ПРАКТИЧЕСКИЕ АСПЕКТЫ ВЫЯВЛЕНИЯ УЯЗВИМОСТЕЙ ПРИ ПРОВЕДЕНИИ СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ ПРОГРАММНЫХ СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ

Вареница В.В.[1], Марков А.С.[2], Савченко В.В.[3], Цирлов В.Л.[4]

Цель: *анализ различных техник и приемы выявления дефектов и уязвимостей при проведении сертификационных испытаний.*

Метод исследования: *компаративный анализ.*

Результат. *сделан вывод об актуальности и приоритетности исследования веб-приложений с открытым исходным кодом. Приведено исследование и показаны недостатки директивных методик выявления уязвимостей и недекларированных возможностей в программных изделиях. Приведена авторская статистика выявленных уязвимостей с детализацией по классам компьютерных атак, производителям средств защиты информации, средам программирования и методикам выявления уязвимостей. Дан сравнительный анализ авторских методик с известными директивными методиками тестирования. Показана актуальность внедрения концепции разработки безопасного программного обеспечения. Даны рекомендации по повышению безопасности программных средств защиты информации.*

Данная работа является развитием более ранней работы двух соавторов этой статьи [22]. В данной работе также выполнено обновление устаревшей статистики по выявленным раннее уязвимостям.

Ключевые слова: *выявления уязвимостей, безопасный программный код, директивные методики, программные закладки, критичные уязвимости.*

DOI:10.21681/2311-3456-2021-5-36-44

Введение

Проблема выявления уязвимостей в программных средствах, конечно, не нова, но она не получила своего исчерпывающего решения, и в настоящее время остается чрезвычайно востребованной [1-3]. Собственно наличие уязвимостей в программном обеспечении составляет основной класс угроз современных компьютерных систем и сетей (см. например [4-10] и др.). Однако, вопросы выявления уязвимостей особенно важны при проведении сертификационных испытаний средств защиты информации (СЗИ), так как данная процедура имеет обязательный характер [11]. Кроме того, выполнение анализа уязвимостей программного обеспечения (ПО) СЗИ сегодня становится одним из основных направлений деятельности в области разработки и поддержки программных изделий в защищенном исполнении [12-13].

Что касается сертификации СЗИ, то указанный 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», так и при испытаниях на соответствие требованиям технических условий или классических руководящих документов. Концептуальный подход к анализу уязвимостей, рекомендуемый 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.

1. Выявление известных (подтвержденных) уязвимостей объекта сертификации. При выполнении данного шага экспертами испытательной лаборатории (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 новыми для отечественных систем сертификации средств защиты

1 Вареница Виталий Викторович, кандидат технических наук, заместитель генерального директора АО «НПО «Эшелон», г. Москва, Россия. E-mail: www@cnpo.ru

2 Марков Алексей Сергеевич, доктор технических наук, старший научный сотрудник, президент АО «НПО «Эшелон», г. Москва, Россия. E-mail: mail@npo-echelon.ru

3 Савченко Владислав Вадимович, руководитель группы АО «НПО «Эшелон, г. Москва, Россия. E-mail: v.savchenko@npo-echelon.ru

4 Цирлов Валентин Леонидович, кандидат технических наук, доцент, доцент МГТУ им. Н.Э.Баумана, Москва, Россия. E-mail: v.tsirlov@bmstu.ru

информации, на текущий момент практически отсутствуют методические документы для ИЛ, которые могли бы использоваться для проведения эффективного анализа уязвимостей веб-приложений. Этим обусловлена актуальность задачи разработки и совершенствования методического обеспечения анализа уязвимостей при проведении сертификационных испытаний по требованиям безопасности информации и при проведении процедуры оценки соответствия требованиям современных стандартов по безопасности.

Адаптированная методика анализа уязвимостей веб-приложений

В рамках выполнения данного исследования была апробирована комбинированная методика анализа уязвимостей ПО, основанная на методиках, предложенных в работа [14, 15] и требованиях современных стандартов ИБ, а также сформулированы рекомендации для экспертов аккредитованных ИЛ.

На рисунках (рис. 1 – рис. 3) представлены основные этапы предлагаемой методики.

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

Этап 1. Получение данных для проведения анализа уязвимостей

Шаг 1. Идентификация минимального набора исходных данных:

- документация на ОО;
- тестовые наборы (test-cases), разработанные производителем для проведения внутреннего тестирования на этапах жизненного цикла ОО.

Шаг 2. Анализ идентифицированных данных. Эксперт на основании тех знаний, что он имеет, должен проанализировать документацию на ОО. Исследование документации позволяет понять, какие технологии и программные средства применялись при проектировании исследуемого изделия, а также сформировать минимальный набор условий, необходимых для корректного функционирования оцениваемого технического средства. Исследование может быть разделено на составляющие, представленные далее по тексту.

1. Поиск идентификационных признаков ОО.
2. Поиск информации о применяемых сторонних технических средствах, необходимых для функционирования ОО.
3. Поиск информации о применяемых заимствованных компонентах ОО.
4. Идентификация перечня конфигураций ОО и сред его функционирования.
5. Идентификация применяемых защитных механизмов ОО.

Шаг 3. Исследование общедоступных источников информации. На основе информации, полученной в ходе выполнения прошлых шагов, эксперт должен проанализировать открытые источники информации (нормативную базу) на предмет наличия шаблонов атак для известных уязвимостей в компонентах ПО среды функционирования и схожих с ОО изделиях, уязвимости конфигураций, применяемых для функционирования ОО, сведения об ошибках при взаимодействии сторонних технических средств с компонентами ОО или схожими изделиями. Результатом выполнения анализа является перечень уязвимостей, определяемых в качестве потенциальных по отношению к ОО.

Шаг 4. 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 средства-

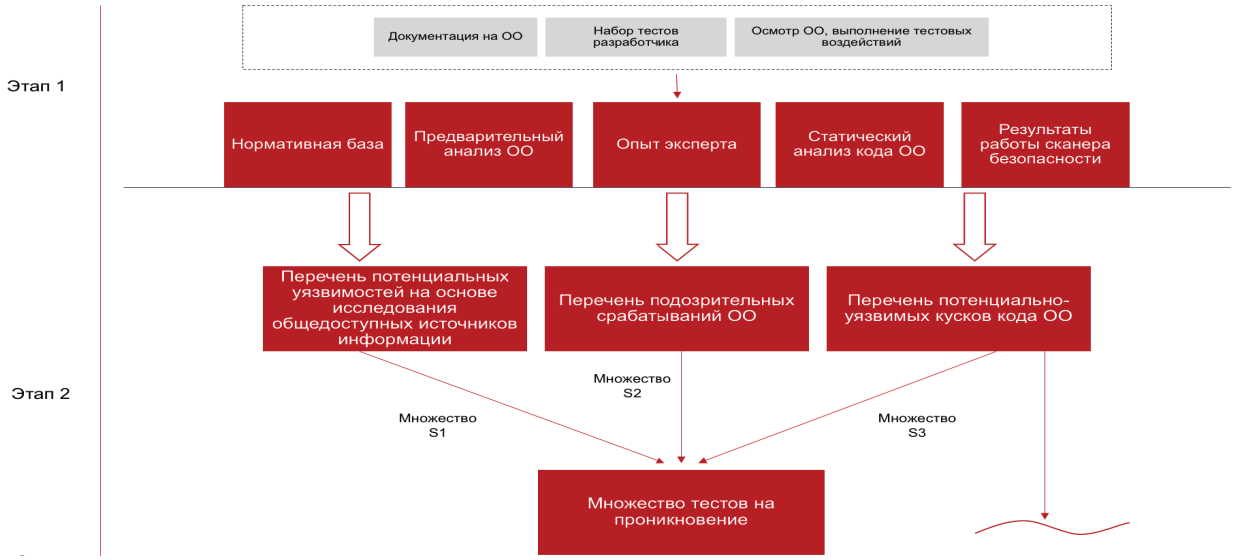

Рис. 1 - Этапы адаптированной методики анализа уязвимостей ПО

ми (мониторы файловой системы, программы аудита файловой системы и пр.). Основная цель данного шага – зафиксировать перечень файлов исходных текстов, участвующих в компиляции объекта сертификации.

Шаг 5. Проведение статического сигнатурного анализа [17, 18] в отношении зафиксированного на шаге 4 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 [19-21].

Шаг 6. Выполнение предварительного анализа ОО. Предварительный анализ выполняется для:

- уточнения сведений, полученных в ходе исследования документации;
- оценки соответствия представленной на испытания документации фактически исследуемому продукту;
- выполнения тестовых воздействий, позволяющих выявить некорректные срабатывания ОО;
- выполнения тестовых воздействий на ОО, позволяющих выделить специальные признаки ОО, свидетельствующие о возможном наличии некоторых разновидностей дефектов кода (уязвимостей);
- определения не описанных в документации дополнительных способов воздействия на ОО, потенциально способных нарушить целостность/доступность/конфиденциальность информации.

Проведение предварительного анализа позволяет эксперту более точно определить назначение изделия, а также понимать, какие из тех навыков, что он имеет, потенциально могут пригодиться при дальнейшем анализе изделия, а также более качественно разработать тесты на проникновение. Данные, получаемые в ходе выполнения предварительного анализа могут повлиять на результаты, полученные в ходе анализа документации на ОО и анализа набора тестов, представленных разработчиком. При исследовании информации, представленной в открытых источниках (нормативной базы), эксперт должен использовать данные, полученные в ходе предварительного анализа, и сформировать дополнительный перечень потенциальных уязвимостей ОО. Также, при проведении предварительного анализа эксперт должен использовать данные, полученные в результате статического анализа кода ОО и результаты работы сканера безопасности. Проводя предварительный анализ, эксперт должен проанализировать результаты работы статического анализатора и сканера безопасности, тем самым минимизировав ложные срабатывания. По результатам проведения предварительного анализа и анализа документации эксперт должен сформировать данные для проведения второго этапа настоящей методики.

Этап 2. Формирование начального перечня потенциальных уязвимостей ОО

Шаг 7. Оценка данных, полученных в ходе этапа 1.

Шаг 8. Формирование перечня потенциальных уязвимостей ОО с использованием результатов выполнения этапа 1.

Шаг 9. Идентификация подозрительных срабатываний ОО. Оценка возможности влияния идентифицированных инцидентов на безопасность веб-приложения.

Шаг 10. Предварительное фаззинг-тестирование. Не интеллектуальный вид фаззинг-тестирования, направленный на получение уточненных входных данных для этапа динамического тестирования.

Шаг 11. Обработка полученного перечня потенциально опасных конструкций с использованием критериев фильтрации, представленных в разделе 6.1.2.1 стандарта ISO/IEC TR 20004.

Шаг 12. Формирование перечня шаблонов атак, являющихся актуальными для исследуемого ПО, с использованием последовательности действий, представленной в разделе 6.1 стандарта ISO/IEC TR 20004.

Этап 13. Формирование пар «потенциальная уязвимость – шаблон атаки». 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 3. Выполнение динамического анализа кода ОО.

Шаг 14. Идентификация участков кода для выполнения инструментации кода.

Шаг 15. Написание инструментирующих и профилирующих функций, вставка инструментирующих и профилирующих функций в код.

Шаг 16. Формирование тестового набора входных данных.

Шаг 17. Выполнение динамического анализа кода. Идентификация подозрительных срабатываний тестируемых функций. Формирование перечня потенциальных уязвимостей.

Шаг 18. Обработка полученного перечня потенциально опасных конструкций с использованием критериев фильтрации, представленных в разделе 6.1.2.1 стандарта ISO/IEC TR 20004. Сравнение с перечнем, представленным на этапе 2 настоящей методики.

Этап 4. Дополнение множества тестов на проникновение с учётом проведённого динамического анализа ОО.

Шаг 19. Формирование перечня шаблонов атак, являющихся актуальными для исследуемого ПО, с использованием последовательности действий, представленной в разделе 6.1 стандарта ISO/IEC TR 20004. Дополнение тестов, полученных в шаге 10 настоящей методики.

Этап 5. Выполнение тестирования на проникновение.

Шаг 20. Анализ тестов, выполненных разработчиками веб-приложения. Оптимизация и доработка тестов.

Шаг 21. Разработка тестов на проникновение на основе сформированного перечня потенциальных уязвимостей и шаблонов атак.

Шаг 22. Монтаж испытательного стенда и проведение тестов на проникновение с использованием разработанных тестов. В случае, если при проведении тестов было установлено, что выявлены новые потенциальные уязвимости, необходимо дополнить перечень тестов на проникновения дополнительными тестами. В случае, если при проведении теста было установлено, что необходимо уточнить тест на проникновение или дополнить его дополнительными действиями, выполнение корректировки теста и повторное выполнение теста.

УДК 004.056

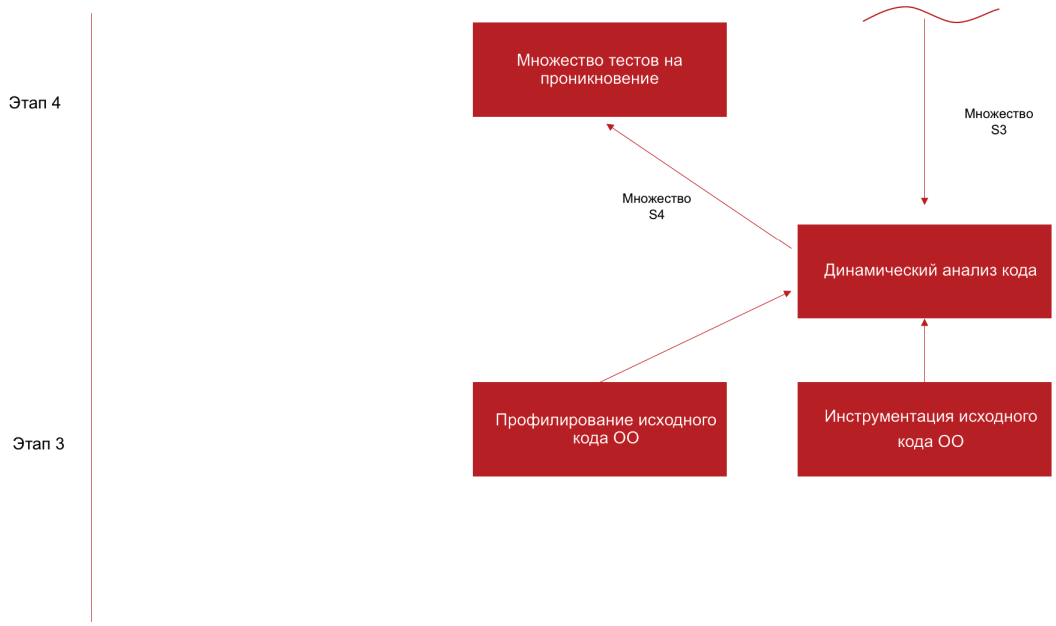

Рис. 2. Этапы адаптированной методики анализа уязвимостей ПО

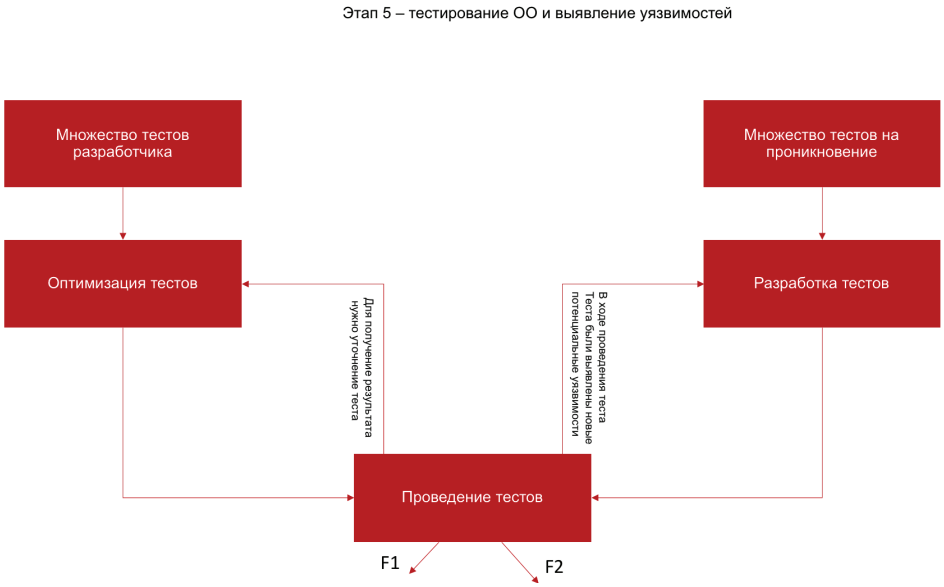

Рис. 3. Этапы адаптированной методики анализа уязвимостей ПО

Шаг 23. Корректировка набора тестов с учетом изменения набора входных данных. Повторение скорректированных тестов при необходимости.

Шаг 24. Определение актуальных уязвимостей ПО по результатам выполнения тестов проникновения и оформление отчетных материалов.

Постановка эксперимента

Экспериментальные исследования адаптированной методики анализа уязвимостей ПО проводились в период с 2019 по 2021 год на научно-исследовательской базе НПО «Эшелон» экспертами аккредитованной ИЛ.

Объектами исследования являлись:

- ПО, проходящее тематические и сертификационные испытания в аккредитованной ИЛ (группа N1, 157 объекта оценки);
- ПО с открытым исходным кодом (группа N2, 91 объект оценки из 157).

При выполнении сигнатурного анализа исходных текстов экспертами ИЛ использовалось средство статического анализа «AppChecker» (разработчик - НПО «Эшелон»). Для проведения тестов на проникновения экспертами ИЛ использовались рекомендации различных тематических ресурсов (CAPEC, OWASP) и инструментальное средство «Сканер-ВС» (разработчик - НПО «Эшелон») [21]. Монтаж и наладка испытательных стен-

DOI:10.21681/2311-3456-2021-5-36-44

дов, 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 N1 экспертами ИЛ было выявлено 235 уязвимостей ПО. Для всех выявленных уязвимостей ПО было получено подтверждение об их актуальности со стороны разработчика ПО. На рисунке 4 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 внедрение SQL-кода и некорректна работа механизмов разграничения доступа. 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.

Часть выявленных уязвимостей была обнаружена в результате исследования исходного кода объектов группы N1 с использованием методов структурного (статического) анализа (рис.5).

Доля уязвимостей, обнаруженных в ПО отечественного производства, значительно больше доли уязвимостей, обнаруженном в зарубежном ПО (рисунок 6). Это объясняется существенными 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 уязви-

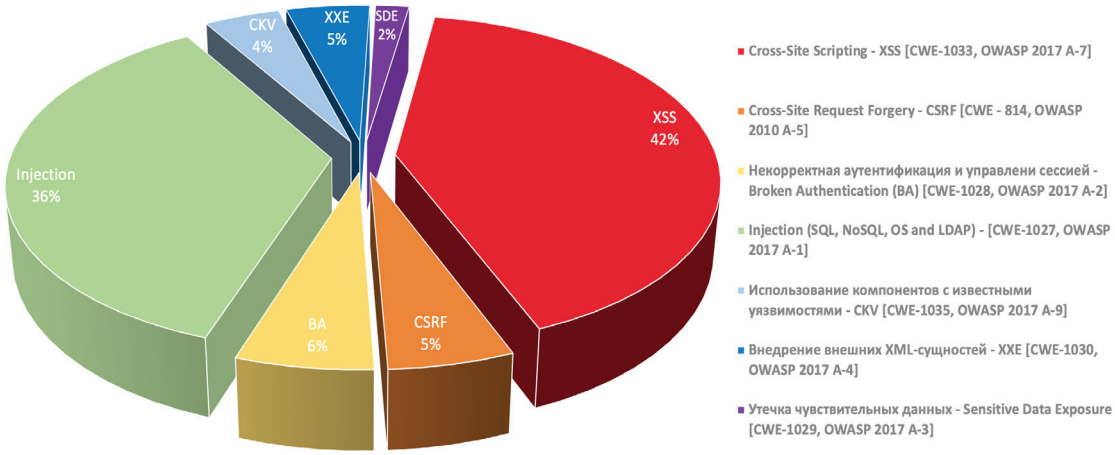

Рис. 4. Распределение выявленных уязвимостей по типам атак на Web-приложения

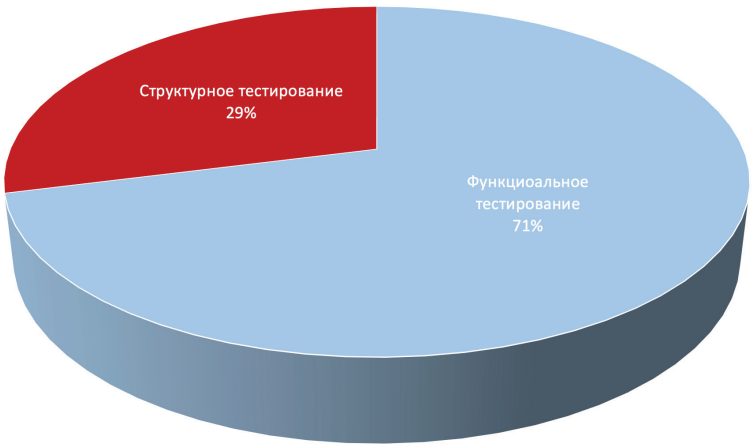

Рис.5. Статистика по методам выявления уязвимостей

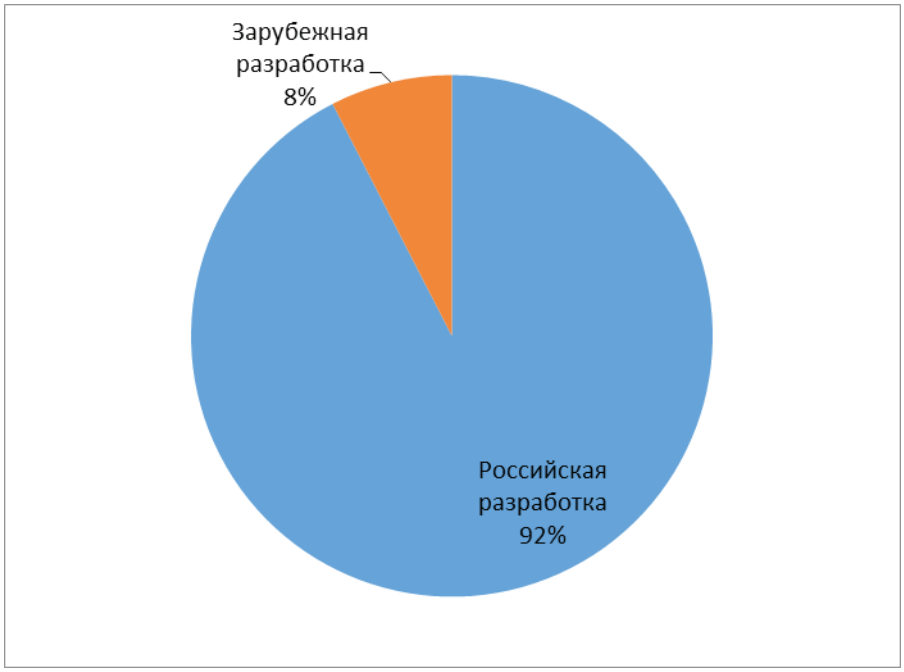

Рис. 6. Статистика по стране происхождения ПО

мостей формировался экспертами ИЛ в условиях отсутствия информации о потенциально опасных конструкция в исходном коде ПО.

Среднее время исправления уязвимости разработчиком ПО составило один месяц.

Следует указать, что современные программные комплексы включают модули программ с открытым кодом. Исследование группы N2 (ПО с открытым исходным кодом) показало, что такие программы тоже содержат уязвимости. В результате проведения исследований было обнаружено 328 дефектов ПО (подтверждены разработчиками ПО), из них 112 – дефекты, приводящие к уязвимости ПО. Дефекты ПО были обнаружены как с использованием статического сигнатурного анализа исходного кода ПО, так и с использованием динамического анализа кода ПО. Распределения обнаруженных уязвимостей по языкам программирования представлены на рис. 6. Наиболее популярными типами обнаруженных дефектов ПО стали ошибки в формировании запросов к СУБД (CWE-89, Improper Neutralization of Special Elements used in an SQL Command) и некорректная работа с входными данными для генерации веб-страниц (CWE-79 Improper Neutralization of Input During Web Page Generation).

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 2019–2021 гг. (выборка – 43 отчета[5]), 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, работающих с веб-приложениями (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 пра-

5 Брались отчёты об изделиях, работающих с веб-приложениями или имеющие в своём составе веб-приложения

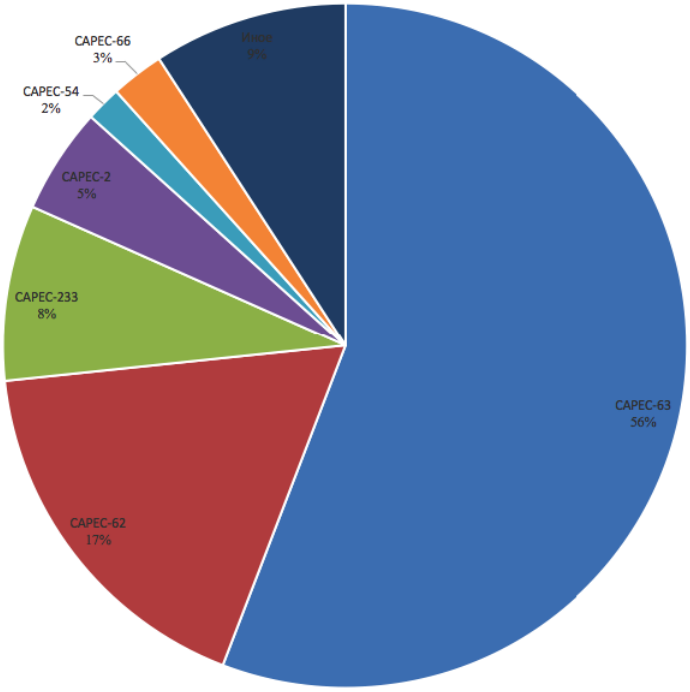

Рис.7. Распределение уязвимостей ПО (ПО с открытым исходным кодом) по типам дефектов CWE

вило, использовались программные средства автоматизации собственной разработки. При этом полноценный динамический анализ не проводился.

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 веб-приложений должен быть первой активностью, выполняемой в рамках сертификационных испытаний, в рамках анализа ПО разработчиком перед выпуском изделия на рынок, и в рамках процедуры оценки соответствия современным стандартам по безопасности, поскольку выявление уязвимостей в объекте оценки на более поздних стадиях (например, после начала сертификационных испытаний, или на этапе поддержки изделия) влечет за собой повторение полного цикла испытаний изделия, большое количество финансовых потерь со стороны разработчика. Следует отметить, что в случае сертификационных испытаний, 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:

- 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 и динамического анализа исходных текстов ПО.

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

Литература

1. Марков А.С., Щербина С.А. Испытания и контроль программных ресурсов // Information Security. 2003. N 6. — С.25.
2. Марков А.С., Миронов С.В., Цирлов В.Л. Выявление уязвимостей в программном коде. Открытые системы. СУБД. 2005. N 12. С. 64-69.
3. Ромашкина Н.П., Марков А.С., Стефанович Д.В. Международная безопасность, стратегическая стабильность и информационные технологии / отв. ред. А.В. Загорский, Н.П. Ромашкина. — М.: ИМЭМО РАН, 2020. С. 98. DOI: 10.20542/978-5-9535-0581-9
4. Барабанов А.В., Марков А.С., Фадин А.А. Оценка возможности выявления уязвимостей программного кода при отсутствии исходных текстов программ // Информационное противодействие угрозам терроризма. 2011. N 16. С. 86-89.
5. Беломестных Д.А., Пушкин П.Ю. Анализ методов выявления уязвимостей веб-приложений // Информационные технологии и проблемы математического моделирования сложных систем. 2017. N 18. С. 5-9.
6. Марков А.С., Миронов С.В., Цирлов В.Л. Выявление уязвимостей программного обеспечения в процессе сертификации // Известия ТРТУ. 2006. N 7 (62). С. 82-87.
7. Сабиров Т.Р., Бирюков Д.Н. Подход к выявлению потенциальных уязвимостей в спецификациях средств защиты информации // Методы и технические средства обеспечения безопасности информации. 2017. N 26. С. 8-10.
8. Сердечный А.Л., Герасимов И.В., Макаров О.Ю., Пастернак Ю.Г., Тихомиров Н.М. Технология выявления сведений об уязвимостях сторонних компонентов программного обеспечения с открытым исходным кодом // Информация и безопасность. 2020. Т. 23. N 3 (4). С. 347-364.
9. Тавасиев Д.А., Команов П.А., Ревазов Х.Ю., Семиков В.С. Анализ методов выявления уязвимостей во встроенном программном обеспечении IoT устройств // Международный научно-исследовательский журнал. 2020. N 1-1 (91). С. 34-37.
10. Марков А.С., Цирлов В.Л., Барабанов А.В. Методы оценки несоответствия средств защиты информации. М.: Радио и связь, 2012. 192 с.
11. Гришин М.И., Марков А.С., Цирлов В.Л. Практические аспекты реализации мер по разработке безопасного программного обеспечения // ИТ-Стандарт. 2019. N 2 (19). С. 29-39.
12. 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.
13. Howard M., Lipner S. The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software. Microsoft Press, 2006. 352 p.
14. 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.
15. Varenitca V. V., Markov A. S., Savchenko V. V. Recommended Practices for the Analysis of Web Application Vulnerabilities. CEUR Workshop Proceedings. 2019. Volume 2603, pp. 75-78.
16. Barabanov A., Markov A., Fadin A., and Tsirlov V. 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
17. Аветисян А.И., Белеванцев А.А., Чукляев И.И. Технологии статического и динамического анализа уязвимостей программного обеспечения // Вопросы кибербезопасности. 2014. N 3 (4). С. 20-28.
18. 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. С. 55-63.
19. Марков А.С., Фадин А.А. Систематика уязвимостей и дефектов безопасности программных ресурсов // Защита информации. Инсайд. 2013. N3. С. 56-61.
20. Бегаев А.Н., Кашин С.В., Маркевич Н.А., Марченко А.А. Выявление уязвимостей и недекларированных возможностей в программном обеспечении — Санкт-Петербург: Университет ИТМО, 2020. — 38 с.
21. Марков А.С. Техническая защита информации. М. АИСНТ. 2020. 234 с.
22. Барабанов А.В., Марков А.С., Фадин А.А., Цирлов В.Л. Статистика выявления уязвимостей программного обеспечения при проведении сертификационных испытаний // Вопросы кибербезопасности. 2017. № 2 (20). С. 2-8. DOI: 10.21681/2311-3456-2017-2-2-8.

PRACTICAL ASPECTS OF VULNERABILITY DETECTION DURING CERTIFICATION TESTS OF INFORMATION SECURITY SOFTWARE

VarenitsaV.V.[6], Markov A.S.[7], Savchenko V.V.[8], Tsirlov V.L.[9]

6 Vitaly Varenitsa, Ph.D., Deputy General Director of JSC «NPO «Echelon», Moscow, Russia. E-mail: www@cnpo. ru

7 Alexey Markov, Dr.Sc., Senior Researcher, President of JSC «NPO «Echelon», Moscow, Russia. E-mail: mail@npo-echelon. ru

8 Vladislav Savchenko, Group Leader of JSC «NPO «Echelon», Moscow, Russia. E-mail: v.savchenko@npo-echelon.ru

9 Valentin Tsirlov, Ph.D., Associate Professor, Associate Professor of Bauman Moscow State Technical University, Moscow, Russia. E-mail: v. tsirlov@bmstu. ru

Abstract

Purpose: *analysis of various techniques and techniques for identifying defects and vulnerabilities during certification tests.*

Research method: comparative analysis.

Result: *the conclusion is made about the relevance and priority of the study of open-source web applications. The study is given and the shortcomings of directive methods for identifying vulnerabilities and undeclared capabilities in software products are shown. The author's statistics of the identified vulnerabilities are given with detailing by classes of computer attacks, manufacturers of information security tools, programming environments and methods for identifying vulnerabilities. A comparative analysis of author's methods with known directive testing methods is given. The relevance of the implementation of the concept of developing secure software is shown. Recommendations on improving the security of software tools for information protection are given.*

Keywords: *vulnerability detection, secure program code, directive methods, program bookmarks, critical vulnerabilities.*

References

1. Markov A.S., Shcherbina S.A. Ispy`taniia i kontrol` programmny`kh resursov // Information Security. 2003. N 6. — S.25.
2. Markov A.S., Mironov S.V., Tcirlov V.L. Vy`iavlenie uiazvimostei` v programmnom kode. Otkry`ty`e sistemy`. SUBD. 2005. N 12. S. 64-69.
3. Romashkina N.P., Markov A.S., Stefanovich D.V. Mezhdunarodnaia bezopasnost`, strategicheskaia stabil`nost` i informatcionny`e tekhnologii / otv. red. A.V. Zagorskii`, N.P. Romashkina. — M.: IME`MO RAN, 2020. S. 98. DOI: 10.20542/978-5-9535-0581-9
4. Barabanov A.V., Markov A.S., Fadin A.A. Ocenka vozmozhnosti vy`iavleniia uiazvimostei` programmnogo koda pri otsutstvii ishodny`kh tekstov programm // Informatcionnoe protivodei`stvie ugrozam terrorizma. 2011. N 16. S. 86-89.
5. Belomestny`kh D.A., Pushkin P.Iu. Analiz metodov vy`iavleniia uiazvimostei` veb-prilozhenii` // Informatcionny`e tekhnologii i problemy` matematicheskogo modelirovaniia slozhny`kh sistem. 2017. N 18. S. 5-9.
6. Markov A.S., Mironov S.V., Tcirlov V.L. Vy`iavlenie uiazvimostei` programmnogo obespecheniia v protcesse sertifikatcii // Izvestiia TRTU. 2006. N 7 (62). S. 82-87.
7. Sabirov T.R., Biriukov D.N. Podhod k vy`iavleniiu potentcial`ny`kh uiazvimostei` v spetcifikatciiakh sredstv zashchity` informatcii // Metody` i tekhnicheskie sredstva obespecheniia bezopasnosti informatcii. 2017. N 26. S. 8-10.
8. Serdechny`i` A.L., Gerasimov I.V., Makarov O.Iu., Pasternak Iu.G., Tihomirov N.M. Tekhnologiia vy`iavleniia svedenii` ob uiazvimostiakh storonnikh komponentov programmnogo obespecheniia s otkry`ty`m ishodny`m kodom // Informatciia i bezopasnost`. 2020. T. 23. N 3 (4). S. 347-364.
9. Tavasiev D.A., Komanov P.A., Revazov KH.Iu., Semikov V.S. Analiz metodov vy`iavleniia uiazvimostei` vo vstroennom programmnom obespechenii IoT ustroi`stv // Mezhdunarodny`i` nauchno-issledovatel`skii` zhurnal. 2020. N 1-1 (91). S. 34-37.
10. Markov A.S., Tcirlov V.L., Barabanov A.V. Metody` ocenki nesootvetstviia sredstv zashchity` informatcii. M.: Radio i sviaz`, 2012. 192 s.
11. Grishin M.I., Markov A.S., Tcirlov V.L. Prakticheskie aspekty` realizatcii mer po razrabotke bezopasnogo programmnogo obespecheniia // IT-Standart. 2019. N 2 (19). S. 29-39.
12. 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.
13. Howard M., Lipner S. The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software. Microsoft Press, 2006. 352 p.
14. Barabanov A.V., Evseev A.N. Primenenie mezhdunarodnogo standarta dlia poiska uiazvimostei`. Bezopasny`e informatcionny`e tekhnologii / Sbornik trudov Piatoi` Vserossii`skoi` nauchno-tekhnicheskoi` konferentcii. M., 2015. S. 50-52.
15. Varenitca V. V., Markov A. S., Savchenko V. V. Recommended Practices for the Analysis of Web Application Vulnerabilities. CEUR Workshop Proceedings. 2019. Volume 2603, pp. 75-78.
16. Barabanov A., Markov A., Fadin A., and Tsirlov V. 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
17. Avetisian A.I., Belevantcev A.A., Chucliaev I.I. Tekhnologii staticheskogo i dinamicheskogo analiza uiazvimostei` programmnogo obespecheniia // Voprosy` kiberbezopasnosti. 2014. N 3 (4). S. 20-28.
18. 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. S. 55-63.
19. Markov A.S., Fadin A.A. Sistematika uiazvimostei` i defektov bezopasnosti programmny`kh resursov // Zashchita informatcii. Insai`d. 2013. N3. S. 56-61.
20. Begaev A.N., Kashin S.V., Markevich N.A., Marchenko A.A. Vy`iavlenie uiazvimostei` i nedeclarirovanny`kh vozmozhnostei` v programmnom obespechenii — Sankt-Peterburg: Universitet ITMO, 2020. — 38 s.
21. Markov A.S. Tekhnicheskaia zashchita informatcii. M. AISNT. 2020. 234 s.
22. Barabanov A.V., Markov A.S., Fadin A.A., Tcirlov V.L. Statistika vy`iavleniia uiazvimostei` programmnogo obespecheniia pri provedenii sertifikatcionny`kh ispy`tanii` // Voprosy` kiberbezopasnosti. 2017. № 2 (20). S. 2-8. DOI: 10.21681/2311-3456-2017-2-2-8.