментальные исследования адаптированной методики анализа уязвимостей ПО проводились в течение 2 лет на научно-исследовательской базе НПО «Эшелон» экспертами аккредитованной ИЛ.

Объектами исследования являлись:

- ПО, проходящее тематические и сертификационные исследования в аккредитованной ИЛ (группа №1, 80 объектов исследования);

- ПО с открытым исходным кодом (группа №2, 102 объекта исследования).

При выполнении сигнатурного анализа исходных текстов экспертами ИЛ использовалось средство статического анализа «AppChecker» (разработчик - НПО «Эшелон») [10]. Для проведения тестов на проникновения экспертами ИЛ использовались рекомендации различных тематических ресурсов (CAPEC, OWASP) и инструментальное средство «Сканер-ВС» (разработчик - НПО «Эшелон») [11]. Монтаж и наладка испытательных стендов, используемых при проведении тестирования на проникновение (шаг 7), выполнялся экспертами ИЛ в полном соответствии с требованиями эксплуатационной и технической документации на объекты исследований.

Результаты экспериментальных исследований

В рамках апробации методики для объектов исследования группы №1 экспертами ИЛ было выявлено 106 уязвимостей ПО. Для всех выявленных уязвимостей ПО было получено подтверждение об их актуальности со стороны разработчика ПО. На рис.2 показано распределение выявленных уязвимостей по типам успешных атак, использующих выявленную уязвимость. Зафиксирован ряд дефектов, которые трудно идентифицировать как преднамеренные, однако их можно эксплуатировать при проведении компьютерных атак, например: атак типа «межсайтовый скриптинг» и внедрение SQL-кода. Исследования показали, что в ПО в явном виде встречаются программные закладки, маскируемые под отладочные средства, например встроенные учетные записи и мастер-пароли. В категорию «Другое» попали менее популярные типы уязвимостей: уязвимости, связанные с XML-инъекцией или фиксацией сессии.

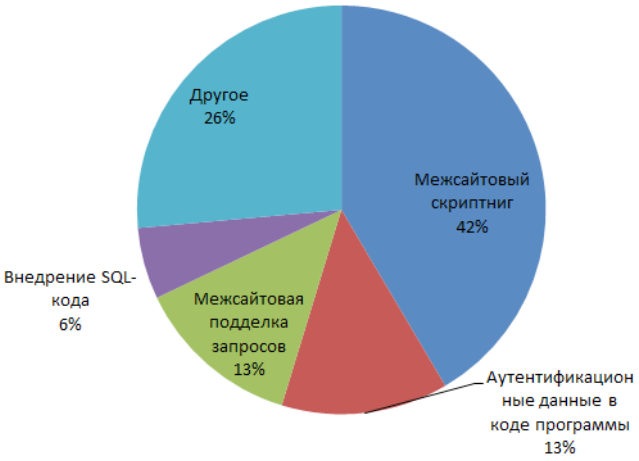

***Рис.2**. Распределение выявленных уязвимостей по типам атак на ПО*

Говоря о распределении уязвимого ПО по типам изделий информационных технологий (операционные системы, антивирусные решения, системы обнаружения вторжений и пр), следует отметить, что полученные результаты соответствуют общемировой практике - большинство уязвимостей обнаружено в прикладном ПО (рис. 3). Следует отметить, что в некоторых исследованных образцах ПО для сетевого оборудования были обнаружены встроенные учетные записи.

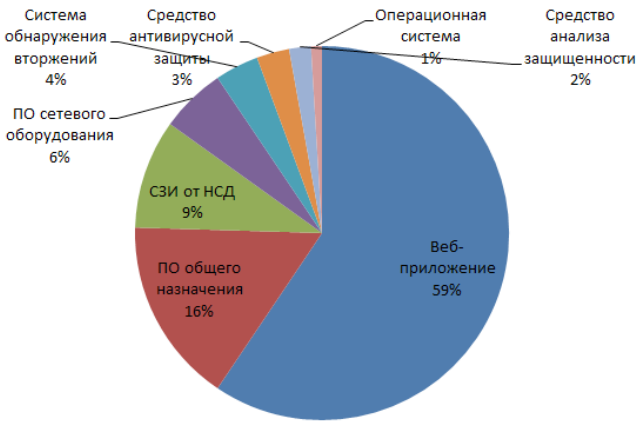

***Рис.3**. Распределение уязвимого ПО по типам изделий информационных технологий*

Часть выявленных уязвимостей была обнаружена в результате исследования исходного кода объектов группы №1 с использованием методов структурного (статического) анализа (рис.4).

Доля уязвимостей, обнаруженных в ПО отечественного производства, значительно больше доли уязвимостей, обнаруженном в зарубежном ПО (рис. 5). Это объясняется существенными

***Рис.4**. Статистика по методам выявления уязвимостей*

различиями в уровнях зрелости процессов жизненного цикла разработки безопасного ПО, внедренных у зарубежных и отечественных разработчиков ПО. Однако следует отметить, что при проведении исследований для ПО зарубежного производства в большинстве случаев разработчиками не обеспечивался доступ к исходному коду объектов исследования, что делало принципиально невозможным выполнение этапа 1 – перечень потенциальных уязвимостей формировался экспертами ИЛ в условиях отсутствия информации о потенциально опасных конструкция в исходном коде ПО.

***Рис.5**. Статистика по стране происхождения ПО*

Среднее время исправления уязвимости разработчиком ПО составило 3 недели.

Следует указать, что современные программные комплексы включают модули программ с открытым кодом. Исследование группы №2 (ПО с открытым исходным кодом) показало, что такие программы тоже содержат уязвимости. В результате проведения исследований было обнаружено 154 дефекта ПО (подтверждены разработчиками ПО), из них 11 – дефекты, приводящие к уязвимости ПО. Все дефекты ПО были обнаружены исключительно с использованием статического сигнатурного анализа исходного кода ПО. Распределения обнаруженных уязвимостей по языкам программирования и типам дефектов CWE представлены на рисунках (рис. 6, рис.7). Наиболее популярными типами обнаруженных дефектов ПО стали ошибки в логических выражениях (CWE-569, Expression Issues) и ошибки, связанные с разыменованием нулевого указателя (CWE-476 NULL Pointer Dereference).

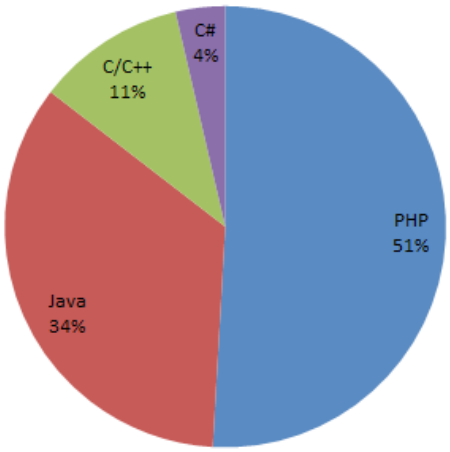

***Рис.6**. Распределение уязвимостей ПО (ПО с открытым исходным кодом) по языкам программирования*

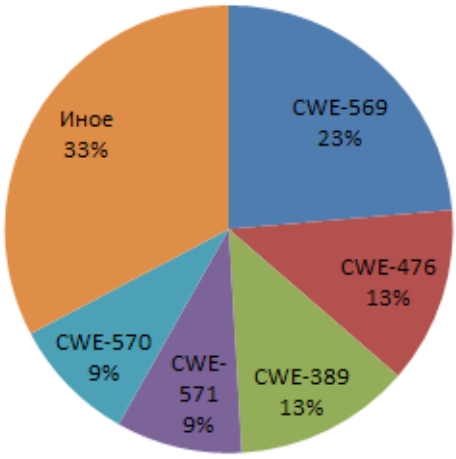

***Рис.7**. Распределение уязвимостей ПО (ПО с открытым исходным кодом) по типам дефектов CWE*

Состояние проблемы в зарубежных системах сертификации

Напомним, что из-за нововведений в зарубежных системах сертификации отчеты ИЛ, которые содержат общие сведения о проведённом анализе уязвимостей, публикуются на официальных сетевых ресурсах систем сертификации. Были проанализированы отчеты ИЛ за период 2016-2017 гг. (выборка – 33 отчета), опубликованные на сайте NIAP – регулятора системы сертификации США. Среди проанализированных отчетов большую часть составили отчеты по результатам проведения испытаний на соответствие требованиям профилей защиты к сетевым устройствам (28 отчетов). В остальной части отчетов (5 отчетов) были отражены результаты испытаний на соответствие требованиям профилей защиты к прикладным программам, операционным системам, средствам управления политиками разграничения доступа и средствам защиты мобильных устройств.

Основные результаты проведенного анализа представлены далее по тексту.

1. В ходе выполнения всех работ зарубежных ИЛ выполнялся поиск информации об известных уязвимостях объекта сертификации в общедоступных базах данных. Некоторые ИЛ выполняли поиск известных уязвимостей не только по ключевым словам, непосредственно связанным с объектом сертификации (название и версия ПО, наименование разработчика ПО), но и по идентификационным данным, имеющим отношение к заимствованным компонентам.

2. Только в половине работ испытательные лаборатории выполняли дополнительные тесты на проникновение. В большинстве работ использовался стандартный набор тестов, применимый практически ко всем типам объектов сертификации (например, сканирование сетевых портов). Только в одной работе была представлена информация о проведении тестов на проникновения, основанных на потенциальных уязвимостях объекта сертификации, сформулированных с учетом анализа свидетельства разработчика.

3. Во всех работах, связанных с сертификацией по требованиям профилей защиты для сетевых устройств, проводилось фаззинг-тестирование. При этом, как правило, использовались программные средства автоматизации собственной разработки.

4. В своих работах испытательные лаборатории не использовали указания ISO/IEC TR 20004 по формованию перечня потенциальных уязвимостей на основе анализа баз данных CWE и CAPEC. Это связано с тем, что требование по предоставлению доступа к исходному коду сертифицируемого ПО не является обязательным в зарубежных системах сертификации. Анализ выполняется только в объеме, соответствующем требованиям ПЗ – дополнительные исследования выполняет только малое число испытательных лабораторий.

Выводы

По результатам проведённого исследования можно сделать вывод об эффективности комбинированной методики анализа уязвимостей ПО и о целесообразности ее внедрения в повседневную деятельность экспертов аккредитованных ИЛ. Анализ уязвимостей ПО должен быть первой активностью, выполняемой в рамках сертификационных испытаний, поскольку выявление уязвимостей в объекте сертификации на более поздних стадиях (например, после выполнения функционального тестирования или проверки производства) влечет за собой повторение полного цикла сертификационных испытаний. Следует отметить, что выявление известных (подтвержденных) уязвимостей объекта сертификации рекомендуется выполнять как на начальной, так и на конечной стадиях сертификационных испытаний.

По результатам апробации методики можно сформулировать следующие краткие выводы:

- большинство уязвимостей зафиксировано на уровне прикладного ПО, а не средств защиты информации;

- количество обнаруженных уязвимостей существенно зависит от существующих в организации – разработчике ПО процессов жизненного цикла разработки безопасного ПО.

- наиболее критичные уязвимости были выявлены только в случае предоставления доступа к исходным текстам ПО;

- большая часть выявленных в рамках исследования уязвимостей могла быть обнаружена разработчиком ПО на ранних стадия разработки ПО с использованием методов статического анализа исходных текстов ПО.

С целью уменьшения количества уязвимостей разработчикам ПО рекомендуется внедрять в процессы жизненного цикла основные активности, направленные на разработку безопасного ПО [12-15] – моделирование угроз безопасности информации, статический анализ исходных текстов, тестирование проникновения. Внедрение подобных процедур в практику отечественных разработчиков ПО, на наш взгляд, повысить уровень защищенности создаваемого ПО и, как следствие, значительно уменьшить число инцидентов информационной безопасности.

***Рецензент**: Ловцов Дмитрий Анатольевич, Заслуженный деятель науки Российской Федерации, доктор технических наук, профессор, заведующий кафедрой информационного права, информатики и математики Российского государственного университета правосудия. E-mail: dal-1206@mail.ru*

Литература:

1. Марков А.С., Шеремет И.А. Теоретические аспекты сертификации средств защиты информации // Оборонный комплекс - научно-техническому прогрессу России. 2015. № 4 (128). С. 7-15.
2. Марков. А.С., Цирлов В.Л. Опыт выявления уязвимостей в зарубежных программных продуктах // Вопросы кибербезопасности. 2013. № 1 (1). С. 42-48.
3. Барабанов А.В., Марков А.С., Цирлов В.Л. Оценка соответствия средств защиты информации «Общим Критериям» // Информационные технологии. 2015. Т. 21. № 4. С. 264-270.
4. Марков А.С., Цирлов В.Л., Барабанов А.В. Методы оценки несоответствия средств защиты информации. М.: Радио и связь, 2012. 192 с.
5. Барабанов А.В., Евсеев А.Н. Применение международного стандарта для поиска уязвимостей // Безопасные информационные технологии: Сборник трудов Пятой Всероссийской научно-технической конференции. - М., 2015. - С. 50-52.
6. Барабанов А.В., Евсеев А.Н. Вопросы повышения эффективности анализа уязвимостей при проведении сертификационных испытаний программного обеспечения по требованиям безопасности информации // Труды международного симпозиума Надежность и качество. 2015. Т. 1. С. 330-333.
7. Аветисян А.И., Белеванцев А.А., Чукляев И.И. Технологии статического и динамического анализа уязвимостей программного обеспечения // Вопросы кибербезопасности. 2014. № 3 (4). С. 20-28.
8. Alexander Barabanov, Alexey Markov, Andrey Fadin, and Valentin Tsirlov. 2015. A Production Model System for Detecting Vulnerabilities in the Software Source Code. In Proceedings of the 8th International Conference on Security of Information and Networks (SIN ‹15). ACM, New York, NY, USA, 98-99. DOI: http://dx.doi.org/10.1145/2799979.2800019
9. Markov A.S., Fadin A.A., Tsirlov V.L. Multilevel metamodel for heuristic search of vulnerabilities in the software source code // International Journal of Control Theory and Applications. 2016. V. 9. N 30. P. 313-320.
10. Markov A., Fadin A., Shvets V., Tsirlov V. The experience of comparison of static security code analyzers // International Journal of Advanced Studies. 2015. V. 5. N 3. P. 55-63.
11. Дорофеев А. Тестирование на проникновение: демонстрация одной уязвимости или объективная оценка защищенности? // Защита информации. Инсайд. 2010. № 6 (36). С. 72-73.
12. Барабанов А.В., Дорофеев А.В., Марков А.С., Цирлов В.Л. Семь безопасных информационных технологий/Под. ред. А.С.Маркова. М.: ДМК Пресс, 2017. 224 с.
13. Barabanov A., Markov A., Fadin A., Tsirlov V., Shakhalov I. Synthesis of Secure Software Development Controls. In Proceedings of the 8th International Conference on Security of Information and Networks (Sochi, Russian Federation, September 08-10, 2015). SIN '15. ACM New York, NY, USA, 2015, pp. 93-97. DOI = http://dx.doi.org/10.1145/2799979.2799998.
14. Barabanov A.V., Markov A.S., Tsirlov V.L. Methodological framework for analysis and synthesis of a set of secure software development controls // Journal of Theoretical and Applied Information Technology. 2016. V. 88. N 1. P. 77-88.
15. Howard M., Lipner S. The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software. Microsoft Press, 2006. 352 p

STATISTICS OF SOFTWARE VULNERABILITIES DETECTION DURING CERTIFIED TESTING

Barabanov A.V.[5], Markov A.S.[6], Fadin A.A.[7], Tsirlov V.L.[8]

This paper reviews practical aspects of implementing methods to identify software vulnerabilities in the daily activities of th e accredited test laboratory. It provides the results of approbation of the procedure for detecting vulnerabilities within the framework of the study into the software with an open source–code and software, which is the target of the study during the certification testing, as to the information security requirements. It describes the results of the study to demonstrate the distribution of the identified vulnerabilities according to the types of attacks, country of origin, programming languages used for development, methods of vulnerabilities detection etc. The experience of the foreign certification systems of the information security tools with regard to identification of vulnerabilities of the certifiable software has been analyzed. The main conclusion based on the study is the need in implementing such practices of secure software development in the development lifecycles. Conclusions and recommendations for the test laboratories on implementing methods of the vulnerabilities analysis have been made.

***Keywords**: security software evaluation, vulnerability analysis*

5 Aleksandr Barabanov, Ph.D., NPO Echelon, Moscow, ab@cnpo.ru
6 Aleksey Markov, Dr.Sc., Associate Professor, Bauman MSTU, Moscow, a.markov@bmstu.ru
7 Andrey Fadin, NPO Echelon, Moscow, af@cnpo.ru
8 Valentin Tsirlov, Ph.D., Bauman MSTU, Moscow, v.tsirlov@bmstu.ru

References

1. Markov A.S., SHeremet I.A. Teoreticheskie aspekty sertifikacii sredstv zashchity informacii, Oboronnyj kompleks - nauchno-tekhnicheskomu progressu Rossii. 2015, N 4 (128), pp.7-15.
2. Markov. A.S., Cirlov V.L. Opyt vyyavleniya uyazvimostej v zarubezhnyh programmnyh produktah, Voprosy kiberbezopasnosti. 2013, N 1 (1), pp.42-48.
3. Barabanov A.V., Markov A.S., Cirlov V.L. Ocenka sootvetstviya sredstv zashchity informacii «Obshchim Kriteriyam», Informacionnye tekhnologii. 2015. T. 21, N 4, pp.264-270.
4. Markov A.S., Cirlov V.L., Barabanov A.V. Metody ocenki nesootvetstviya sredstv zashchity informacii. M.: Radio i svyaz', 2012. 192 s.
5. Barabanov A.V., Evseev A.N. Primenenie mezhdunarodnogo standarta dlya poiska uyazvimostej, Bezopasnye informacionnye tekhnologii: Sbornik trudov Pyatoj Vserossijskoj nauchno-tekhnicheskoj konferencii. M., 2015. S. 50-52.
6. Barabanov A.V., Evseev A.N. Voprosy povysheniya ehffektivnosti analiza uyazvimostej pri provedenii sertifikacionnyh ispytanij programmnogo obespecheniya po trebovaniyam bezopasnosti informacii, Trudy mezhdunarodnogo simpoziuma Nadezhnost' i kachestvo. 2015. T. 1, pp.330-333.
7. Avetisyan A.I., Belevancev A.A., CHuklyaev I.I. Tekhnologii staticheskogo i dinamicheskogo analiza uyazvimostej programmnogo obespecheniya, Voprosy kiberbezopasnosti. 2014, N 3 (4), pp.20-28.
8. Alexander Barabanov, Alexey Markov, Andrey Fadin, and Valentin Tsirlov. 2015. A Production Model System for Detecting Vulnerabilities in the Software Source Code. In Proceedings of the 8th International Conference on Security of Information and Networks (SIN ‹15). ACM, New York, NY, USA, 98-99. DOI: http://dx.doi.org/10.1145/2799979.2800019
9. Markov A.S., Fadin A.A., Tsirlov V.L. Multilevel metamodel for heuristic search of vulnerabilities in the software source code, International Journal of Control Theory and Applications. 2016. V. 9. N 30. P. 313-320.
10. Markov A., Fadin A., Shvets V., Tsirlov V. The experience of comparison of static security code analyzers, International Journal of Advanced Studies. 2015. V. 5. N 3. P. 55-63.
11. Dorofeev A. Testirovanie na proniknovenie: demonstraciya odnoj uyazvimosti ili ob»ektivnaya ocenka zashchishchennosti?, Zashchita informacii. Insajd. 2010, N 6 (36), pp.72-73.
12. Barabanov A.V., Dorofeev A.V., Markov A.S., Cirlov V.L. Sem' bezopasnyh informacionnyh tekhnologij/Pod. red. A.S.Markova. M.: DMK Press, 2017. 224 s.
13. Barabanov A., Markov A., Fadin A., Tsirlov V., Shakhalov I. Synthesis of Secure Software Development Controls. In Proceedings of the 8th International Conference on Security of Information and Networks (Sochi, Russian Federation, September 08-10, 2015). SIN '15. ACM New York, NY, USA, 2015, pp. 93-97. DOI = http://dx.doi.org/10.1145/2799979.2799998.
14. Barabanov A.V., Markov A.S., Tsirlov V.L. Methodological framework for analysis and synthesis of a set of secure software development controls, Journal of Theoretical and Applied Information Technology. 2016. V. 88. N 1. P. 77-88.
15. Howard M., Lipner S. The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software. Microsoft Press, 2006. 352 p