БЕЗОПАСНОСТЬ В ЧРЕЗВЫЧАЙНЫХ СИТУАЦИЯХ

SAFETY IN EMERGENCY SITUATIONS

УДК 004.05
doi:10.21685/2307-4205-2021-2-10

ПРАКТИЧЕСКИЕ ВОПРОСЫ ПРОВЕДЕНИЯ СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ ПРОГРАММНЫХ ИЗДЕЛИЙ

В. В. Самаров

ООО «16 НИИЦ», Мытищи, Россия
samarov_vladimir@mail.ru

Аннотация. *Актуальность и цели.* Рассматриваются проблемные вопросы, возникающие при проведении сертификационных испытаний программных изделий, предназначенных для обработки конфиденциальной информации, не содержащей сведения, составляющие государственную тайну, в системе сертификации Минобороны России. *Материалы и методы*. При рассмотрении решаемых при проведении сертификационных испытаний задач был сделан акцент на возможность и важность практической реализации связанных с этими задачами операций, успешное выполнение которых в свою очередь представляется необходимым для контроля отсутствия недекларированных возможностей в исследуемом программном изделии. *Результаты и выводы*. По результатам рассмотрения обозначенных проблемных вопросов дано обоснование целесообразности решения рассмотренных задач и актуальности разработки методики проведения соответствующих этапов сертификационных испытаний.

Ключевые слова: сертификационные испытания программных изделий, проблемные вопросы сертификационных испытаний, автоматизация сертификационных испытаний

Для цитирования: Самаров В. В. Практические вопросы проведения сертификационных испытаний программных изделий // Надежность и качество сложных систем. 2021. № 2. С. 99–103. doi:10.21685/2307-4205-2021-2-10

PRACTICAL ISSUES OF CONDUCTING CERTIFICATION TESTS OF SOFTWARE PRODUCTS

V.V. Samarov

LLC "16 NIITS", Mytishchi, Russia
samarov_vladimir@mail.ru

Abstract. *Background*. The article deals with the problematic issues that arise during certification tests of software products intended for processing confidential information that does not contain information constituting a state secret in the certification system of the Ministry of Defense of Russia. *Materials and methods*. When considering the tasks solved during certification tests, an emphasis was made on the possibility and importance of the practical implementation of operations related to these tasks, the successful implementation of which, in turn, seems to be necessary to control the absence of undeclared capabilities in the software product under study. *Results and conclusions*. Based on the results of considering the identified problematic issues, a justification was given for the expediency of solving the considered problems and the relevance of developing a methodology for conducting the corresponding stages of certification tests.

Keywords: certification tests of software products, problematic issues of certification tests, automation of certification tests

For citation: Samarov V.V. Practical issues of conducting certification tests of software products. *Nadezhnost' i kachestvo slozhnykh sistem = Reliability and quality of complex systems.* 2021;2:99–103. (In Russ.). doi:10.21685/2307-4205-2021-2-10

При проведении сертификационных испытаний программных изделий в системе сертификации Минобороны России, по 4-му уровню контроля отсутствия недекларированных возможностей, в соответствии с руководящим документом «Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей» (Гостехкомиссия России, Москва, 1999) (далее – РД НДВ) перед специалистами испытательных лабораторий ставятся следующие задачи:

1) контроль требований по составу и содержанию программной (в т.ч. эксплуатационной) документации;

2) контроль исходного состояния программного обеспечения (ПО);

3) проведение статического анализа исходных текстов ПО, в том числе:

– контроль полноты и отсутствия избыточности исходных текстов на уровне файлов;

– контроль соответствия исходных текстов ПО его объектному (загрузочному) коду.

Практика проведенных работ по сертификационным испытаниям программных изделий в испытательной лаборатории ООО «16 НИИЦ» показала, что при проведении проверок 2–3 зачастую возникают проблемные вопросы, требующие решения. Ниже рассматриваются решаемые при проведении сертификационных испытаний программных изделий задачи и проблемные вопросы, возникающие в контексте выполнения этих задач.

1. Задача по контролю исходного состояния программного изделия.

При решении этой задачи на практике зачастую возникают следующие проблемы:

А. «Потеря» файлов при разархивировании пакетов с исходными текстами[1] и дистрибутивами. Данная ситуация встречается достаточно часто, при осуществлении соответствующих работ в операционных системах семейства Windows (далее – ОС Windows). Это обусловлено тем, что большинство приложений ОС Windows (включая архиваторы и рекомендованные регулятором программы по контролю исходного состояния программных изделий) не умеют работать с длинными путями (рудимент старых версий ОС Windows, где значение параметра[2] Win32 API MAX_PATH = 260).

Решить данную проблему можно, проводя соответствующий этап работ в ОС семейства Linux[3] (где данная проблема как таковая отсутствует), однако здесь зачастую возникает другая проблема, связанная со сложившейся ситуацией, описанной в п. Б).

Б. Отсутствие рекомендованных регулятором[4] средств контроля исходного состояния сертифицируемых программных изделий, функционирующих во всех ОС семейства Linux и имеющих возможность автоматизированного контроля файлов по всем (большинству) используемым алгоритмам контрольного суммирования[5]. Таким образом, в случае, если заявитель сертификационных

[1] Задача корректного разархивирования пакетов с исходными текстами также может быть актуальной при осуществлении работ в рамках статического анализа кода (при выполнении проверок в соответствии с требованиями 1–3 уровней контроля НДВ). Связано это с тем, что большая часть статических анализаторов исходных кодов предназначена для функционирования в ОС Windows. Следует отметить, что в рассмотренном случае (после выполнения статического анализа) дополнительно возникает обратная задача по приведению обработанного пакета с файлами исходных текстов к исходному виду и структуре.

[2] Стоит отметить, что в ОС Windows 10 (начиная с версии 1607) появилась возможность отключить проверку MAX_PATH с помощью групповых политик (**gpedit.msc**) или путем редактирования реестра [1] (команда в PowerShell : *Set-ItemProperty –Path HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem –Name LongPathsEnabled –Value 1*). Вместе с тем выполнение данных действий полностью не решает проблему ввиду того, что проверка длины пути интегрирована в код соответствующих прикладных приложений.

[3] Здесь и далее под ОС семейства Linux будем понимать операционные системы, построенные на базе ОС Linux и имеющие сертификаты соответствия (заключения по безопасности), выданные органом по сертификации Минобороны России.

[4] Стоит отметить, что рекомендованные средства контроля имеются, однако поддержка ими устаревших, а также современных алгоритмов хеширования ограничена (неполная).

[5] На практике разработчики заявляемых на сертификационные испытания программных изделий при проведении соответствующих работ, проведение которых предполагает использование хэш-функций, используют множество алгоритмов контрольного суммирования (от старых, например: CRC8/16/32, ГОСТ 34.11-94, MD-4 и др.; до современных, например: SHA-3, ГОСТ Р 34.11-2012 «Стрибог» (Хэш 256 и 512 бит) и др.).

испытаний использует алгоритм, поддержка которого отсутствует в рекомендованных регулятором средствах анализа, испытательная лаборатория вынуждена решать данную проблему самостоятельно, в том числе путем экспертной верификации алгоритма, используемого заявителем при контрольном суммировании файлов.

2. Задача по идентификации в пакете с исходными текстами бинарных файлов, не имеющих соответствующих им исходных текстов, а также полного (фактического) перечня файлов исходных текстов.

Важность данной задачи обусловлена тем, что:

– в пакетах с исходными текстами, представляемых на сертификационные испытания, недопустимо наличие бинарных файлов различного назначения (библиотеки, драйвера, исполняемые файлы, объектные файлы компилятора и иные бинарные файловые объекты) без присутствия соответствующих им исходных кодов, а также декларирования такого соответствия в программной документации (формально, данное требование можно отнести к требованию, как по контролю полноты, так и по отсутствию избыточности исходных текстов);

– при проведении сертификационных испытаний должны быть проверены все наличествующие в пакете с исходными текстами файлы исходных текстов, а не только те, которые декларированы разработчиком.

Сложность решения данной задачи заключается в том, что встречаются ситуации, когда в пакете с исходными текстами присутствуют файлы без расширений или с расширениями, не соответствующими их действительному типу. Таким образом, формирование (поиск) перечней как бинарных файлов, так и файлов исходных текстов по расширению может не достичь поставленных целей.

Вместе с тем следует отметить, что ввиду того, что данная задача никак не выделена на уровне РД НДВ, то ее решение в принципе не проработано, а методы контроля, носящие рекомендательный характер, отсутствуют.

3. Задача по контролю полноты и отсутствию избыточности исходных текстов на уровне файлов[1] [2].

Данная задача представляет практический интерес при выполнении соответствующего этапа сертификационных испытаний для программных изделий, разработанных на компилируемых языках программирования (в том числе для наиболее широко используемых при разработке программных изделий языках C/C + + /C# и их диалектов, а также на языке Java).

При этом разработчиками используется все многообразие существующих средств разработки (в том числе средств компиляции и сборки пакетов с исходными текстами в дистрибутивные пакеты), а унификация этих средств отсутствует.

В результате на практике при использовании распространенной схемы проведения данного этапа испытаний, при которой необходимо получить информативный протокол сборки[2] соответствующих дистрибутивных пакетов из исходных текстов, зачастую возникают проблемы, требующие для их решения значительных трудозатрат специалистов испытательных лабораторий.

4. Задача по идентификации среды разработки и сборки.

Идентификация среды разработки важна в контексте определения перечня использованных в исследуемом проекте библиотек (как статических, так и динамических). Если по результатам идентификации окажется, что используемая среда разработки не доверенная, т.е. не имеет подтверждения о соответствии требованиям безопасности информации (в том числе и в составе других программных изделий (средств), то при проведении сертификационных испытаний программного изделия необходимо будет к файлам исходных текстов проверяемого изделия добавить файлы исходных текстов соответствующих библиотек[3] (как статических, так и разделяемых (динамических).

[1] Как было отмечено выше, этапом, предшествующим выполнению рассматриваемой задачи, можно считать действия, описанные в п. 2 рассматриваемых задач.

[2] Данная задача была решена с использованием системы аудита ОС Linux [3]. По результатам решения данной задачи был разработан кросс-платформенный программный модуль на языке Питон, который успешно используется специалистами ООО «16 НИИЦ» в повседневной деятельности.

[3] В перечень дополнительно проверяемых файлов должны быть включены файлы исходных текстов библиотек, непосредственно задействованных в исследуемом проекте (подключаемых к создаваемым объектным файлам на этапе линковки (в случае со статическими библиотеками) или необходимыми для функционирования скомпилированных объектных файлов (в части разделяемых библиотек).

Идентификация среды сборки важна из-за того, что средства компиляции и сборки разных версий не могут обеспечить получение одинаковых бинарных целевых файлов, что необходимо при проведении соответствующего этапа сертификации (контроля соответствия исходных текстов их объектному (загрузочному) коду)[1]. Кроме того, при выполнении данного этапа проверок также проверяется происхождение соответствующих утилит, осуществляющих компиляцию и сборку контролируемого программного изделия[2].

На практике основная сложность решения рассматриваемой задачи заключается в определении фактически используемых (задействованных) для исследуемого проекта библиотек и последующая верификация данных библиотек на предмет их происхождения. В дополнение к этому необходимо произвести идентификацию среды компиляции и сборки.

Решение данной задачи из-за отсутствия в ее части строгих требований в РД НДВ в целом не проработано. Вместе с тем, по мнению автора, имеется необходимость в анализе возможных источников получения соответствующих релевантных данных и используемых для этого методах с целью выработки комплексного алгоритма по решению рассматриваемой задачи.

В статье рассмотрены актуальные задачи, решаемые при проведении сертификационных испытаний программных изделий по требованиям РД НДВ по четвертому уровню контроля, и практические проблемы, возникающие при их решении.

Выполнение рассмотренных задач в части деятельности испытательных лабораторий имеет значительный практический интерес и позволяет в случае их успешного решения более эффективно проводить сертификационные испытания[3].

На основании изложенного материала представляется целесообразным:

– проработать варианты решения рассмотренных выше задач и с учетом обозначенных проблемных вопросов разработать методику проведения соответствующих этапов сертификационных испытаний по четвертому уровню;

– рассмотреть возможность автоматизации выполнения соответствующих задач, решаемых на этапах проведения сертификационных испытаний.

Список литературы

1. Ошибка «слишком длинный путь» (path too long) в Windows 10. URL: https://www.geeklib.ru/users/windows10/2018/04/oshibka-slishkom-dlinnyj-put-path-too-long-v-windows-10 (дата обращения: 24.03.2021).
2. Самаров В. В. Использование системы аудита операционных систем семейства Linux при проведении сертификационных испытаний программных изделий // Надежность и качество сложных систем. 2021. № 1. С. 144–150. doi:10.21685/2307-4205-2021-1-14

References

1. *Oshibka «slishkom dlinnyy put'» (path too long) v Windows 10 = Ошибка "слишком длинный путь" (слишком длинный путь) в Windows 10*. (In Russ.). Available at: https://www.geeklib.ru/users/windows10/2018/04/oshibka-slishkom-dlinnyj-put-path-too-long-v-windows-10 (accessed 24.03.2021).
2. Samarov V.V. Use of the Linux operating system audit system when conducting certification tests of software products. *Nadezhnost' i kachestvo slozhnykh system = Reliability and quality of complex systems*. 2021;(1): 144–150. (In Russ.). doi:10.21685/2307-4205-2021-1-14

[1] К среде компиляции и сборки относятся как непосредственно сам компилятор, так и программные средства, управляющие сборкой (например: make, cmake, maven, ant и др.), а также утилиты, осуществляющие упаковку/архивирование скомпилированных бинарных файлов проекта в соответствующие пакеты (например: deb, rpm, cpio, tar и др.).

[2] В идеале используемые средства компиляции и сборки должны быть сертифицированы по требованиям безопасности информации. Минимальное требование к таким средствам – фиксация в архиве предприятия разработчика как непосредственно исполняемых файлов этих средств, так и всех соответствующих им файлов исходных текстов (на этапе проведения сертификационных испытаний для этих средств проводится контрольная сборка с целью установления соответствия заложенных в архив исходных текстов, соответствующим им загрузочным модулям).

[3] Под эффективностью в данном контексте понимается возможность формирования обоснованного заключения по результатам контроля требований РД НДВ, с учетом рассмотренных задач (в том числе и задач, отсутствующих в явном виде в РД НДВ).

Информация об авторах / Information about the authors

Владимир Владимирович Самаров
заместитель начальника испытательной лаборатории,
ООО «16 НИИЦ»
(Россия, Московская обл., г. Мытищи,
Олимпийский просп., 29, вл. 2, 7А-4)
E-mail: samarov_vladimir@mail.ru

Vladimir V. Samarov
Deputy head of test laboratory,
LLC "16 NIITS "
(7A-4, 2, 29 Olimpiyskiy avenue, Mytischi,
Moscow region, Russia)

Авторы заявляют об отсутствии конфликта интересов /
The authors declare no conflicts of interests.

Поступила в редакцию/Received 17.04.2021
Поступила после рецензирования/Revised 27.04.2021
Принята к публикации/Accepted 01.05.2021