ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56939— 2024

Защита информации

РАЗРАБОТКА БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Общие требования

Издание официальное

Москва
Российский институт стандартизации
2024

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Федеральной службой по техническому и экспортному контролю (ФСТЭК России), Акционерным обществом «Лаборатория Касперского» (АО «Лаборатория Касперского»), Федеральным государственным бюджетным учреждением науки Институт системного программирования им. В.П. Иванникова Российской академии наук (ИСП РАН), Акционерным обществом «Информационные технологии и коммуникационные системы» (АО «ИнфоТеКС»), Акционерным обществом «Позитив Текнолоджиз» (АО «Позитив Текнолоджиз»), Обществом с ограниченной ответственностью «РусБИТех-Астра» (ООО «РусБИТех-Астра»), Акционерным обществом «Сбербанк-Технологии» (АО «СберТех»), Обществом с ограниченной ответственностью Научно-технический центр «Фобос-НТ» (ООО НТЦ «Фобос-НТ»), Обществом с ограниченной ответственностью «Центр безопасности информации» (ООО «ЦБИ»), Акционерным обществом «Научно-производственное объединение «Эшелон» (АО НПО «Эшелон»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 362 «Защита информации»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 24 октября 2024 г. № 1504-ст

4 ВЗАМЕН ГОСТ Р 56939—2016

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Оформление. ФГБУ «Институт стандартизации», 2024

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины и определения 2
4 Общие требования к разработке безопасного программного обеспечения 4
5 Процессы разработки безопасного программного обеспечения 5
Приложение А (справочное) Инициализация процессов разработки безопасного программного обеспечения 28

Введение

Настоящий стандарт направлен на достижение целей, связанных с предотвращением появления, выявлением и устранением недостатков, в том числе уязвимостей, в программном обеспечении, и содержит общие требования, предъявляемые к разработчикам и производителям программного обеспечения при реализации процессов разработки безопасного программного обеспечения.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Защита информации

РАЗРАБОТКА БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Общие требования

Information protection. Secure software development. General requirements

Дата введения — 2024—12—20

1 Область применения

Настоящий стандарт устанавливает общие требования к содержанию и порядку выполнения работ, связанных с созданием безопасного программного обеспечения (ПО) и устранением выявленных недостатков, в том числе уязвимостей, ПО.

Настоящий стандарт предназначен для разработчиков и производителей ПО, а также для организаций, выполняющих оценку соответствия процессов разработки ПО положениям настоящего стандарта.

Примечание — В настоящем стандарте разработчики и производители ПО обозначены термином «разработчик(и)».

Настоящий стандарт предусматривает применение в комплексе с другими национальными стандартами по разработке безопасного ПО, в которых раскрываются вопросы внедрения и оценки соответствия положениям настоящего стандарта, задаются требования к отдельным технологиям, применяемым в процессах разработки.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р ИСО/МЭК 12207 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств

ГОСТ Р 50922 Защита информации. Основные термины и определения

ГОСТ Р 58412 Защита информации. Разработка безопасного программного обеспечения. Угрозы безопасности информации при разработке программного обеспечения

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

Издание официальное

3 Термины и определения

В настоящем стандарте применены термины по ГОСТ Р 50922, а также следующие термины с соответствующими определениями:

3.1 **безопасное программное обеспечение:** Программное обеспечение, разработанное в ходе реализации совокупности процессов (мер), направленных на предотвращение появления и устранение недостатков программы.

3.2 **динамический анализ кода программы:** Вид работ по инструментальному исследованию программы, основанный на анализе кода программы в режиме непосредственного исполнения (функционирования) кода.

3.3 **документация разработчика программного обеспечения:** Совокупность программных и иных документов, предназначенных для организации и проведения работ по созданию программного обеспечения и/или подтверждения соответствия требованиям настоящего стандарта.

Примечание — К программным относятся документы, содержащие сведения, необходимые для разработки, изготовления, сопровождения и эксплуатации программ. К иным документам относятся документы, не относящиеся к программным, и содержащие сведения, подтверждающие соответствие требованиям настоящего стандарта.

3.4

инструментальное средство: Компьютерная программа, используемая как средство разработки, тестирования, анализа, производства или модификации других программ или документов на них.

[ГОСТ Р 51904—2002, пункт 3.17]

3.5 **интерфейс программного обеспечения:** Способ взаимодействия между программами или между программами и пользователями.

3.6 **компонент программного обеспечения:** Составная часть (программный модуль) программного обеспечения, выполняющая определенную функцию.

3.7

компьютерная атака: Целенаправленное несанкционированное воздействие на информацию, на ресурс автоматизированной информационной системы или получение несанкционированного доступа к ним с применением программных или программно-аппаратных средств.

[ГОСТ Р 51275—2006, пункт 3.11]

3.8 **недостаток программы:** Любое несоответствие программы заданным требованиям или любая ошибка, допущенная в ходе проектирования или реализации программы, которая в случае ее неисправления может являться причиной невозможности выполнения требуемых функциональных возможностей или уязвимости программы.

3.9

поверхность атаки: Множество подпрограмм (функций, модулей) программного обеспечения, обрабатывающих данные, поступающие посредством интерфейсов, напрямую или косвенно подверженных риску атаки.

[адаптировано из ГОСТ Р 56498—2015, пункт 3.1.7]

3.10 **пользователь (программного обеспечения):** Лицо, применяющее программное обеспечение.

3.11

программа: Данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма.

[ГОСТ 19781—90, статья 1]

3.12

программное обеспечение: Совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ.

[ГОСТ 19781—90, статья 2]

3.13

программный модуль (модуль программного обеспечения): Программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

[ГОСТ 19781—90, статья 15]

3.14 **сборка программного обеспечения:** Процесс построения из исходного кода программ программных модулей, готовых к выполнению или интерпретации, и/или библиотек.

3.15

сборочная среда: Совокупность программных и аппаратных средств, служб связи, интерфейсов, форматов данных, протоколов, стандартов, обеспечивающих преобразование исходного кода программ в программные модули в соответствии с представленными метаданными и с учетом зависимостей программного модуля.

[Адаптировано из ГОСТ Р 54593—2011, пункт 3.13]

3.16 **система сборки программного обеспечения:** Совокупность программных средств и конфигурационных файлов, которая позволяет выполнить сборку конкретного экземпляра конкретной программы.

3.17

среда разработки программного обеспечения: Интегрированная система, включающая в себя аппаратные средства, программное обеспечение, программно-аппаратные средства, процедуры и документы, необходимые для разработки программного обеспечения.

[ГОСТ Р 51904—2002, пункт 3.62]

3.18 **статический анализ исходного кода программы:** Вид работ по инструментальному исследованию программы, основанный на анализе исходных текстов программы с использованием специализированных инструментальных средств (статических анализаторов) в режиме, не предусматривающем исполнения кода, и выполняемый для определения свойств программы; в частности, статический анализ применяется для выявления потенциальных ошибок в программе.

Примечание — Термины «анализ исходных текстов» и «анализ исходного кода» взаимозаменяемы в силу устоявшейся профессиональной терминологии.

3.19 **управление конфигурацией программного обеспечения:** Скоординированные действия, направленные на формирование и контроль конфигурации программного обеспечения.

3.20 **уязвимость программы:** Недостаток программы, который может быть использован для реализации угроз безопасности информации.

Примечание — Уязвимость программы может быть результатом ее разработки без учета требований по обеспечению безопасности информации, вследствие наличия ошибок проектирования или реализации или в результате применения небезопасных инструментальных средств и языков программирования, имеющих ошибки проектирования, реализации или конфигурирования.

3.21 **функциональное тестирование программы:** Вид работ по исследованию программы, направленный на выявление отличий между ее реально существующими и требуемыми свойствами.

3.22 **фаззинг-тестирование программы:** Вид работ по исследованию программы, основанный на передаче программе случайных или специально сформированных входных данных, отличных от данных, предусмотренных алгоритмом работы программы.

3.23 **экспертиза исходного кода программы:** Вид работ по выявлению недостатков программы в исходном коде программы, направленный на оценку ее свойств и основанный на анализе исходного кода программы в режиме, не предусматривающем реального выполнения кода.

4 Общие требования к разработке безопасного программного обеспечения

4.1 Основной целью разработки безопасного ПО является:

- выявление недостатков, в том числе уязвимостей, в разрабатываемом ПО;
- снижение количества недостатков, в том числе уязвимостей ПО;
- снижение ущерба от невыявленных уязвимостей ПО;
- оперативное устранение выявляемых уязвимостей в ПО.

4.2 В настоящем стандарте общие требования к разработке безопасного ПО представлены в виде описания процессов, реализация которых направлена на достижение цели разработки безопасного ПО.

4.3 Поскольку модель жизненного цикла ПО зависит от специфики, масштаба, сложности ПО и условий, в которых ПО создается и функционирует, приведенные в настоящем стандарте процессы намеренно не связываются с конкретной моделью жизненного цикла ПО. По тексту стандарта процессы могут быть соотнесены с обобщенными этапами жизненного цикла ПО (например, с этапом эксплуатации), что дает разработчику гибкость реализации процессов и используемых подходов, таких как изложенных в ГОСТ Р ИСО/МЭК 12207.

4.4 Методика оценки соответствия реализации процессов разработки безопасного ПО установленным настоящим стандартом требованиям является предметом рассмотрения отдельного национального стандарта.

4.5 Внедрение и реализация процессов разработки безопасного ПО подразумевают непосредственное участие руководства разработчика и выделение необходимых ресурсов.

4.6 Процессы разработки безопасного ПО реализуются комплексом организационных и технических мероприятий.

4.7 Условия реализации процессов разработки безопасного ПО выражены в форме следующих понятий: требование, рекомендация или допустимое действие. С целью подчеркнуть различие между разными формами условий для реализации процессов разработки безопасного ПО в настоящем стандарте используются вспомогательные глаголы «должен», «следует» «рекомендуется» и «может». Глаголы «должен» и «следует» — для выражения условия, требуемого для соответствия требованию; «рекомендуется» — для выражения рекомендации по реализации, «может» — для того чтобы отразить возможные направления допустимых действий.

4.8 Процессы разработки безопасного ПО изложены в виде следующей структуры: наименование процесса (наименование соответствующего подраздела раздела 5), цели, требования к реализации, артефакты реализации требований (пункты соответствующих подразделов).

4.9 Пункт «Цели» включает формулировки целей реализации процесса разработки безопасного ПО.

4.10 Пункт «Требования к реализации» включает формулировки требований к реализации процесса разработки безопасного ПО.

4.11 Пункт «Артефакты реализации требований» включает формулировки наименования артефактов, подтверждающих выполнение требований к реализации процессов разработки безопасного ПО, и их содержание. Под артефактами реализации требований в настоящем стандарте понимается любая информация (документы, отчеты, файлы, журналы, результаты работы инструментов, процессов), сохраненная в любом виде (электронном, физическом), на основании которой возможно подтвердить реализацию соответствующих требований.

4.12 Наиболее общими для большинства рассматриваемых процессов артефактами реализации требований разработчика в настоящем стандарте являются регламенты. В общем случае регламент осуществления процесса разработки безопасного ПО должен содержать информацию об обязанностях сотрудников и их ролях при реализации соответствующих процессов, а также информацию, непосредственно относящуюся к особенностям реализации процесса. Требований к оформлению регламентов не предъявляется. Регламенты, относящиеся к различным процессам разработки безопасного ПО, могут быть оформлены как самостоятельные документы в электронном или физическом виде или объединены в рамках общего документа (например, в виде руководства по разработке безопасного ПО).

4.13 При разработке ПО средой разработки ПО должны быть обеспечены контроль версий разрабатываемого ПО, непрерывная интеграция разрабатываемого ПО, управление задачами, в том числе по отслеживанию ошибок в коде ПО. Процессы разработки безопасного ПО необходимо интегрировать с применяемыми в среде разработки ПО системами в целях обеспечения регулярности и своевременности проверок кода, прослеживаемости устранения выявленных ошибок.

Примечание — Под непрерывной интеграцией здесь понимается автоматизация внесений изменений кода ПО (модулей ПО, компонентов ПО) от нескольких разработчиков в единый проект ПО.

4.14 Конкретная совокупность процессов разработки безопасного ПО, подлежащая реализации разработчиком ПО, определяется требованиями нормативных правовых актов, национальных и отраслевых стандартов, технических заданий на выполнение научно-исследовательских и опытно-конструкторских работ, иными документами. В случае, когда в соответствующих документах определена необходимость соответствия настоящему стандарту, обязательной реализации подлежат все требования стандарта, за исключением рекомендуемых к реализации требований, в формулировках таких требований используются вспомогательные глаголы «рекомендуется» и «может».

4.15 Предъявление требований настоящего стандарта к ПО, разрабатываемому в рамках научно-исследовательских и опытно-конструкторских работ, возможно только в форме явного перечисления в техническом задании процессов, подлежащих реализации. Допускается предъявлять требования настоящего стандарта не в полном объеме, указывать условия применимости каждого из реализуемых процессов, требований к ним, состава и содержания артефактов реализации требований.

4.16 В приложении А в качестве справочной информации приведено описание инициализации процессов разработки безопасного ПО. Оценка указанных процессов не является обязательной при внешнем контроле реализации (аудите).

4.17 В ходе реализации процессов разработки безопасного ПО должны быть реализованы меры по защите информации, относящейся к этим процессам, — разграничение доступа к результатам исследований и тестирования, обеспечение защищенного хранения соответствующей информации и антивирусная защита. Меры защиты информации для процессов разработки безопасного ПО могут быть включены в регламенты реализации соответствующих процессов разработки безопасного ПО или содержаться в отдельном документе разработчика.

5 Процессы разработки безопасного программного обеспечения

5.1 Планирование процессов разработки безопасного программного обеспечения

5.1.1 Цели

5.1.1.1 Обеспечение потребностей в ресурсах, необходимых для реализации процессов разработки безопасного ПО.

5.1.1.2 Подготовка и планирование внедрения и улучшения процессов разработки безопасного ПО.

5.1.1.3 Определение области применения процессов разработки безопасного ПО.

5.1.2 Требования к реализации

5.1.2.1 Выполнять периодический анализ текущего статуса реализации процессов разработки безопасного ПО.

5.1.2.2 Выполнять периодический анализ потребностей в ресурсах, необходимых для реализации процессов разработки безопасного ПО.

5.1.2.3 Разрабатывать план развития процессов разработки безопасного ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.1.3.1, 5.1.3.2.

5.1.2.4 Разрабатывать план реализации процессов разработки безопасного ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.1.3.1, 5.1.3.2.

5.1.2.5 Определять область применения процессов разработки безопасного ПО.

5.1.3 Артефакты реализации требований

5.1.3.1 Результаты анализа текущего статуса реализации процессов разработки безопасного ПО должны содержать следующие сведения:

- перечень процессов разработки безопасного ПО, реализованных и не реализованных разработчиком;
- результаты определения достаточности и соответствия процессов разработки безопасного ПО, реализованных разработчиком, положениям настоящего стандарта и иным стандартам, содержащим требования к разработке безопасного ПО, используемым инструментам и технологиям.

5.1.3.2 Результаты анализа потребностей в ресурсах, необходимых для реализации процессов разработки безопасного ПО, могут содержать оценочные показатели в материальных и людских ресурсах для каждого реализуемого или планируемого к реализации процесса разработки безопасного ПО.

5.1.3.3 План развития процессов разработки безопасного ПО должен содержать порядок (очередность) внедрения процессов разработки безопасного ПО с учетом приоритетов разработчика и имеющихся ресурсов, планируемые изменения в организационно-штатной структуре разработчика, планируемые закупки необходимых инструментов, затраты на обучение и др.

Примечание — План развития процессов разработки безопасного ПО может разрабатываться и быть представлен в системе управления задачами.

5.1.3.4 План реализации процессов разработки безопасного ПО должен содержать цели, сроки и этапы внедрения процессов разработки безопасного ПО; перечень необходимых ресурсов; информацию об ответственных за внедрение процессов сотрудниках.

Примечание — План реализации процессов разработки безопасного ПО может разрабатываться и быть представлен в системе управления задачами.

5.1.3.5 Описание области применения процессов разработки безопасного ПО должно содержать состав ПО (версии, модули, компоненты, функциональные подсистемы и т. п.), в отношении которого должны быть реализованы процессы разработки безопасного ПО, с обоснованием выбора указанного состава ПО.

5.2 Обучение сотрудников

Примечание — В данном подразделе под обучением понимается совокупность методов и подходов, направленных на постоянное повышение квалификации, развитие профессиональных навыков, знаний и компетенций сотрудников разработчика, реализуемых как с привлечением сторонних организаций, так и самим разработчиком.

5.2.1 Цели

5.2.1.1 Получение актуальной информации о существующих (доступных для анализа) практиках, документах, обучающих курсах и тренингах по разработке безопасного ПО.

5.2.1.2 Организация обучения сотрудников типовым практикам разработки безопасного ПО с учетом актуальных потребностей.

5.2.1.3 Создание условий для снижения количества возможных типовых ошибок и уязвимостей в разрабатываемом ПО.

5.2.2 Требования к реализации

5.2.2.1 Проводить анализ существующих (доступных для анализа) практик, документов, обучающих курсов и тренингов по разработке безопасного ПО.

Входными данными требования к реализации являются существующие (доступные для анализа) практики, документы, обучающие курсы и тренинги по разработке безопасного ПО.

5.2.2.2 Разрабатывать план обучения с учетом потребностей разработчика в части используемых средств и технологий разработки.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.2.3.1.

5.2.2.3 Проводить обучение сотрудников.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.2.3.2.

5.2.2.4 Вести учет обучения сотрудников.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.2.3.2, 5.2.3.3, 5.2.3.6.

5.2.2.5 Определить критерии пересмотра программ обучения (курсов, тренингов и т. п.).

5.2.2.6 Повышать осведомленность сотрудников разработчика о возможных типовых угрозах, ошибках и уязвимостях в разрабатываемом ПО, механизмах их недопущения или минимизации вероятности их возникновения, порядке сопровождения ПО и управления жизненным циклом.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.2.3.1.

5.2.3 Артефакты реализации требований

5.2.3.1 Результаты анализа существующих (доступных для анализа) практик, документов, обучающих курсов и тренингов по разработке безопасного ПО с точки зрения их применимости для обучения сотрудников разработчика.

5.2.3.2 План обучения, включающий:

- список сотрудников, направляемых на обучение;
- сроки прохождения обучения;
- наименование программы (курса, тренинга) обучения;
- ожидаемый результат обучения.

5.2.3.3 Артефакты реализации требований, подтверждающие прохождение обучения, включают (в зависимости от учебной программы, курса) свидетельства, дипломы, отчеты обучающих платформ и иные документы и материалы, подтверждающие прохождение сотрудником обучения.

5.2.3.4 Артефакты реализации требований, подтверждающие осуществление учета обучения сотрудников, должны содержать информацию о сотрудниках, прошедших обучение, пройденных программах (курсах) и результатах прохождения обучения.

5.2.3.5 Критерии необходимости пересмотра программ обучения (курсов, тренингов и т. п.) должны содержать информацию о периодичности пересмотра (уточнения) программ обучения (курсов, тренингов и т. п.) или о событиях, при наступлении которых необходимо изменение программ обучения (курсов, тренингов и т. п.).

5.2.3.6 Артефакты реализации требований, подтверждающие осуществление повышения осведомленности сотрудников, должны содержать информацию о проведенных мероприятиях по повышению осведомленности сотрудников о возможных типовых угрозах, ошибках и уязвимостях в разрабатываемом ПО, механизмах их недопущения или минимизации вероятности их возникновения, порядке сопровождения ПО и управления жизненным циклом, а также о сотрудниках (подразделениях), для которых проводились мероприятия по повышению осведомленности.

5.3 Формирование и предъявление требований безопасности к программному обеспечению

5.3.1 Цели

5.3.1.1 Обеспечение безопасности ПО посредством предъявления к нему требований и управления требованиями в процессе изменения (разработки) ПО.

5.3.2 Требования к реализации

5.3.2.1 Разработать регламент управления требованиями безопасности ПО.

5.3.2.2 Предъявлять к ПО требования безопасности.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.3.3.1.

5.3.2.3 Вести учет предъявленных требований безопасности и контроль однозначности трактования и непротиворечивости набора требований безопасности ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.3.3.2.

Примечание — Критерии однозначности трактования и непротиворечивости набора требований безопасности ПО определяются разработчиком экспертным методом.

5.3.2.4 Осуществлять пересмотр набора требований безопасности на основе выполнения критериев пересмотра — с установленной периодичностью или при наступлении определенных событий.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.3.3.1, 5.3.3.2.

5.3.3 Артефакты реализации требований

5.3.3.1 Регламент управления требованиями безопасности ПО должен включать следующие положения:

- порядок предъявления требований безопасности ПО;
- порядок предоставления требований безопасности ПО исполнителям;
- порядок отслеживания процесса предоставления, получения и выполнения требований безопасности ПО;
- критерии пересмотра требований безопасности ПО (периодически, при наступлении определенных событий).

5.3.3.2 Набор требований безопасности ПО должен содержать следующую информацию:

- идентификатор требования безопасности ПО;
- формулировку требования безопасности ПО;
- дату предъявления требований безопасности ПО;

- приоритет/важность требования безопасности ПО;
- предполагаемые сроки реализации;
- сведения о сотрудниках (подразделениях), предъявивших требования;
- сведения о сотрудниках (подразделениях), принявших требования к реализации.

5.3.3.3 Артефакты реализации требований, подтверждающие осуществление учета предъявленных требований безопасности ПО, должны включать, как минимум, следующую информацию:

- сведения о принятии требований к реализации, подтверждающие однозначность трактования и непротиворечивость набора требований безопасности ПО;
- текущий статус реализации требований;
- сведения об изменениях статуса реализации предъявленных требований безопасности ПО;
- сведения об изменениях предъявленных требований безопасности ПО.

Примечание — Управление требованиями безопасности ПО рекомендуется осуществлять с использованием средств автоматизации (например, системы управления изменениями, системы управления задачами и т. п.).

5.3.3.4 Набор требований безопасности ПО, уточненный по результатам выполнения требований 5.3.2.4, должен содержать информацию об особенностях реализации требований безопасности ПО в процессе разработки ПО, принятых решениях по корректировкам требований безопасности ПО в процессе разработки.

5.4 Управление конфигурацией программного обеспечения

5.4.1 Цели

5.4.1.1 Осуществление уникальной идентификации ПО, документации на ПО, других элементов, подлежащих отслеживанию в рамках управления конфигурацией ПО (элементов конфигурации).

5.4.1.2 Контроль реализации изменений ПО, документации на ПО, других элементов, подлежащих отслеживанию в рамках управления конфигурацией ПО (элементов конфигурации).

5.4.2 Требования к реализации

5.4.2.1 Разработать регламент управления конфигурацией в рамках жизненного цикла ПО.

5.4.2.2 Определить перечень элементов конфигурации, подлежащих отслеживанию в рамках управления конфигурацией ПО.

5.4.2.3 Контролировать реализацию изменений ПО, документации на ПО, других элементов, подлежащих отслеживанию в рамках управления конфигурацией ПО (элементов конфигурации).

5.4.3 Артефакты реализации требований

5.4.3.1 Регламент управления конфигурацией ПО должен содержать:

- порядок формирования перечня элементов ПО (компонентов, модулей и т. п.), документации на ПО, подлежащих отслеживанию в рамках жизненного цикла ПО;
- порядок идентификации ПО (версий ПО, модулей ПО), документации для отслеживаемых элементов.

5.4.3.2 Перечень элементов конфигурации, подлежащих отслеживанию в рамках управления конфигурацией ПО.

5.4.3.3 Артефакты реализации требований, подтверждающие реализацию управления изменениями ПО, документации на ПО в рамках управления конфигурацией ПО, должны отображать факты изменения ПО (модулей ПО), документации на ПО.

5.5 Управление недостатками и запросами на изменение программного обеспечения

5.5.1 Цели

5.5.1.1 Обеспечение управления недостатками ПО.

5.5.1.2 Обеспечение управления запросами на изменение ПО.

Примечание — Управление недостатками и запросами на изменение ПО способствует систематическому устранению ошибок программирования, отклонений от заданных требований и корректировке требований в необходимых случаях путем осуществления запросов на изменение ПО — предложений о добавлении, модификации или удалении каких-либо элементов (модулей, компонентов, функциональных возможностей) ПО.

5.5.2 Требования к реализации

5.5.2.1 Разработать регламент управления недостатками ПО.

5.5.2.2 Разработать регламент управления запросами на изменение ПО.

5.5.2.3 Контролировать реализацию изменений, связанных с недостатками ПО.

5.5.2.4 Контролировать реализацию запросов на изменение в рамках жизненного цикла ПО.

5.5.2.5 Использовать средства автоматизации для управления недостатками и запросами на изменение разрабатываемого ПО.

Примечание — В качестве средств автоматизации рекомендуется использовать системы управления изменениями, системы управления задачами, системы контроля версий и т. п. При этом рекомендуется обеспечивать взаимосвязь (перекрестные ссылки) между такими системами при исправлении недостатков.

5.5.3 Артефакты реализации требований

5.5.3.1 Регламент управления недостатками ПО должен содержать:

- порядок идентификации недостатков ПО;
- порядок управления недостатками ПО, включающий сведения о действиях, выполняемых при выявлении, устранении, тестировании, принятии решения об окончании работы с недостатком (закрытии недостатка).

5.5.3.2 Регламент управления запросами на изменение ПО должен содержать:

- порядок идентификации запросов на изменение ПО;
- порядок управления запросами на изменение ПО, включающий сведения о действиях, выполняемых при осуществлении запроса на изменение, тестировании, принятии решения о закрытии запроса на изменение.

5.5.3.3 Артефакты реализации требований, подтверждающие реализацию управления недостатками ПО, должны содержать зафиксированные факты изменений, связанных с недостатками, включающие следующую информацию:

- уникальный идентификатор недостатка ПО;
- описание недостатка ПО;
- версию ПО (модуля ПО, компонента ПО), к которому относится недостаток ПО;
- приоритет выполнения действий с недостатком ПО;
- текущий статус обработки изменений, связанных с недостатками ПО.

5.5.3.4 Артефакты реализации требований, подтверждающие реализации управления запросами на изменение ПО, должны содержать следующую информацию:

- уникальный идентификатор запроса на изменение ПО;
- краткую характеристику запроса на изменение ПО;
- версию ПО (модуля ПО, компонента ПО), к которому относится запрос на изменение;
- приоритет выполнения действий с запросом на изменение ПО;
- текущий статус обработки запроса на изменение ПО.

5.6 Разработка, уточнение и анализ архитектуры программного обеспечения

5.6.1 Цели

5.6.1.1 Создание условий для снижения количества возможных недостатков при разработке архитектуры ПО.

5.6.1.2 Уточнение архитектуры ПО в процессе разработки кода.

5.6.2 Требования к реализации

5.6.2.1 Определить требования безопасности к принципам проектирования архитектуры ПО, направленным на снижение количества возможных недостатков в разрабатываемом ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.3.3.2.

5.6.2.2 Выполнить первичное проектирование архитектуры ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.3.3.1, 5.6.3.1.

5.6.2.3 Установить критерии необходимости уточнения архитектуры ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2.

5.6.2.4 Выполнять уточнение архитектуры ПО в процессе разработки кода и его изменений с установленной периодичностью или при наступлении определенных событий.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.6.3.3.

5.6.3 Артефакты реализации требований

5.6.3.1 Требования к принципам проектирования архитектуры ПО должны содержать информацию, позволяющую на начальном этапе проектирования ПО получить представление о принятых подходах и принципах проектирования архитектуры ПО (например, инкапсуляция, уникальность, разделение задач, применение заимствованных компонентов и т. п.), в том числе с точки зрения безопасности («нулевое доверие», «протоколирование событий», «резервное копирование», «формирование перечня недопустимых событий», «приоритетное использование языков с безопасной моделью памяти» и т. п.).

5.6.3.2 Описание архитектуры ПО должно включать, как минимум, следующую информацию:

- назначение ПО и сценарии его использования;
- описание среды функционирования;
- ограничения и указания по применению;
- проект ПО на уровне подсистем (модулей), включающий описание их назначения, структуры, особенностей реализации, применяемых языков программирования, взаимодействия друг с другом и другим ПО с указанием соответствующих интерфейсов, сетевых портов, протоколов.

5.6.3.3 Критерии необходимости уточнения архитектуры ПО должны содержать информацию о периодичности пересмотра (уточнения) архитектуры ПО в процессе разработки ПО или о событиях, при наступлении которых необходимо уточнять архитектуру ПО.

5.6.3.4 Архитектура ПО, уточненная по результатам выполнения требований 5.6.2.4, должна содержать информацию об особенностях реализации ПО в процессе разработки ПО, принятых решениях по корректировкам архитектурных решений в процессе разработки, в том числе связанных с безопасностью, и причинах, их вызвавших.

5.7 Моделирование угроз и разработка описания поверхности атаки

5.7.1 Цели

5.7.1.1 Создание условий для снижения количества недостатков, связанных с особенностями реализации архитектуры ПО и логики его функционирования, выработка мер по нейтрализации угроз безопасности, связанных с особенностями реализации архитектуры ПО.

5.7.1.2 Уточнение модели угроз и описания поверхности атаки по результатам разработки кода и его изменений.

5.7.2 Требования к реализации

5.7.2.1 Выполнить первичное моделирование угроз для ПО (разработать модель угроз ПО); для выявленных угроз безопасности информации составить перечень мер по их нейтрализации (снижению вероятности возникновения).

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2.

5.7.2.2 Выполнить первичное описание поверхности атаки.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2.

5.7.2.3 Сформировать перечень целей (функциональных подсистем, модулей (компонентов) ПО и их интерфейсов) для проведения дальнейших исследований безопасности ПО (например, фаззинг-тестирования) с учетом архитектуры ПО, результатов моделирования угроз и выполнения анализа поверхности атаки.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2.

5.7.2.4 Выполнять уточнение модели угроз ПО для изменяемого в ходе разработки ПО кода с установленной периодичностью или при наступлении определенных событий.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.7.3.1, 5.7.3.3, 5.7.3.4.

5.7.2.5 Выполнять уточнение описания поверхности атаки для изменяемого в ходе разработки ПО кода с установленной периодичностью или при наступлении определенных событий.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.7.3.1, 5.7.3.3, 5.7.3.4.

5.7.2.6 При уточнении описания поверхности атаки выполнять анализ поверхности атаки методом идентификации интерфейсов ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.7.3.1, 5.7.3.3, 5.7.3.4.

Примечание — Используемые методы анализа сетевых интерфейсов способствуют получению информации об узлах сети, именах устройств, IP-адресах, операционных системах, запущенных программах и службах, именах пользователей, группах и открытых портах и могут включать анализ локальных и сетевых интерфейсов взаимодействия пользователя с ПО (модулями ПО, компонентами ПО) и взаимодействий модулей (компонентов) ПО между собой, средой функционирования и внешними объектами при их наличии.

5.7.2.7 Уточнять перечень целей (функциональных подсистем, модулей (компонентов) ПО и их интерфейсов) для проведения дальнейших исследований безопасности ПО (например, фаззинг-тестирования) с учетом уточненной архитектуры ПО, результатов моделирования угроз и выполнения анализа поверхности атаки для разработанного кода ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.7.3.1, 5.7.3.3, 5.7.3.4.

5.7.3 Артефакты реализации требований

5.7.3.1 Модель угроз должна включать совокупность угроз безопасности, актуальных для разрабатываемого ПО.

Каждая угроза безопасности представляется в виде совокупности свойств (характеристик), включающей, как минимум, краткое описание угрозы, предполагаемый объект воздействия и возможные последствия реализации угрозы.

Примечание — При составлении перечня угроз безопасности и их описания рекомендуется учитывать положения ГОСТ Р 58412, а также угрозы безопасности информации Банка данных угроз безопасности информации ФСТЭК России, других источников (например, методологии STRIDE, Open Web Application Security Project (OWASP), DREAD и пр.). В модели угроз рекомендуется указывать использованную при моделировании методологию, в том числе в случае ее собственной разработки.

5.7.3.2 Перечень мер по нейтрализации (снижению вероятности возникновения) угроз безопасности информации содержит перечень необходимых действий (доработок ПО, иных мер). Перечень мер по нейтрализации (снижению вероятности возникновения) угроз безопасности информации должен быть приоритизирован с точки зрения критичности возможного ущерба от реализации угроз безопасности информации.

5.7.3.3 Описание поверхности атаки должно включать совокупность потенциальных областей воздействия на информационную (автоматизированную) систему с использованием разрабатываемого ПО, которые могут быть использованы нарушителем для проведения компьютерной атаки. Описание поверхности атаки может быть частью модели угроз.

5.7.3.4 Перечень целей должен включать список функциональных подсистем, модулей (компонентов) ПО и их интерфейсов, составляющих поверхность атаки, подлежащих дополнительному анализу с точки зрения безопасности.

5.7.3.5 Модель угроз, уточненная по результатам выполнения требований 5.7.2.4, должна дополнительно (в случае применимости) содержать угрозы безопасности ПО, актуальные для выполненных изменений.

5.7.3.6 Описание поверхности атаки, уточненное по результатам выполнения требований 5.7.2.5, должно включать перечень функциональных подсистем, модулей (компонентов) ПО и их интерфейсов, составляющих поверхность атаки, актуальных для разработанного кода ПО.

5.7.3.7 Перечень целей, уточненный по результатам выполнения требований 5.7.2.7, для проведения дальнейших исследований безопасности ПО должен содержать описание функциональных подсистем, модулей (компонентов) ПО, их интерфейсов, для которых предполагаются дальнейшие исследования в части безопасности при реализации других процессов разработки безопасного ПО.

5.8 Формирование и поддержание в актуальном состоянии правил кодирования

5.8.1 Цели

5.8.1.1 Обеспечение эффективной и единообразной организации оформления и использования исходного кода в соответствии с предъявляемыми к ПО требованиями.

5.8.2 Требования к реализации

5.8.2.1 Принять и использовать в процессе разработки кода ПО регламент оформления исходного кода и безопасного кодирования для используемых разработчиком языков программирования.

Примечание — Под безопасным кодированием здесь и далее понимаются практики разработки ПО в соответствии с предъявляемыми в указанном выше регламенте требованиями по безопасности.

5.8.2.2 Учитывать при разработке регламента оформления исходного кода и безопасного кодирования примеры опасных и безопасных конструкций для используемых в ПО языков программирования.

5.8.2.3 Учитывать при разработке регламента оформления исходного кода и безопасного кодирования общепринятые стандарты и рекомендации разработчиков (экспертов, специалистов) для соответствующих языков программирования.

5.8.2.4 При разработке ПО рекомендуется использовать программные средства автоматической проверки правил кодирования.

Примечание — Допускается реализовывать проверку правил кодирования средствами компиляции или статического анализа.

5.8.3 Артефакты реализации требований

5.8.3.1 Регламент оформления исходного кода и безопасного кодирования должен содержать:

- информацию о способах оформления исходного кода (например, способы выбора наименований переменных, функций, классов и т. п.; стиль отступов при оформлении логических блоков; способы ограничения логических блоков; правила использования пробелов при оформлении логических и арифметических выражений; стиль комментариев и правила документирования кода; ограничения [например, размер строк кода по горизонтали, строк в модуле и т. п.)];
- перечень запрещенных способов кодирования, конструкций и т. п. (например, указание паролей в исходном коде ПО в явном виде, использование «магических чисел» и т. п.);
- примеры опасных и безопасных конструкций для используемых языков программирования;
- область применения правил кодирования;
- порядок проверки выполнения правил кодирования для вносимых изменений в исходный код ПО;
- рекомендации разработчиков языков программирования по использованию стандартов кодирования (языков программирования, в том числе собственной разработки), принятые разработчиком ПО.

5.9 Экспертиза исходного кода

5.9.1 Цели

5.9.1.1 Обеспечение соответствия исходного кода ПО предъявляемым к нему требованиям.

5.9.2 Требования к реализации

Примечание — Требования настоящего подраздела затрагивают анализ исходного кода ПО разработчиком, непосредственно не участвовавшем в разработке анализируемого кода ПО, а также анализ кода ПО, для которого отсутствуют инструменты статического анализа или результаты их работы нуждаются в подтверждении.

5.9.2.1 Разработать регламент проведения экспертизы исходного кода ПО.

5.9.2.2 Проводить экспертизу определенных областей кода ПО (в первую очередь для модулей (компонентов) ПО, составляющих поверхность атаки) в соответствии с регламентом проведения экспертизы исходного кода ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.9.3.1.

5.9.2.3 Проводить экспертизу кода рекомендуется с использованием программных средств, автоматизирующих проведение экспертизы и интегрированных с системой контроля версий разрабатываемого ПО.

5.9.3 Артефакты реализации требований

5.9.3.1 Регламент проведения экспертизы исходного кода ПО должен содержать следующие сведения:

- обязанности сотрудников и их роли при проведении экспертизы исходного кода ПО;
- базовые требования к экспертизе (количество участников; области кода, подлежащего экспертизе; используемые инструменты и т. д.);
- описание основных проверок (например, сценариев, шаблонов, чек-листов) проведения экспертизы исходного кода ПО.

5.9.3.2 Результаты экспертизы кода должны содержать следующие сведения:

- информацию о проанализированных модулях (компонентах) ПО;
- перечень необходимых изменений;
- вопросы к частям кода, экспертиза которых затруднена и требует дополнительных разъяснений;
- предложения по улучшению.

5.10 Статический анализ исходного кода

5.10.1 Цели

5.10.1.1 Предотвращение внесения потенциально опасных конструкций и ошибок в ПО, а также использования опасных конструкций и уязвимостей из заимствованного кода.

5.10.2 Требования к реализации

5.10.2.1 Разработать регламент проведения статического анализа исходного кода ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2.

5.10.2.2 Определить инструменты статического анализа для каждого используемого в ПО языка программирования.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.10.3.1.

5.10.2.3 Определить конфигурацию и параметры настройки инструментов статического анализа.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.10.3.1, 5.10.3.2.

5.10.2.4 Проводить статический анализ с использованием инструментов статического анализа с регистрацией всех предупреждений о потенциальных ошибках, полученных по результатам работы инструментов статического анализа.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.10.3.1, 5.10.3.2, 5.10.3.3.

5.10.2.5 Осуществлять пересмотр конфигурации и параметров настройки инструментов статического анализа при выполнении установленных событий (изменениях в правилах сборки, применяемых статических анализаторах, получении информации об уязвимостях и т. п.).

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.10.3.1, 5.10.3.2, 5.10.3.3.

5.10.2.6 Осуществлять повторный статический анализ ПО после устранения ранее выявленных ошибок и уязвимостей; внесения изменений в ходе разработки в исходные тексты ПО; изменения используемых версий компиляторов, сред выполнения (для компилируемого в промежуточное представление или интерпретируемого кода), обновлений используемых инструментов статического анализа.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.10.3.1, 5.10.3.2.

5.10.3 Артефакты реализации требований

5.10.3.1 Регламент проведения статического анализа исходного кода ПО должен содержать следующие сведения:

- обязанности сотрудников и их роли при проведении статического анализа;
- критерии выбора инструментов статического анализа;
- критерии выбора ПО (модулей ПО, компонентов ПО, функциональных подсистем ПО), подлежащих проведению статического анализа;
- правила обработки срабатываний средств статического анализа;
- типы и критичность ошибок (уязвимостей), выявляемых статическим анализатором, подлежащих устранению, и приоритеты устранения ошибок (уязвимостей);
- периодичность проведения статического анализа или события, при наступлении которых необходимо выполнять повторный статический анализ;
- критерии пересмотра конфигурации и параметров настройки инструментов статического анализа.

5.10.3.2 Перечень инструментов статического анализа должен включать наименования инструментов статического анализа, их версии и информацию о соответствии используемым языкам программирования.

5.10.3.3 Конфигурации и параметры настройки инструментов статического анализа должны обеспечивать выполнение требований регламента проведения статического анализа в части выявления типов и критичности ошибок (уязвимостей), периодичности проведения статического анализа или событий, при наступлении которых необходимо выполнять повторный статический анализ.

5.10.3.4 Отчеты по результатам проведения статического анализа должны включать:

- срабатывания инструментов статического анализа;
- результаты анализа (разметки) выявленных ошибок (срабатываний статического анализатора).

5.10.3.5 Конфигурации и параметры настройки инструментов статического анализа, уточненные по результатам выполнения требований 5.10.2.5, должны обеспечивать выполнение требований регламента проведения статического анализа в части выполнения критериев их пересмотра.

5.11 Динамический анализ кода программы

5.11.1 Цели

5.11.1.1 Обнаружение недостатков и уязвимостей в коде ПО в процессе его выполнения.

5.11.2 Требования к реализации

5.11.2.1 Разработать регламент проведения динамического анализа кода ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.7.3.1, 5.7.3.3.

5.11.2.2 Определить инструменты динамического анализа и фаззинг-тестирования, порядок их применения.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.7.3.1, 5.7.3.3, 5.7.3.4, 5.11.3.1.

5.11.2.3 Определить перечень модулей (компонентов) ПО, которые необходимо подвергнуть динамическому анализу, включая фаззинг-тестирование.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.7.3.1, 5.7.3.3, 5.7.3.4, 5.11.3.1, 5.11.3.2.

5.11.2.4 Определить сценарии проведения тестирования для каждого исследуемого модуля (компонента) ПО средствами динамического анализа, включая инструменты проведения фаззинг-тестирования.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.7.3.1, 5.7.3.3, 5.7.3.4, 5.11.3.1, 5.11.3.2, 5.11.3.3.

5.11.2.5 Проводить динамический анализ с использованием инструментов динамического анализа.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.11.3.1, 5.11.3.2, 5.11.3.3, 5.11.3.4.

Примечание — Используемые методы динамического анализа могут позволять осуществлять динамический анализ кода программы путем подачи заведомо некорректных входных данных, динамического профилирования, путем отладки программы, путем поиска защищаемой информации (в оперативной памяти, других местах среды исполнения кода), путем исследования поведения программы с использованием встраиваемых инструментальных датчиков срабатывания ошибок (санитайзеров) или инструментированных с использованием средств динамического двоичного инструментирования, другими применимыми методами, в том числе определенными соответствующими национальными стандартами.

5.11.2.6 Проводить повторный динамический анализ модулей (компонентов) ПО с целью контроля устранения ошибок.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.11.3.1, 5.11.3.2, 5.11.3.3, 5.11.3.4.

5.11.2.7 Проводить фаззинг-тестирование.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.11.3.1, 5.11.3.2, 5.11.3.3, 5.11.3.4.

5.11.2.8 При проведении фаззинг-тестирования использовать тестовые коллекции входных данных, подлежащие дальнейшим мутациям, для каждого из подвергаемых фаззинг-тестированию модуля (компонента) ПО (при использовании инструментов выполнения фаззинг-тестирования, использующих коллекции входных данных), вызывающие использование различных функциональных возможностей тестируемого модуля (компонента) ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.2, 5.7.3.1, 5.7.3.3, 5.7.3.4, 5.11.3.1, 5.11.3.2, 5.11.3.3, 5.11.3.4.

5.11.2.9 Устранять выявленные в процессе динамического анализа, включая фаззинг-тестирование, ошибки в соответствии с принятыми процедурами устранения найденных средствами динамического анализа ошибок.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.11.3.1, 5.11.3.5, 5.11.3.6.

5.11.3 Артефакты реализации требований

5.11.3.1 Регламент проведения динамического анализа кода ПО должен содержать следующие сведения:

- обязанности сотрудников и их роли при проведении динамического анализа и фаззинг-тестирования;
- критерии выбора инструментов динамического анализа, включая инструменты проведения фаззинг-тестирования;
- критерии выбора методов и способов динамического анализа;
- критерии выбора модулей (компонентов) ПО, которые необходимо подвергнуть динамическому анализу, включая фаззинг-тестирование;
- правила обработки срабатываний средств динамического анализа, требующих обработки (аварийная остановка, зависание и т. п.);
- процедуры устранения найденных средствами динамического анализа ошибок;
- периодичность проведения динамического анализа или события, при наступлении которых необходимо выполнять повторный динамический анализ (критерии проведения повторного динамического анализа);
- периодичность проведения фаззинг-тестирования и критерии его завершения.

5.11.3.2 Перечень инструментов динамического анализа, включая инструменты проведения фаззинг-тестирования, должен включать:

- наименования инструментов динамического анализа, их версии и их соответствие исследуемым модулям (компонентам) ПО;
- параметры эксплуатации инструментов динамического анализа (для платформ, языков программирования и т. п.).

5.11.3.3 Перечень модулей (компонентов) ПО, которые необходимо подвергнуть динамическому анализу, включая фаззинг-тестирование, отвечающий требованиям регламента проведения динамического анализа, должен включать:

- наименование модуля (компонента) ПО;
- идентификатор модуля (компонента) ПО.

5.11.3.4 Сценарии проведения тестирования для каждого исследуемого модуля (компонента) ПО средствами динамического анализа, включая инструменты проведения фаззинг-тестирования, обеспечивающие выполнение требований регламента проведения динамического анализа, должны включать:

- идентификатор модуля (компонента) ПО;
- наименование используемого инструмента;
- параметры настройки инструмента;
- критерии запуска и остановки тестирования.

5.11.3.5 Отчеты по результатам проведения динамического анализа должны включать:

- срабатывания инструментов динамического анализа;
- результаты анализа (обработки) выявленных ошибок (срабатываний динамического анализатора) для определенных регламентом типов ошибок, требующих обработки (аварийная остановка, зависание и т. п.).

5.11.3.6 Отчеты по результатам проведения фаззинг-тестирования должны включать:

- сведения о результатах работы инструментов фаззинг-тестирования (длительность проведения фаззинг-тестирования, количество аварийных завершений работы ПО, количество найденных путей выполнения и др.);
- результаты анализа (обработки) аварийных завершений работы ПО, выявленных при проведении фаззинг-тестирования.

5.12 Использование безопасной системы сборки программного обеспечения

5.12.1 Цели

5.12.1.1 Обеспечение безопасности при сборке ПО, недопущение привнесения в код ошибок, обусловленных небезопасными преобразованиями кода.

5.12.2 Требования к реализации

5.12.2.1 Разработать регламент использования системы безопасной сборки ПО.

5.12.2.2 Для разрабатываемого ПО должна быть зафиксирована информация о системе сборки ПО и сборочной среде.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.12.3.1.

Примечание — Составлять и актуализировать перечень программных инструментов системы сборки ПО допустимо как вручную, так и средствами композиционного анализа.

5.12.2.3 Обеспечивать выполнение рекомендаций производителя по безопасному использованию инструмента из состава системы сборки ПО (при их наличии).

5.12.3 Артефакты реализации требований

5.12.3.1 Регламент использования системы безопасной сборки ПО должен содержать, как минимум, следующие сведения:

- обязанности сотрудников и их роли при выполнении сборки ПО;
- критерии выбора инструментов сборки ПО;
- критерии приемки результатов сборки;
- порядок регистрации событий, генерируемых инструментами сборки ПО.

5.12.3.2 Информация о сборочной среде должна содержать:

- описание особенностей функционирования сборочной среды;
- перечень программных инструментов, применяемых в системе сборки ПО, их версий и конфигураций.

5.12.3.3 Артефакты реализации требований, подтверждающие соответствие инструмента из состава системы сборки ПО рекомендациям производителя по безопасному использованию, должны содержать перечень выполненных рекомендаций производителя инструмента сборки ПО с указанием конкретных параметров настроек и конфигураций.

5.13 Обеспечение безопасности сборочной среды программного обеспечения

5.13.1 Цели

5.13.1.1 Обеспечение безопасности при сборке ПО, недопущение привнесения в результаты сборки ПО уязвимостей и ошибок со стороны сборочной среды.

5.13.2 Требования к реализации

5.13.2.1 Разработать регламент обеспечения безопасности сборочной среды.

5.13.2.2 Зафиксировать описание ожидаемых результатов сборки ПО, прав доступа к среде сборки ПО и хранилищу результатов сборки ПО и ролей пользователей, участвующих в процессе сборки ПО.

5.13.2.3 Разработать схему сборочной среды.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.13.3.2.

5.13.2.4 Обеспечивать регистрацию всех выполняемых действий при сборке ПО в журналах аудита; журналы аудита должны храниться способом, обеспечивающим их целостность; сроки хранения журналов аудита должны быть зафиксированы в регламенте обеспечения безопасности сборочной среды.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.13.3.1, 5.13.3.2.

5.13.2.5 Обеспечивать хранение результатов сборки ПО в выделенном хранилище — хранилище результатов сборки ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.13.3.1, 5.13.3.2.

5.13.2.6 Обеспечивать повторяемость сборки ПО (если применимо).

Примечание — Под повторяемостью сборки имеется в виду обеспечение предсказуемости результатов сборок, обеспечение полного бинарного соответствия результатов повторных сборок не требуется.

5.13.2.7 Обеспечивать управление доступом к среде сборки ПО и хранилищу результатов сборки ПО на основе ролей пользователей.

5.13.2.8 Обеспечивать защиту каналов связи с внешними источниками данных для обеспечения конфиденциальности информации, обрабатываемой в сборочной среде.

5.13.3 Артефакты реализации требований

5.13.3.1 Регламент обеспечения безопасности сборочной среды должен содержать, как минимум, следующие сведения:

- обязанности сотрудников и их роли при проведении сборок ПО;

- порядок регистрации событий безопасности при реализации сборок ПО в журналах аудита;
- сроки хранения журналов аудита;
- описание мер безопасности, необходимых для реализации в сборочной среде.

5.13.3.2 Информация о безопасности сборочной среды должна содержать:

- описание ожидаемых результатов сборки ПО;
- описание прав доступа к сборочной среде и хранилищу результатов сборки ПО, а также ролей пользователей, участвующих в процессе сборки ПО.

5.13.3.3 Схематическое изображение сборочной среды должно содержать:

- элементы сборочной среды (серверы, узлы, виртуальные узлы, элементы среды контейнеризации и т. п.);
- связи между элементами сборочной среды, позволяющие отследить порядок (очередность) выполнения сборочных действий;
- компоненты сборочной среды, реализующие отдельные функции, в том числе меры безопасности (средства защиты информации, инструменты статического анализа и др.).

Примечание — При изображении сборочной среды в графическом виде рекомендуется использовать стандартизированные графические нотации (UML, IDEF, C4 и т. п.).

5.13.3.4 Журналы аудита процессов сборки ПО должны содержать следующую информацию:

- дату и время начала и завершения сборки ПО;
- информацию о версии собираемого ПО (модуля ПО, компонента ПО);
- информацию об используемой конфигурации сборки ПО;
- информацию о шагах сборки ПО;
- информацию о событиях безопасности в соответствии с регламентом обеспечения безопасности сборочной среды.

5.13.3.5 В качестве артефакта реализации требований, подтверждающих хранение результатов сборки ПО в выделенном хранилище, может использоваться журнал аудита сборки ПО, в котором указано место сохранения собранного модуля (компонента) ПО, результаты контрольного суммирования файлов, скачанных из хранилища результатов сборки ПО, и последующего сравнения их с контрольными суммами, указанными в журнале аудита сборки ПО или в графическом интерфейсе системы хранения результатов сборки ПО.

Примечание — При разработке и реализации регламента доступа к исходному коду ПО рекомендуется руководствоваться принципами минимизации привилегий и разделения полномочий.

5.13.3.6 В качестве артефактов реализации требований, подтверждающих повторяемость сборки ПО, могут использоваться журналы аудита выполненных сборок, сравненные друг с другом; результаты контрольного суммирования файлов, полученных при разных запусках сборок, и последующего их сравнения (по контрольным суммам, по бинарному представлению, по наименованию и размеру и др.).

5.14 Управление доступом и контроль целостности кода при разработке программного обеспечения

5.14.1 Цели

5.14.1.1 Обеспечение управления доступом к исходному коду и его целостности.

5.14.2 Требования к реализации

5.14.2.1 Разработать регламент доступа к исходному коду ПО и обеспечения его целостности.

Примечание — При разработке и реализации регламента доступа к исходному коду ПО рекомендуется руководствоваться принципами минимизации привилегий и разделения полномочий.

5.14.2.2 Осуществлять управление доступом к исходному коду ПО на основе выбранной модели управления доступом.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.14.3.1.

5.14.2.3 Осуществлять контроль целостности собственного исходного кода.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.14.3.1, 5.14.3.2.

5.14.3 Артефакты реализации требований

5.14.3.1 Регламент доступа к исходному коду ПО и обеспечения его целостности должен содержать следующие сведения:

- обязанности сотрудников, их права и роли при разработке ПО;
- правила хранения исходного кода ПО, включая правила резервного копирования исходного кода ПО;
- правила внесения изменений (модификации, добавления, удаления) в исходный код ПО;
- критерии выбора способов и инструментов контроля целостности ПО;
- критерии выбора модулей (компонентов) ПО, подлежащих контролю целостности;
- описание процедуры контроля целостности исходного кода ПО.

5.14.3.2 Описание модели управления доступом к исходному коду ПО должно включать:

- перечень сотрудников, их права и обязанности при разработке ПО;
- описание выбранной модели управления доступом и используемых инструментов управления доступом.

5.14.3.3 Результаты выполнения контроля целостности собственного исходного кода должны обеспечивать соответствие требованиям регламента доступа к исходному коду ПО и обеспечения его целостности и позволять сделать однозначный вывод о целостности собственного исходного кода.

5.15 Обеспечение безопасности используемых секретов

5.15.1 Цели

5.15.1.1 Обеспечение безопасного использования секретов.

Примечание — В данном подразделе под секретами понимаются данные в любом виде, которые могут использоваться для обеспечения аутентификации и/или целостности и/или конфиденциальности информации (пароли, цифровые сертификаты и т. п.), в том числе путем применения в соответствии с законодательством Российской Федерации средств криптографической защиты информации или иными методами.

5.15.2 Требования к реализации

Примечание — Требования к реализации, изложенные в данном подразделе, применяются пользователями стандарта по их усмотрению и в необходимых им объемах.

5.15.2.1 Разрабатывать регламент использования секретов.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3.

5.15.2.2 Использовать секреты.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.15.3.1.

5.15.2.3 Проверять код и конфигурационные файлы ПО на предмет включения секретов для вносимых изменений в ПО.

Примечание — Допускается реализовывать проверку безопасности используемых секретов средствами статического или композиционного анализа.

5.15.2.4 Для хранения, управления и предоставления секретов использовать систему управления секретами.

5.15.3 Артефакты реализации требований

5.15.3.1 Регламент использования секретов может содержать следующие сведения:

- обязанности сотрудников и их роли при использовании секретов;
- основные принципы использования секретов;
- зоны ответственности подразделений и сотрудников в части использования секретов;
- порядок предоставления доступа к секретам;
- типы секретов, сроки их эксплуатации, действия при компрометации;
- порядок формирования и хранения секретов;
- порядок ротации секретов;
- требования к системам хранения секретов.

5.15.3.2 Описание реализации процедуры использования секретов может включать следующие сведения:

- порядок подписи исполняемого кода ПО (например, с использованием цифровых сертификатов);
- порядок подписи исходного кода (например, с использованием цифровых сертификатов).

5.16 Использование инструментов композиционного анализа

5.16.1 Цели

5.16.1.1 Создание условий для снижения рисков наследования уязвимостей и недекларированных возможностей при использовании заимствованного кода в коде ПО разработчика.

5.16.2 Требования к реализации

5.16.2.1 Разработать регламент композиционного анализа.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3.

Примечание — В данном подразделе под композиционным анализом понимается вид работ, основанный на формировании перечня зависимостей ПО, определении особенностей их использования, выявлении наличия уязвимостей и/или иных недостатков в зависимостях ПО.

5.16.2.2 Формировать перечень зависимостей ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3, 5.16.3.1.

5.16.2.3 Контролировать и актуализировать перечень зависимостей ПО в соответствии с регламентом композиционного анализа на предмет наличия известных уязвимостей.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3, 5.16.3.1, 5.16.3.2.

5.16.2.4 Осуществлять анализ заимствованных компонентов, составляющих поверхность атаки, на предмет наличия известных уязвимостей при сборке (непосредственно перед сборкой) ПО (модулей ПО, компонентов ПО).

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3, 5.16.3.1, 5.16.3.2.

5.16.2.5 Применять корректирующие воздействия по результатам анализа заимствованных компонентов на предмет наличия известных уязвимостей.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.16.3.1, 5.16.3.2, 5.16.3.3, 5.16.3.4, 5.16.3.5.

5.16.3 Артефакты реализации требований

5.16.3.1 Регламент композиционного анализа должен содержать следующие сведения:

- обязанности сотрудников и их роли при проведении композиционного анализа;
- правила отслеживания уязвимостей для заимствованных компонентов, участвующих в сборке ПО;
- правила проведения анализа заимствованных компонентов на предмет наличия известных уязвимостей;
- правила принятия компенсирующих и защитных мер по противодействию выявленным угрозам безопасности в цепочке поставки сторонних компонентов;
- периодичность проведения композиционного анализа в соответствии с установленными практиками сборки ПО.

5.16.3.2 Перечень зависимостей ПО должен включать следующие сведения:

- перечень модулей (компонентов) заимствованного ПО с указанием их версий;
- источник (поставщик) модулей (компонентов) заимствованного ПО.

5.16.3.3 Результаты контроля актуальности перечня зависимостей ПО должны включать следующие сведения:

- описание процедуры контроля перечня зависимостей ПО и его актуализации;
- описание инструментов контроля актуальности перечня зависимостей ПО;
- журналы регистрации событий, связанных с контролем актуальности перечня зависимостей ПО, а также связанных с обновлениями модулей (компонентов) заимствованного ПО, участвующих в сборке ПО.

5.16.3.4 Результаты анализа заимствованных компонентов должны содержать следующие сведения:

- сведения о наличии/отсутствии известных уязвимостей в заимствованных компонентах;
- сведения о критичности выявленных уязвимостей в заимствованных компонентах.

5.16.3.5 Результаты применения корректирующих воздействий по устранению уязвимостей в зависимостях ПО могут содержать:

а) для ПО с закрытыми исходными текстами (проприетарного ПО):

1) результаты анализа применимости и реализуемости уязвимости,

2) результаты обращения к поставщику (разработчику) уязвимых модулей (компонентов) ПО по поводу их обновления,

3) результаты обновления зависимых компонентов ПО по мере поступления обновлений от поставщика (разработчика);

б) для ПО с открытыми исходными текстами:

1) результаты анализа применимости и реализуемости уязвимости,

2) результаты попыток обновления зависимых компонентов, в случае невозможности обновления путем обновления версии — применения собственного механизма исправления.

5.17 Проверка кода на предмет внедрения вредоносного программного обеспечения через цепочки поставок

5.17.1 Цели

5.17.1.1 Создание условий для снижения рисков внедрения вредоносного ПО посредством воздействий на ПО или механизмы его доставки до получения ПО конечными пользователями и недопущение компрометации данных (информации) или информационной системы, использующей такое ПО.

5.17.2 Требования к реализации

5.17.2.1 Осуществлять контроль зависящих от сторонних поставщиков элементов разработки (процессов; компонентов инфраструктуры разработки ПО, зависящих от сторонних поставщиков; компонентов, являющихся частью разрабатываемого ПО, которые поставляются или заимствуются от сторонних поставщиков).

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.12.3.2, 5.16.3.2.

5.17.2.2 Осуществлять контроль договорных обязательств со сторонними поставщиками.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.17.3.1.

Примечание — Показателями контроля договорных обязательств со сторонними поставщиками могут являться перечни сторонних поставщиков, факты заключения договоров о поставках продуктов (услуг), перечень обязательств сторонних поставщиков.

5.17.2.3 Осуществлять выявление элементов инфраструктуры разработчика, воздействие на которые может повлиять на возникновение недекларированных возможностей в ПО.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3, 5.17.3.1.

Примечание — Результаты выявления указанных элементов инфраструктуры разработчика рекомендуется использовать для дальнейшего принятия мер для нейтрализации потенциальных угроз безопасности.

5.17.2.4 Осуществлять контроль использования предсобранного поставщиком ПО (кода, для которого отсутствуют исходные тексты).

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1.

5.17.2.5 Осуществлять анализ кода ПО, полученного через цепочки поставок, на предмет внедрения вредоносного программного обеспечения.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.17.3.1, 5.17.3.3.

5.17.3 Артефакты реализации требований

5.17.3.1 Перечень процессов, компонентов инфраструктуры, частей разрабатываемого ПО, зависящих от сторонних поставщиков, должен содержать следующие сведения:

- описание внутренних процессов, зависящих от сторонних поставщиков;
- описание компонентов инфраструктуры разработки ПО, зависящих от сторонних поставщиков;
- описание компонентов, являющихся частью разрабатываемого ПО, которые поставляются или заимствуются от сторонних поставщиков.

5.17.3.2 Сведения о договорных обязательствах со сторонними поставщиками могут включать следующую информацию:

- перечень поставщиков с указанием поставляемых продуктов (услуг);

- сведения о заключенных договорах со сторонними поставщиками, включающие информацию о поставляемых продуктах (услугах), сроках начала и окончания договоров, иную информацию.

5.17.3.3 Сведения о критичных и вероятных с точки зрения внедрения недекларированных возможностей элементах инфраструктуры (компонентах инфраструктуры разработки ПО, зависящих от сторонних поставщиков) должны содержать следующую информацию:

- перечень элементов инфраструктуры разработчика, воздействие на которые может повлиять на возникновение недекларированных возможностей в ПО;
- информацию о поставщиках продуктов (услуг) для указанных в перечне элементов инфраструктуры разработчика.

5.17.3.4 Результаты контроля использования предсобранного поставщиком ПО должны содержать информацию, позволяющую определить наличие предсобранных поставщиком ПО компонентов и осуществить их идентификацию (по свойствам файлов, контрольным суммам файлов и т. п.).

5.17.3.5 Результаты анализа кода ПО, полученного через цепочки поставок, на предмет внедрения вредоносного программного обеспечения должны содержать, как минимум, отчеты сканирования средств антивирусной защиты.

5.18 Функциональное тестирование

5.18.1 Цели

5.18.1.1 Контроль полноты реализованных функциональных возможностей, обнаружение и исправление ошибок с использованием технологий функционального тестирования.

5.18.2 Требования к реализации

5.18.2.1 Разработать план функционального тестирования.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.3.3.2, 5.6.3.1, 5.7.3.1, 5.7.3.3.

5.18.2.2 Проводить функциональное тестирование, по результатам тестирования разрабатывать отчеты о выполненном функциональном тестировании.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.18.3.1.

5.18.2.3 При проведении функционального тестирования выполнять тестирование на уровне модулей (компонентов), подсистем, всего ПО в целом.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.18.3.1.

5.18.2.4 Регистрировать ход проведения тестирования.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.18.3.1.

5.18.2.5 Организовывать процесс исправления выявленных в ходе тестирования ошибок с использованием системы управления ошибками.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.18.3.2, 5.18.3.3.

5.18.3 Артефакты реализации требований

5.18.3.1 План функционального тестирования должен содержать следующие сведения:

- обязанности сотрудников и их роли при проведении функционального тестирования;
- описание тестового стенда (тестовой среды);
- описание сценариев тестирования для каждой функциональной возможности ПО, включающее формулировку функциональных требований, выполняемые действия по оценке, ожидаемые результаты тестирования и критерии успешного прохождения проверок;
- критерии выполнения повторного тестирования;
- критерии завершения и остановки тестирования.

5.18.3.2 Отчет по результатам функционального тестирования должен содержать, как минимум, следующую информацию:

- описание тестируемого ПО (версии ПО/модулей (компонентов) ПО, номера (идентификаторы) сборок ПО/модулей (компонентов) ПО, системные требования к тестируемому ПО) и его среды функционирования для каждого выполненного сценария тестирования;
- перечень выполненных сценариев тестирования и последовательность их выполнения;
- перечень выполненных действий и ограничений (описание отдельных аспектов, которые не проверялись);

- описание полученных результатов, перечень обнаруженных и исправленных ошибок;
- выводы по результатам тестирования.

В отчетах по результатам функционального тестирования должна содержаться информация, позволяющая идентифицировать выполненные функциональные тесты ПО на уровне модулей (компонентов), подсистем, всего ПО в целом.

5.18.3.3 Журналы функционального тестирования должны содержать, как минимум, следующую регистрационную информацию:

- дату и время выполнения тестовых операций (запуск и завершение сценария тестирования);
- результат выполнения сценария тестирования;
- изменения тестируемого ПО (состава и содержания модулей ПО, компонентов ПО, функциональных подсистем ПО и т. п.);
- возникновение любых сбоев и ошибок.

5.18.3.4 Сведения системы управления ошибками должны содержать, как минимум, регистрационную информацию о выявленных ошибках:

- дату и время тестирования, при котором была выявлена ошибка;
- тестовый сценарий;
- идентификационную информацию о модуле (компоненте) ПО, в котором выявлена ошибка;
- категорию ошибки;
- принятое решение;
- информацию о подтверждении факта наличия ошибки от разработчика;
- информацию об исправлении ошибки.

5.19 Нефункциональное тестирование

5.19.1 Цели

5.19.1.1 Подтверждение того, что поверхность атаки, модель угроз и архитектура ПО содержат необходимую информацию.

5.19.1.2 Обнаружение недостатков программы путем выполнения нефункциональных тестов, в том числе имитирующих действия потенциального нарушителя.

5.19.2 Требования к реализации

5.19.2.1 Проводить нефункциональное тестирование в отношении ПО (модулей ПО, компонентов ПО) в объемах, определяемых разработчиком.

Примечание — В настоящем разделе под нефункциональным тестированием понимаются проверки, не относящиеся к тестированию функциональных возможностей ПО. В рамках нефункционального тестирования могут выполняться следующие проверки:

- сетевых взаимодействий ПО;
- локальных интерфейсов взаимодействия ПО;
- производительности функционирования ПО;
- операций, выполняемых с высокими привилегиями;
- работы с конфиденциальными данными;
- корректности выполнения файловых операций;
- реализации защищенности бинарных файлов;
- реализации системы управления секретами;
- реализации безопасности сетевых протоколов;
- работы системы развертывания продукта;
- реализации мер по устранению или снижению критичности угроз, выявленных при моделировании угроз;
- возможности нарушения логики работы программы;
- безопасности реализации механизмов аутентификации и авторизации;
- безопасности обработки данных, полученных от потенциального нарушителя;
- безопасности реализации клиентской и серверной частей ПО.

5.19.2.2 Разработать регламент нефункционального тестирования.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3.

5.19.2.3 Проводить нефункциональное тестирование с целью выявления локальных и сетевых интерфейсов взаимодействия с ПО (модулями ПО, компонентами ПО) пользователя и взаимодействий модулей (компонентов) ПО между собой, средой функционирования и внешними объектами.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3, 5.19.3.1.

5.19.2.4 Осуществлять выполнение нефункциональных тестов, в том числе имитирующих действия потенциального нарушителя.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3, 5.19.3.1.

5.19.2.5 Осуществлять корректировку описания поверхности атаки, модели угроз и архитектуры ПО по результатам выполнения нефункционального тестирования.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3, 5.19.3.2, 5.19.3.3.

5.19.3 Артефакты реализации требований

5.19.3.1 Регламент нефункционального тестирования должен содержать следующие сведения:

- критерии выбора версий ПО (модулей ПО, компонентов ПО), подлежащих нефункциональному тестированию, и определения периодичности тестирования;
- перечень используемых для нефункционального тестирования методов и средств;
- обязанности сотрудников и их роли при проведении нефункционального тестирования;
- описание типовых сценариев тестирования;
- описание возможностей и мотивации потенциального нарушителя в соответствии с результатами моделирования угроз разрабатываемого ПО;
- описание типовых сценариев проведения компьютерных атак для основных сценариев работы ПО (модулей ПО, компонентов ПО).

5.19.3.2 Отчет по результатам нефункционального тестирования должен содержать следующую информацию:

- краткое описание тестируемого ПО и его инфраструктуры развертывания;
- описание выполненных сценариев тестирования и последовательности их выполнения;
- набор целей (модулей ПО, компонентов ПО) тестирования;
- перечень выполненных действий и ограничений (описание отдельных аспектов, которые не проверялись);
- результаты нефункционального тестирования (снимки экрана (скриншоты), рабочие файлы инструментов нефункционального тестирования и т. п.);
- выводы, включающие следующую информацию: найденные недостатки (уязвимости) программ, средства и методы их выявления, результаты оценки опасности уязвимостей, описание возможных последствий эксплуатации уязвимостей, рекомендации по устранению найденных уязвимостей.

5.19.3.3 Результаты сравнения архитектуры ПО, модели угроз и описания поверхности атаки с полученными фактическими результатами, перечень необходимых изменений в указанных артефактах реализации требований (при необходимости).

5.20 Обеспечение безопасности при выпуске готовой к эксплуатации версии программного обеспечения

5.20.1 Цели

5.20.1.1 Организация приемки ПО с целью недопущения недостатков кода ПО перед его предоставлением пользователям.

5.20.2 Требования к реализации

5.20.2.1 Разработать регламент приемки ПО.

5.20.2.2 Осуществлять анализ степени влияния на безопасность ПО неустраненных ошибок. Информация о неустраненных ошибках выпускаемого ПО должна быть зафиксирована (например, в системе управления изменениями, системе отслеживания ошибок и т. п.).

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.9.3.2, 5.20.3.1, 5.10.3.4, 5.11.3.5, 5.16.3.5, 5.18.3.2.

5.20.2.3 Разработать регламент обеспечения целостности ПО, передаваемого пользователям.

5.20.2.4 Обеспечивать возможность проверки пользователями целостности ПО.

5.20.3 Артефакты реализации требований

5.20.3.1 Регламент приемки ПО должен содержать следующие сведения:

- обязанности сотрудников и их роли при проведении приемки ПО;
- описание типовых сценариев приемки ПО перед предоставлением его пользователям.

5.20.3.2 Результаты анализа влияния на безопасность ПО неустраненных ошибок должны включать следующие сведения:

- перечень выявленных несоответствий;
- принятые решения по устранению несоответствий;
- принятые решения по результатам анализа о влиянии на безопасность ПО неустраненных ошибок.

5.20.3.3 Регламент обеспечения целостности ПО, передаваемого пользователям, должен содержать:

- перечень мер, реализуемых разработчиком ПО с целью обеспечения возможности проверки целостности ПО пользователями;
- порядок применения мер по обеспечению возможности проверки целостности ПО пользователями;
- порядок информирования пользователей ПО о механизмах проверки целостности ПО.

5.20.3.4 Результаты проверки выполнения мер по обеспечению целостности ПО.

5.21 Безопасная поставка программного обеспечения пользователям

5.21.1 Цели

5.21.1.1 Обеспечение защиты ПО, в том числе документации ПО, от угроз, возникающих в процессе передачи ПО пользователю.

5.21.2 Требования к реализации

5.21.2.1 Разработать регламент безопасной поставки ПО пользователям.

5.21.2.2 Фиксировать версии поставляемого пользователям ПО и соответствующей поставляемой документации.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.21.3.1.

5.21.2.3 Организовать хранение копий версий поставляемого пользователям ПО и соответствующей поставляемой документации.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.21.3.1.

5.21.2.4 Поставлять ПО вместе с эксплуатационной документацией, содержащей, как минимум, описание штатного функционирования ПО, параметров настроек (конфигураций) ПО и среды функционирования, действий по установке и настройке средства, как с точки зрения штатного функционирования, так и с точки зрения обеспечения безопасности.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.21.3.1.

5.21.3 Артефакты реализации требований

5.21.3.1 Регламент безопасной поставки ПО пользователям должен содержать следующие сведения:

- обязанности сотрудников и их роли при осуществлении безопасной доставки ПО;
- процедуры хранения копий версий поставляемого пользователям ПО;
- процедуры снятия копий (тиражирования) поставляемого пользователям ПО;
- процедуры поставки ПО (обновлений ПО, включая обновления безопасности, предназначенных для устранения недостатков, в том числе уязвимостей);
- процедуры проверки подлинности ПО (обновлений ПО) пользователем.

5.21.3.2 Сведения о версии поставляемого пользователям ПО должны быть зафиксированы в поставляемой документации (в электронном виде или на физическом носителе).

5.21.3.3 Сведения о месте хранения копий (подлинников, дубликатов) версий поставляемого пользователям ПО (инсталляционных пакетов, дистрибутивных носителей) должны быть зафиксированы в документации (в электронном виде или на физическом носителе), например в регламенте безопасной поставки ПО.

5.21.3.4 Сведения о поставляемой эксплуатационной документации на ПО должны быть зафиксированы (в электронном виде или на физическом носителе), например в регламенте безопасной поставки ПО, в паспорте (формуляре) ПО.

5.22 Обеспечение поддержки программного обеспечения при эксплуатации пользователями

5.22.1 Цели

5.22.1.1 Обеспечение технической поддержки ПО при его эксплуатации с целью устранения выявляемых в ходе использования и обновления ПО недостатков.

5.22.2 Требования к реализации

5.22.2.1 Разработать регламент технической поддержки.

5.22.2.2 Организовать работу службы технической поддержки.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.22.3.1.

5.22.2.3 Разработать процедуру оповещения пользователей о выпуске обновлений (включая обновления безопасности) и необходимости их установки.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.22.3.1.

5.22.2.4 Организовывать обучение специалистов службы технической поддержки работе с поставляемым ПО, особенностям его установки и функционирования, ограничениям и указаниям по эксплуатации.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.22.3.1.

5.22.2.5 Разработать процедуру информирования пользователей ПО о выявленных уязвимостях и способах реализации мер по их нейтрализации до разработки обновлений безопасности, устраняющих уязвимость, по установленным каналам взаимодействия.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.22.3.1.

5.22.3 Артефакты реализации требований

5.22.3.1 Регламент технической поддержки должен содержать следующие сведения:

- обязанности сотрудников и их роли при оказании технической поддержки;
- описание организации службы технической поддержки: режим работы, сроки оказания услуг по технической поддержке пользователей, иная информация об организации службы технической поддержки;
- используемые инструменты;
- описание процедуры взаимодействия службы технической поддержки с пользователями (способы получения обращений пользователей, процесс обработки поступающих сообщений и др.);
- описание процедур оповещения пользователей о выпуске обновлений (включая обновления безопасности) и необходимости их установки;
- описание процедур информирования пользователей ПО о выявленных уязвимостях и способах реализации мер по их нейтрализации до разработки обновлений безопасности, устраняющих уязвимость, по установленным каналам взаимодействия;
- информацию об обучении сотрудников службы технической поддержки.

5.22.3.2 Артефактами реализации требований, подтверждающими наличие технической поддержки, могут являться записи о наличии должностей сотрудников технической поддержки в штатном расписании организации, документированные факты оказания конкретных услуг по технической поддержке пользователей, иные факты и документированные материалы.

5.22.3.3 Артефакты реализации требований, подтверждающие наличие внедренных процессов оповещения пользователей, должны содержать информацию об используемых каналах взаимодействия с пользователями при выпуске обновлений (включая обновления безопасности) и необходимости их установки.

5.22.3.4 Артефакты реализации требований, подтверждающие наличие процессов обучения специалистов службы технической поддержки, могут содержать информацию о пройденных семинарах, вебинарах, курсах или иную информацию, подтверждающую умения и знания специалистов службы технической поддержки, необходимые для работы с поставляемым ПО, его особенностям установки и функционирования, ограничениями и указаниями по эксплуатации.

5.22.3.5 Артефакты реализации требований, подтверждающие наличие процессов информирования пользователей ПО о выявленных уязвимостях, должны содержать информацию об используемых каналах взаимодействия с пользователями и способах реализации мер по их нейтрализации до разработки обновлений безопасности, устраняющих уязвимость.

5.23 Реагирование на информацию об уязвимостях

5.23.1 Цели

5.23.1.1 Обеспечение выявления и устранения уязвимостей при эксплуатации ПО.

5.23.2 Требования к реализации

5.23.2.1 Разработать регламент реагирования на информацию об уязвимостях.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3.

5.23.2.2 Осуществлять обработку поступающих запросов от пользователей (через службу технической поддержки, по иным каналам взаимодействия) с последующим анализом ошибок функционирования на предмет наличия уязвимостей (в случае получения таких запросов).

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.23.3.1.

5.23.2.3 При обработке поступающих запросов и при последующем анализе использовать средства автоматизации (например, систему управления изменениями, систему отслеживания ошибок, систему управления задачами и т. п.).

5.23.2.4 Осуществлять анализ информации о найденных уязвимостях в ПО на предмет подтверждения наличия/отсутствия уязвимостей и принимать решение о необходимости их устранения по результатам оценки.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.23.3.2.

5.23.2.5 Осуществлять оценку актуальности и критичности уязвимости с точки зрения безопасности ПО (в случае получения информации об уязвимости ПО из внешнего источника) и принимать решение о необходимости ее устранения по результатам оценки.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.23.3.3.

5.23.3 Артефакты реализации требований

5.23.3.1 Регламент реагирования на информацию об уязвимостях должен содержать:

- обязанности сотрудников и их роли при реагировании на информацию об уязвимостях ПО;
- правила реагирования на информацию об уязвимостях;
- правила оценки актуальности и критичности уязвимости с точки зрения безопасности ПО;
- периодичность проведения поиска известных (подтвержденных) уязвимостей в общедоступных источниках информации об уязвимостях ПО.

5.23.3.2 Артефакты реализации требований, подтверждающие получение и обработку запросов от пользователей, должны содержать следующие сведения:

- информацию о запросах пользователей об ошибках (уязвимостях) ПО (дата, время запроса, идентификатор пользователя, статус запроса);
- результат анализа ошибок функционирования на предмет наличия уязвимостей.

5.23.3.3 Артефакты реализации требований, подтверждающие выполнение анализа информации о найденных уязвимостях в ПО, должны содержать следующие сведения:

- информацию о результатах тестирования ПО на предмет применимости информации об уязвимости ПО;
- проект (шаблон) ответа пользователям на запросы пользователей об ошибках (уязвимостях) ПО (о применимости информации о найденных уязвимостях);
- решение по результатам анализа информации о найденных уязвимостях в ПО.

5.23.3.4 Артефакты реализации требований, подтверждающие выполнение оценки актуальности и критичности уязвимости с точки зрения безопасности, должны содержать следующие сведения:

- информацию об оценке актуальности уязвимости;
- информацию об оценке уровня критичности уязвимости ПО;
- решение по результатам анализа актуальности и критичности уязвимости.

5.24 Поиск уязвимостей в программном обеспечении при эксплуатации

5.24.1 Цели

5.24.1.1 Организация систематического и углубленного поиска ошибок и уязвимостей в ПО при его эксплуатации в целях упреждающего реагирования: обработки ошибок кода ПО и его конфигураций (настроек) до того, как они будут выявлены сторонними лицами и повлекут инциденты информационной безопасности.

5.24.2 Требования к реализации

Примечание — Требования 5.24.2.1—5.24.2.2 являются обязательными. Требования к реализации, изложенные в 5.24.2.3—5.24.2.4, применяются пользователями стандарта по их усмотрению и в необходимых им объемах.

5.24.2.1 Разработать регламент поиска ошибок и уязвимостей в ПО при его эксплуатации.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.6.3.1, 5.7.3.1, 5.7.3.3.

5.24.2.2 Актуализировать информацию об уязвимостях ПО из открытых источников на регулярной основе на всем протяжении срока действия его технической поддержки: выполнять поиск в открытых источниках информации об уязвимостях самого ПО и его сторонних компонентов.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.16.3.2, 5.24.3.1.

Примечание — В случае наличия угроз безопасности, связанных с недостатками среды функционирования ПО, может осуществляться поиск информации об уязвимостях сред функционирования.

5.24.2.3 Проводить проверки кода ПО и настроек конфигураций ПО на регулярной основе на всем протяжении срока действия его технической поддержки с целью поиска ошибок и уязвимостей.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в 5.24.3.1.

Примечания

1 Проверки реализуются другими инструментами анализа или теми же инструментами, но с другими настройками конфигурации, с целью обеспечения анализа с меньшей долей пропусков ошибок за счет применения специализированных алгоритмов, привлечения больших вычислительных и временных ресурсов.

2 Проверки кода ПО и настроек конфигураций ПО при его эксплуатации могут выполняться как собственными силами разработчика, так и с привлечением сторонних организаций и исследователей, в том числе в рамках публичных программ поиска уязвимостей за вознаграждение (программ багбаунти).

5.24.2.4 Оценивать выявленные ошибки на предмет наличия уязвимостей.

5.24.3 Артефакты реализации требований

5.24.3.1 Регламент поиска ошибок и уязвимостей в эксплуатирующемся ПО должен содержать:

- обязанности сотрудников и их роли при поиске ошибок и уязвимостей в эксплуатирующемся ПО;
- правила поиска известных (подтвержденных) уязвимостей в общедоступных источниках информации об уязвимостях ПО, его программных компонентов и сред его функционирования;
- состав проводимых проверок и периодичность их проведения на протяжении всего срока действия технической поддержки ПО для каждой версии ПО.

5.24.3.2 Регулярные отчеты по результатам проводимых проверок, в которые включается информация об исправлении найденных ошибок, выпуска обновлений ПО и доставки обновлений ПО пользователям.

5.25 Обеспечение безопасности при выводе программного обеспечения из эксплуатации

5.25.1 Цели

5.25.1.1 Недопущение реализации угроз безопасности, связанных с эксплуатацией неподдерживаемой версии ПО.

5.25.2 Требования к реализации

5.25.2.1 Разработать регламент вывода ПО из эксплуатации.

5.25.2.2 Информировать пользователя о планах прекращения технической поддержки ПО (версии ПО) и своевременно уведомлять об этом.

5.25.3 Артефакты реализации требований

5.25.3.1 Регламент вывода ПО из эксплуатации должен содержать описание условий, при которых ПО (версию ПО) необходимо выводить из эксплуатации, обязанности сотрудников и их роли при осуществлении вывода ПО из эксплуатации ПО и порядок оповещения пользователей о планах прекращения технической поддержки ПО (версии ПО).

Приложение А
(справочное)

Инициализация процессов разработки безопасного программного обеспечения

А.1 Инициализация процессов разработки безопасного программного обеспечения

Примечание — Инициализация процессов разработки безопасного ПО предполагает первоначальную реализацию — при инициализации соответствующих процессов и обосновании необходимости их внедрения. Оценка указанных процессов не является обязательной при внешнем контроле реализации (аудите), а их описание приведено в настоящем стандарте в качестве справочной информации.

А.1.1 Цели

А.1.1.1 Оценка готовности разработчика к внедрению процессов разработки безопасного ПО, текущего статуса внедрения.

А.1.1.2 Подготовка к внедрению процессов разработки безопасного ПО.

А.1.2 Требования к реализации

А.1.2.1 Выполнить анализ текущего статуса реализации процессов, которые реализованы разработчиком в области разработки безопасного ПО (допускается проведение анализа как силами сотрудников разработчика, так и привлекаемых сторонних организаций).

А.1.2.2 Выполнить анализ потребностей в ресурсах, необходимых для реализации процессов разработки безопасного ПО.

А.1.2.3 Определить пути изменения и совершенствования реализованных ранее процессов с учетом полученного ранее опыта и развитием технологий и формализовать их в виде плана.

Артефакты реализации требований, учитываемые при выполнении требования, представлены в А.1.3.1, А.1.3.2.

А.1.3 Артефакты реализации требований

А.1.3.1 Результаты анализа текущего статуса реализации процессов, которые реализованы разработчиком в области разработки безопасного ПО, должны содержать следующие сведения:

- перечень процессов разработки безопасного ПО, реализованных и нереализованных разработчиком (в соответствии с настоящим стандартом);
- результаты определения достаточности и соответствия процессов разработки безопасного ПО, реализованных разработчиком, положениям настоящего стандарта и иным стандартам, содержащим требования к разработке безопасного ПО, используемым инструментам и технологиям.

А.1.3.2 Результаты анализа потребностей в ресурсах, необходимых для реализации процессов разработки безопасного ПО, могут содержать оценочные показатели в материальных и людских ресурсах для каждого реализуемого или планируемого к реализации процесса разработки безопасного ПО.

А.1.3.3 План развития процессов разработки безопасного ПО может содержать порядок (очередность) внедрения процессов разработки безопасного ПО с учетом приоритетов разработчика и имеющихся ресурсов, планируемые изменения в организационно-штатной структуре разработчика, планируемые закупки необходимых инструментов, затраты на обучение и др.

УДК 004.05:006.85:006.354 ОКС 35.020

Ключевые слова: безопасное программное обеспечение, уязвимость программы, статический анализ исходного кода, динамический анализ кода программы, композиционный анализ, функциональное тестирование

Редактор *З.А. Лиманская*
Технический редактор *И.Е. Черепкова*
Корректор *С.И. Фирсова*
Компьютерная верстка *Е.А. Кондрашовой*

Сдано в набор 28.10.2024. Подписано в печать 11.11.2024. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 4,18. Уч.-изд. л. 3,55.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
gost.e-ecolog.ru gost.e-ecolog.ru