ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56939–202X

(проект)

Защита информации

РАЗРАБОТКА БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Общие требования

Настоящий проект стандарта не подлежит применению до его утверждения

Москва

Российский институт стандартизации

202X

Предисловие

1 РАЗРАБОТАН Федеральной службой по техническому и экспортному контролю (ФСТЭК России), Федеральным государственным бюджетным учреждением науки Институт системного программирования им. В.П. Иванникова Российской академии наук (ИСП РАН), Акционерным обществом «Лаборатория Касперского» (АО «Лаборатория Касперского»), 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 №

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 (www.rst.gov.ru)

Содержание

Введение

Настоящий стандарт направлен на достижение целей, связанных с предотвращением появления, выявлением и устранением уязвимостей и недекларированных возможностей в программном обеспечении, и содержит общие требования, предъявляемые к разработчикам 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 - _____

1 Область применения

Настоящий стандарт устанавливает общие требования к содержанию и порядку выполнения работ, связанных с созданием безопасного программного обеспечения, а также формированием и поддержанием среды обеспечения оперативного устранения выявленных недостатков и уязвимостей программного обеспечения.

Настоящий стандарт предназначен для разработчиков и производителей программного обеспечения, а также для организаций, выполняющих оценку соответствия процессов разработки программного обеспечения и предъявляемых к ним требований положениям настоящего стандарта.

Примечание – В настоящем стандарте далее по тексту разработчики и производители программного обеспечения 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 требованиям настоящего стандарта, задаются требования к отдельным технологиям, применяемым в процессах разработки.

2 Нормативные ссылки

Указанные в данном разделе документы являются необходимыми для применения настоящего стандарта. Для датированных ссылок используют только указанное издание. Для недатированных ссылок – последнее издание со всеми изменениями и дополнениями.

ГОСТ 19.101—77 Единая система программной документации. Виды программ и программных документов

ГОСТ Р 50922 Защита информации. Основные термины и определения

П р и м е ч а н и е – 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.

3 Термины и определения

В настоящем стандарте применены термины и определения, приведенные в ГОСТ Р 50922, а также другие.

3.1

безопасность информации: Состояние защищенности информации, при котором обеспечены ее конфиденциальность, доступность и целостность.

[ГОСТ Р 50922—2006, статья 2.4.5]

3.2 **безопасное программное обеспечение:** Программное обеспечение, разработанное с использованием совокупности мер, направленных на предотвращение появления и устранение уязвимостей программы.

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

3.4 **документация разработчика программного обеспечения:** Совокупность программных документов, предназначенных для организации работ по созданию программного обеспечения, выполняемых в рамках процессов жизненного цикла программного обеспечения, и/или подтверждения соответствия требованиям настоящего стандарта.

Примечание – К программным относятся документы, содержащие сведения, необходимые для разработки, изготовления, сопровождения и эксплуатации программ.

3.5

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

[ГОСТ Р 51904—2002, статья 3.17]

3.6 **интерфейс программного обеспечения:** Способ взаимодействия между программным обеспечением и другим программным обеспечением или субъектами доступа.

3.7

компонент программного обеспечения: Программа, рассматриваемая как единое целое, выполняющая законченные функции и применяемая самостоятельно или в составе комплекса.

[ГОСТ 19.101-77, статья 1.2]

3.8 **недостаток программы:** Любая ошибка, допущенная в ходе проектирования или реализации программы, которая, в случае ее неисправления, может являться причиной уязвимости программы.

3.9

поверхность атаки: Множество подпрограмм (функций, модулей) программного обеспечения, обрабатывающих данные из интерфейсов, напрямую или косвенно подверженным потенциальному риску атаки.

[адаптировано из ГОСТ Р 56498-2015 п. 3.1.7]

3.10 **пользователь (программного обеспечения):** Лицо, применяющее программное обеспечение или участвующее в деятельности, прямо или косвенно зависящей от функционирования данного программного обеспечения.

3.11 **сборка программного обеспечения:** Процесс построения из исходных модулей программного обеспечения программных модулей, готовых к выполнению или интерпретации, и/или библиотек.

3.12

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

[адаптировано из ГОСТ Р 54593—2011, п. 3.13]

3.13 **система сборки программного обеспечения:** Совокупность программных средств и конфигурационных файлов, которая позволяет выполнить сборку программного обеспечения.

3.14

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, статья 3.62]

3.15 **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 реального выполнения 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.

3.16 **тестирование на проникновение:** Вид работ по выявлению (подтверждению) уязвимостей программы, основанный на моделировании (имитации) действий потенциального нарушителя.

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

3.18 **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 или результатом наличия ошибок проектирования или реализации.

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

3.20 **фаззинг-тестирование программы:** Вид работ по исследованию программы, направленный на оценку ее свойств и основанный на передаче программе случайных или специально сформированных входных данных, отличных от данных, предусмотренных алгоритмом работы программы.

3.21 **экспертиза исходного кода программы:** Вид работ по выявлению недостатков программы (потенциально уязвимых конструкций) в исходном коде программы, 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 создание предпосылок:

– к снижению вероятности возникновения уязвимостей в разрабатываемом ПО;

– к снижению количества потенциальных уязвимостей ПО;

– к снижению уровня критичности потенциальных уязвимостей ПО;

– к снижению ущерба от потенциальных уязвимостей ПО;

– к оперативному устранению возникающих уязвимостей в ПО.

4.2 В настоящем стандарте общие требования к разработке безопасного ПО представлены в виде описания процессов, реализация которых направлена на достижение результатов по разработке 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 в условиях использования стандартизированных моделей жизненного цикла, таких как ГОСТ 34.601-90.

4.4 В настоящем стандарте представлены общие требования к процессам разработки безопасного ПО. Методика оценки соответствия реализации процессов разработки безопасного ПО требованиям является предметом рассмотрения отдельного национального стандарта.

4.5 Внедрение и реализация процессов разработки безопасного ПО подразумевает 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 сущностей: требование, рекомендация или допустимое действие. С целью подчеркнуть различие между разными формами условий к реализации процессов разработки безопасного ПО, в настоящем стандарте используются вспомогательные глаголы «должен», «следует» и «может». Глагол «должен» использован для выражения условия, требуемого для соответствия требованию; «следует» — для выражения рекомендации по реализации, «может» — для того, чтобы отразить возможные направления допустимых действий.

4.8 Процессы разработки безопасного ПО изложены в следующей структуре: наименование процесса (наименование соответствующего подраздела раздела 5), цели, требования к реализации, состав и содержание документированных свидетельств, критерии положительного заключения о реализации требований к процессу (пункты соответствующего подраздела).

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 «Состав и содержание документированных свидетельств» включает формулировки наименования свидетельств выполнения требований к реализации процессов разработки безопасного ПО и их содержание. Под документированным свидетельством в настоящем стандарте понимаются сведения, подтверждающие выполнение соответствующего процесса, в электронном или физическом виде.

4.12 Наиболее общими для большинства рассматриваемых процессов документированными свидетельствами разработчика в настоящем стандарте являются регламенты. В общем случае, регламент осуществления процесса разработки безопасного ПО должен быть разработан, утвержден и 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 или объединены в рамках общего документа.

4.13 Пункт «Критерии положительного заключения о реализации требований к процессу» включает формулировки критериев выполнения требований к реализации процессов разработки безопасного ПО, применяемых при дальнейшем выполнении процедур оценки.

4.14 При разработке ПО комплексом программных систем или интегрированной программной системой 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 должны быть интегрированы с применяемым комплексом систем в целях обеспечения регулярности и своевременности проверок кода, прослеживаемости устранения выявленных ошибок.

4.15 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 «следует» и «может».

4.16 Предъявление требований настоящего стандарта к ПО, разрабатываемому в рамках научно-исследовательских и опытно-конструкторских работ, допускается только в форме явного перечисления в техническом задании процессов, подлежащих реализации. Рекомендации по формированию совокупности процессов, подлежащих реализации разработчиком ПО, приведены в Приложении В. Допускается задавать требования настоящего стандарта не в полном объеме, указывать условия применимости каждого из реализуемых процессов, требований к ним, состава и содержания документированных свидетельств и критериев положительного заключения о реализации требований к процессам.

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, которые реализованы разработчиком в области 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 и утверждать у руководства план реализации процессов разработки безопасного ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.1.3.1, 5.1.3.2.

5.1.2.4 Определять область применения процессов разработки безопасного ПО.

5.1.3 Состав и содержание документированных свидетельств

5.1.3.1 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 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 сотрудников.

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.

5.1.4 Критерии положительного заключения о реализации требований к процессу

5.1.4.1 Анализ текущего статуса реализации процессов, которые реализованы разработчиком в области разработки безопасного ПО, выполняется.

5.1.4.2 Анализ потребностей в ресурсах, необходимых для реализации процессов разработки безопасного ПО, выполняется.

5.1.4.3 План реализации процессов разработки безопасного ПО разработан и утвержден, учитывает сроки реализации процессов, этапы реализации процессов, выполняемые на каждом этапе мероприятия, необходимые ресурсы (материальные и людские) для его реализации.

5.1.4.4 Определена область применения процессов разработки безопасного ПО, описание области применения процессов разработки безопасного ПО содержит обоснованный состав ПО (версии, модули, компоненты, функциональные подсистемы и т.п.), в отношении которого должны быть реализованы процессы разработки безопасного ПО.

5.2 Обучение сотрудников

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 безопасной разработки ПО.

5.2.1.3 Организация постоянного обучения сотрудников типовым практикам безопасной разработки ПО с учетом актуальных потребностей.

5.2.1.4 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.2.3.1.

5.2.2.3 Проводить обучение сотрудников.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.2.3.2.

5.2.2.4 Вести учет обучения сотрудников.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.2.3.2, 5.2.3.3.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.2.3.2.

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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.2.3.1.

5.2.3 Состав и содержание документированных свидетельств

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 должен включать:

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

5.2.3.3 Документированные свидетельства прохождения обучения включают (в зависимости от учебной программы, курса) свидетельства, дипломы, отчеты обучающих платформ и иные свидетельства, подтверждающие прохождение сотрудником обучения.

5.2.3.4 Документированные свидетельства учета обучения сотрудников должны содержать информацию о сотрудниках, прошедших обучение, пройденных программах (курсах) и результатов 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 Документированные свидетельства учета обучения сотрудников должны содержать информацию о сотрудниках, мероприятиях по повышению осведомленности 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.

5.2.4 Критерии положительного заключения о реализации требований к процессу

5.2.4.1 Анализ существующих (доступных для анализа) практик, документов, обучающих курсов и тренингов по разработке безопасного ПО и их применимости для разработчика проводится.

5.2.4.2 План обучения разработан, включает информацию о сотрудниках, сроках и программах (курсах, тренингах) их обучения, учитывает потребности разработчика в части используемых средств и технологий разработки.

5.2.4.3 Обучение сотрудников запланировано и выполняется в соответствии с планом обучения, по результатам обучения выдаются свидетельства, дипломы, отчеты обучающих платформ и иные свидетельства, подтверждающие прохождение сотрудником обучения.

5.2.4.4 Ведется учет обучения сотрудников. В свидетельствах учета обучения представлена требуемая информация.

5.2.4.5 Установлены критерии необходимости пересмотра программ обучения (курсов, тренингов и т.п.).

5.2.4.6 Проводятся мероприятия по повышению осведомленности 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.

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.

5.3.2 Требования к реализации

5.3.2.1 Разработать процедуру управления требованиями безопасности ПО.

5.3.2.2 Предъявлять к ПО требования безопасности.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.3.3.1.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.3.3.2.

5.3.3 Состав и содержание документированных свидетельств

5.3.3.1 Документированная процедура управления требованиями безопасности ПО должна 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;
- предполагаемые сроки реализации;
- сведения о сотрудниках (подразделениях), предъявивших требования;
- сведения о сотрудниках (подразделениях), принявших требования к реализации.

5.3.3.3 Свидетельство регистрации предъявленных требований безопасности ПО должно 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 следует 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 пп. 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.

5.3.4 Критерии положительного заключения о реализации требований к процессу

5.3.4.1 Процедура управления требованиями безопасности ПО разработана, документированные свидетельства содержат требуемую информацию.

5.3.4.2 Сформирован набор требований безопасности ПО, содержащий требуемую информацию.

5.3.4.3 Набор требований безопасности ПО пересматривается на основе выполнения критериев пересмотра – с установленной периодичностью или при наступлении определенных событий.

5.3.4.4 Осуществляется учет предъявленных требований безопасности ПО, контролируется однозначность трактования и непротиворечивость набора требований безопасности ПО.

5.4 Управление конфигурацией программного обеспечения

5.4.1 Цели

5.4.1.1 Обеспечение управления конфигурацией ПО.

5.4.2 Требования к реализации

5.4.2.1 Разработать и утвердить регламент управления конфигурацией в рамках жизненного цикла ПО.

5.4.2.2 Контролировать реализацию изменений ПО, документации на ПО, других элементов, подлежащих отслеживанию в рамках управления конфигурацией ПО.

5.4.3 Состав и содержание документированных свидетельств

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 Свидетельство реализации управления изменениями ПО, документации на ПО в рамках управления конфигурацией ПО должно отображать факты изменения ПО (модулей ПО), документации на ПО.

5.4.4 Критерии положительного заключения о реализации требований к процессу

5.4.4.1 Регламент управления конфигурацией ПО разработан и утвержден.

5.4.4.2 Контролируются факты изменений ПО, документации и иных элементов, подлежащих отслеживанию в рамках управления конфигурацией 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.

П р и м е ч а н и е – Управление недостатками и запросами на изменение ПО способствует систематическому устранению ошибок кодирования, отклонений от заданных требований и корректировку требований в необходимых случаях путем осуществления запросов на изменение ПО.

5.5.2 Требования к реализации

5.5.2.1 Разработать и утвердить регламент управления недостатками ПО.

5.5.2.2 Разработать и утвердить 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 для управления недостатками и запросами на изменение разрабатываемого ПО средства автоматизации.

П р и м е ч а н и е – В качестве средств автоматизации следует использовать системы управления изменениями, системы управления задачами, системы контроля версий и т.п. При этом следует обеспечивать взаимосвязь (перекрестные ссылки) между такими системами при исправлении недостатков.

5.5.3 Состав и содержание документированных свидетельств

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 о 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 Свидетельство реализации управления недостатками ПО должно содержать зафиксированные факты изменений, связанных с недостатками, включающие следующую информацию:

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

5.5.3.4 Свидетельство реализации управления запросами на изменение ПО должно содержать следующую информацию:

- уникальный идентификатор запроса на изменение ПО;
- краткая характеристика запроса на изменение ПО;
- версия ПО (модуля ПО) к которому относится запрос на изменение;
- приоритет выполнения действий с запросом на изменение ПО;
- текущий статус обработки запроса на изменение ПО.

5.5.4 Критерии положительного заключения о реализации требований к процессу

5.5.4.1 Регламент управления недостатками ПО разработан и утвержден.

5.5.4.2 Регламент управления запросами ПО разработан и утвержден.

5.5.4.3 Контролируются факты изменений, связанных с недостатками ПО, в рамках жизненного цикла ПО.

5.5.4.4 Контролируются факты реализации запросов на изменение в рамках жизненного цикла ПО.

5.6 Разработка, уточнение и анализ архитектуры программного обеспечения

5.6.1 Цели

5.6.1.1 Создание условий для снижения количества потенциальных уязвимостей 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 потенциальных уязвимостей.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.3.3.2.

5.6.2.2 Выполнить первичное проектирование архитектуры ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.3.3.1, 5.5.3.1.

5.6.2.3 Установить критерии необходимости уточнения архитектуры ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2.

5.6.2.4 Выполнить уточнение архитектуры ПО в процессе разработки кода и его изменений (например, имплементации кода) с установленной периодичностью или при наступлении определенных событий.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 5.5.3.3.

5.6.3 Состав и содержание документированных свидетельств

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 к проектированию архитектуры ПО (например, инкапсуляция, инверсия зависимостей, уникальность, разделение задач, применение заимствованных компонентов 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 и т.п.

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 пп. 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.

5.6.4 Критерии положительного заключения о реализации требований к процессу

5.6.4.1 Приняты требования к принципам проектирования архитектуры ПО на основе требований безопасности к ПО.

5.6.4.2 Архитектура ПО с учетом требований безопасности спроектирована.

5.6.4.3 Установлены критерии необходимости уточнения архитектуры ПО.

5.6.4.4 Выполняется уточнение архитектуры ПО в процессе разработки кода с установленной периодичностью или при наступлении определенных событий.

5.7 Моделирование угроз и разработка описания поверхности атаки

5.7.1 Цели

5.7.1.1 Создание условий для снижения количества потенциальных уязвимостей, 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 и верификация модели угроз 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).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2.

5.7.2.2 Выполнить первичное описание поверхности атаки.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2.

5.7.2.3 Сформировать перечень целей (функциональных подсистем, программных модулей 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 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 сканирования интерфейсов ПО (локальных и сетевых интерфейсов взаимодействия с ПО (модулями ПО) пользователя и взаимодействий модулей ПО между собой, средой функционирования и внешними объектами при их наличии).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.7.3.1, 5.7.3.3, 5.7.3.4.

Примечание – Используемые методы сетевого сканирования способствуют получению информации об узлах сети, именах устройств, IP-адресах, операционных системах, запущенных программах и службах, именах пользователей, группах и открытых портах.

5.7.2.7 Уточнять перечень целей (функциональных подсистем, программных модулей 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 для разработанного кода ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.7.3.1, 5.7.3.3, 5.7.3.4.

5.7.3 Состав и содержание документированных свидетельств

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, предполагаемые источники возникновения угрозы, предполагаемый объект воздействия, сведения о потенциальном нарушителе и возможные последствия реализации угрозы.

Примечание – При составлении перечня угроз безопасности и их описания следует учитывать положения ГОСТ Р 58412, а также угрозы безопасности информации Банка данных угроз безопасности информации ФСТЭК России, других источников (например, методологии STRIDE, 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) потенциальных угроз содержит перечень необходимых действий (доработок ПО, иных мер). Перечень мер по нейтрализации (снижению вероятности возникновения) потенциальных угроз должен быть приоритизирован с точки зрения критичности выявленных потенциальных угроз.

5.7.3.3 Описание поверхности атаки должно включать совокупность потенциальных областей воздействия на информационную 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 перечень функциональных подсистем, программных модулей 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 пп. 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 пп. 5.7.2.5, должно включать перечень функциональных подсистем, программных модулей ПО и их интерфейсов, составляющих поверхность атаки, актуальных для разработанного кода ПО.

5.7.3.7 Перечень целей, уточненный по результатам выполнения требований пп. 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.

5.7.4 Критерии положительного заключения о реализации требований к процессу

5.7.4.1 Разработана модель угроз ПО.

5.7.4.2 Разработан перечень мер по нейтрализации потенциальных угроз.

5.7.4.3 Перечень мер по нейтрализации потенциальных угроз приоритизирован.

5.7.4.4 Определена поверхность атаки.

5.7.4.5 Сформирован перечень целей для проведения дальнейших исследований безопасности.

5.7.4.6 Модель угроз ПО учитывает актуальные угрозы безопасности и уточняется с установленной периодичностью или при наступлении определенных событий.

5.7.4.7 Описание поверхности атаки актуально для разработанного кода и уточняется с установленной периодичностью или при наступлении определенных событий.

5.7.4.8 Выполняется анализ поверхности атаки методом сканирования интерфейсов ПО.

5.7.4.9 Перечень целей для проведения дальнейших исследований безопасности ПО уточняется при изменениях архитектуры ПО и поверхности атаки.

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.

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 кода ПО следует использовать программные средства автоматической проверки правил кодирования.

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

5.8.3 Состав и содержание документированных свидетельств

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 при написании кода (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 код ПО;

- рекомендации по использованию стандартов кодирования (языков программирования, собственной разработки), принятые разработчиком ПО, разработчиками языков программирования.

5.8.4 Критерии положительного заключения о реализации требований к процессу

5.8.4.1 Регламент оформления исходного кода и безопасного кодирования разработан и содержит требуемую информацию.

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

5.9.2.1 Разработать и утвердить регламент проведения экспертизы исходного кода ПО.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.9.3.1.

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

5.9.3 Состав и содержание документированных свидетельств

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

- базовые требования к экспертизе (количество участников; области кода, подлежащего экспертизе; используемые инструменты и т.д.);

- описание шаблонов проведения атак для типовых сценариев работы ПО (модулей ПО), подлежащих контролю при экспертизе.

5.9.3.2 Результаты экспертизы кода должны содержать следующие сведения:

- информацию о проанализированных модулях;

- перечень необходимых изменений;

- вопросы к частям кода, экспертиза которых затруднена и требует дополнительных разъяснений;

- предложения по улучшению.

5.9.4 Критерии положительного заключения о реализации требований к процессу

5.9.4.1 Регламент проведения экспертизы исходного кода ПО разработан, утвержден и содержит требуемую информацию.

5.9.4.2 Экспертиза исходного кода проводится в соответствии с требованиями.

5.10 Статический анализ исходного кода

5.10.1 Цели

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

5.10.2 Требования к реализации

5.10.2.1 Разработать и утвердить регламент проведения статического анализа исходного кода ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 5.10.3.1.

5.10.2.3 Определить конфигурацию и параметры настройки инструментов статического анализа.

Входными данными требования к реализации являются свидетельства, указанные в пп. 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.10.3.1, 5.10.3.2.

5.10.2.5 Осуществлять пересмотр конфигурации и параметров настройки инструментов статического анализа с установленной периодичностью и/или при выполнении установленных событий (изменениях в правилах сборки, применяемых статических анализаторах, получении информации о потенциальных уязвимостях и т.п.).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.10.3.1, 5.10.3.2.

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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.10.3.1, 5.10.3.2.

5.10.3 Состав и содержание документированных свидетельств

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;
- 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 пп. 5.10.2.5, должны обеспечивать выполнение требований регламента проведения статического анализа в части выполнения критериев их пересмотра.

5.10.4 Критерии положительного заключения о реализации требований к процессу

5.10.4.1 Регламент проведения статического анализа исходного кода ПО разработан, утвержден и содержит требуемую информацию.

5.10.4.2 Перечень инструментов статического анализа для каждого используемого в ПО языка программирования сформирован.

5.10.4.3 Конфигурации и параметры настройки инструментов статического анализа определены.

5.10.4.4 Статический анализ проводится в соответствии с установленными требованиями, все срабатывания инструментов статического анализа регистрируются, для установленных типов и уровней критичности ошибок выполняется разметка.

5.10.4.5 Пересмотр конфигурации и параметров настройки инструментов статического анализа осуществляется с установленной периодичностью и/или при выполнении установленных событий.

5.10.4.6 Статический анализ проводится в соответствии с установленными требованиями, все срабатывания инструментов статического анализа регистрируются, для установленных типов и уровней критичности ошибок выполняется разметка.

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 и утвердить регламент проведения динамического анализа кода ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 5.7.3.1, 5.7.3.3.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 5.7.3.1, 5.7.3.3, 5.7.3.4, 5.11.3.1.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 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.

Примечание – Используемые методы динамического анализа могут позволять осуществлять динамический анализ кода программы путем подачи заведомо некорректных входных данных, динамическим профилированием, 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 ПО с целью контроля устранения ошибок.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 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.

5.11.2.7 Проводить фаззинг-тестирование.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 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.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.2, 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.

5.11.3 Состав и содержание документированных свидетельств

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 тестированию, 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 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 ПО, которые необходимо подвергнуть динамическому анализу, включая фаззинг-тестирование, отвечающий требованиям регламента проведения динамического анализа.

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.

5.11.3.5 Отчеты по результатам проведения динамического тестирования 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.

5.11.4 Критерии положительного заключения о реализации требований к процессу

5.11.4.1 Регламент проведения динамического анализа кода ПО разработан, утвержден и содержит требуемую информацию.

5.11.4.2 Перечень инструментов динамического анализа, включая инструменты фаззинг-тестирования, сформирован и содержит необходимую информацию.

5.11.4.3 Перечень модулей ПО, которые необходимо подвергнуть динамическому анализу, включая фаззинг-тестирование, сформирован.

5.11.4.4 Сценарии проведения тестирования для каждого исследуемого модуля (компонента) ПО средствами динамического анализа, включая инструменты проведения фаззинг-тестирования, определены.

5.11.4.5 Динамический анализ проводится в соответствии с установленными требованиями, все срабатывания инструментов динамического анализа регистрируются, для установленных типов и уровней критичности ошибок выполняется анализ (обработка).

5.11.4.6 Проводится повторный динамический анализ модулей ПО в соответствии с установленными регламентом динамического анализа критериями.

5.11.4.7 Фаззинг-тестирование проводится в соответствии с установленными требованиями, факты аварийных завершений работы ПО в процессе фаззинг-тестирования анализируются (обрабатываются).

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 трансформациями кода.

5.12.2 Требования к реализации

5.12.2.1 Разработать и утвердить регламент безопасной сборки ПО.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 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 Состав и содержание документированных свидетельств

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

- критерии выбора инструментов сборки ПО;
- критерии приемки 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 Свидетельство соответствия инструмента из состава системы сборки ПО рекомендациям производителя по безопасному использованию должно содержать перечень выполненных рекомендаций производителя инструмента сборки ПО, с указанием конкретных параметров настроек и конфигураций.

5.12.4 Критерии положительного заключения о реализации требований к процессу

5.12.4.1 Регламент безопасной сборки ПО разработан, утвержден и содержит требуемую информацию.

5.12.4.2 Информация о сборочной среде содержит описание особенностей функционирования, перечень инструментов сборки ПО; для каждого используемого инструмента сборки ПО в документации зафиксированы его версия и конфигурация.

5.12.4.3 Для всех инструментов из состава системы сборки ПО выполнены рекомендации со стороны их производителей по безопасному использованию (при условии наличия таких рекомендаций).

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 и утвердить регламент обеспечения безопасности сборочной среды.

5.13.2.2 Зафиксировать описание воспроизводимых 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 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 политике безопасности сборочной среды.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.13.3.1, 5.13.3.2.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.13.3.1, 5.13.3.2.

5.13.2.6 Обеспечивать повторяемость сборки ПО (если применимо).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.13.3.1.

5.13.2.7 Обеспечивать защиту внешних каналов связи для обеспечения конфиденциальности информации, обрабатываемой в сборочной среде.

5.13.3 Состав и содержание документированных свидетельств

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 воспроизводимых 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);

- информацию об используемой конфигурации сборки ПО;

- информацию о шагах сборки ПО;

- информацию о событиях безопасности в соответствии с политикой безопасности сборочной среды.

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

5.13.3.6 В качестве свидетельства повторяемости 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.).

5.13.4 Критерии положительного заключения о реализации требований к процессу

5.13.4.1 Регламент обеспечения безопасности сборочной среды разработан, утвержден и содержит требуемую информацию.

5.13.4.2 Зафиксировано описание воспроизводимых результатов сборки ПО, прав доступа к сборочной среде и ролей пользователей, участвующих в процессе сборки ПО.

5.13.4.3 Разработана схема сборочной среды.

5.13.4.4 Обеспечивается регистрация выполняемых действий при сборке ПО в журналах аудита, журналы аудита хранятся способом, обеспечивающим их целостность; сроки хранения журналов аудита зафиксированы в политике безопасности сборочной среды.

5.13.4.5 Обеспечено хранение результатов сборки ПО в выделенном хранилище кода ПО.

5.13.4.6 Обеспечивается повторяемость сборки ПО (если применимо).

5.13.4.7 Обеспечивается защита внешних каналов связи для обеспечения конфиденциальности информации, обрабатываемой в сборочной среде.

5.14 Обеспечение целостности кода при разработке программного обеспечения

5.14.1 Цели

5.14.1.1 Обеспечение целостности и необходимого уровня конфиденциальности кода ПО.

5.14.2 Требования к реализации

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

Примечание – При разработке и реализации политики доступа к исходному коду ПО следует руководствоваться принципами минимизации привилегий и разделения полномочий.

5.14.2.2 Осуществлять управление доступом к исходному коду ПО на основе ролей и прав доступа.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.14.3.1.

5.14.2.3 Осуществлять контроль целостности собственного исходного кода.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.14.3.1, 5.14.3.2.

5.14.3 Состав и содержание документированных свидетельств

5.14.3.1 Политика доступа к исходному коду ПО и обеспечения его целостности должна 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 обязанности, права и роли при разработке ПО;

- описание используемых инструментов разграничения доступа.

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

5.14.4 Критерии положительного заключения о реализации требований к процессу

5.14.4.1 Политика доступа к исходному коду ПО и обеспечения его целостности разработана, утверждена и содержит требуемую информацию.

5.14.4.2 Управление доступом к исходному коду ПО осуществляется на основе ролей и прав доступа в соответствии с моделью управления доступом.

5.14.4.3 Контроль целостности собственного исходного кода осуществляется.

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, ключи шифрования, цифровые подписи и т.п.).

5.15.2 Требования к реализации

Обязательных требований по обеспечению информационной безопасности используемых секретов не предъявляется. Разработчик может выполнять действия, указанные в пп. 5.15.2.1 и 5.15.2.2, и иные действия, направленные на обеспечение безопасного использования секретов.

5.15.2.1 Разрабатывать регламент использования секретов.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2.

5.15.2.2 Использовать секреты.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.15.3.1.

5.15.2.3 Проверять код и конфигурации 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 Использовать систему управления секретами.

5.15.3 Состав и содержание документированных свидетельств

5.15.3.1 Регламент использования секретов может содержать следующие сведения:

- 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 (цифровой подписи, электронной цифровой подписи) исполняемого кода ПО;

- порядок подписи (цифровой подписи, электронной цифровой подписи) исходного кода.

5.15.4 Критерии положительного заключения о реализации требований к процессу

5.15.4.1 Регламент использования секретов разработан, утвержден и содержит требуемую информацию.

5.15.4.2 Разработано описание реализации процедуры использования секретов.

5.15.4.3 Код и конфигурации ПО для вносимых изменений проверяются на предмет включения секретов.

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 стороннего кода в коде ПО разработчика.

5.16.2 Требования к реализации

5.16.2.1 Разработать и утвердить регламент композиционного анализа.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2.

5.16.2.2 Формировать перечень зависимостей ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2, 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2, 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 (компонентов, модулей ПО).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2, 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.16.3.1, 5.16.3.2, 5.16.3.3, 5.16.3.4, 5.16.3.5.

5.16.3 Состав и содержание документированных свидетельств

5.16.3.1 Регламент композиционного анализа должен содержать следующие сведения:

- правила отслеживания уязвимостей для заимствованных компонентов, участвующих в сборке ПО;

- правила проведения анализа заимствованных компонентов на предмет наличия известных уязвимостей;

- положения о назначении ответственных лиц за выполнение 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 проприетарного кода:

- результаты анализа применимости и реализуемости уязвимости;

- результаты обращения к поставщику (разработчику) уязвимых модулей по поводу их обновления;

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

б) для кода с открытыми исходными текстами:

- результаты анализа применимости и реализуемости уязвимости;

- результаты попыток обновления зависимых компонентов, в случае невозможности обновления путем обновления версии – применения собственного механизма исправления.

5.16.4 Критерии положительного заключения о реализации требований к процессу

5.16.4.1 Регламент композиционного анализа разработан, утвержден и содержит требуемую информацию.

5.16.4.2 Перечень зависимостей ПО сформирован.

5.16.4.3 Актуальность перечня зависимостей ПО контролируется.

5.16.4.4 Анализ заимствованных компонентов, составляющих поверхность атаки, на предмет наличия известных уязвимостей при сборке (непосредственно перед сборкой) ПО (компонентов, модулей ПО) проводится.

5.16.4.5 Корректирующие воздействия по результатам анализа уязвимостей в зависимостях ПО выполняются.

5.17 Проверка кода на предмет внедрения вредоносного кода через цепочки поставок

5.17.1 Цели

5.17.1.1 Создание условий для снижения рисков внедрения вредоносного кода 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).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.12.3.2, 5.16.3.2.

5.17.2.2 Осуществлять контроль договорных обязательств со сторонними поставщиками.

Входными данными требования к реализации являются свидетельства, указанные в пп. 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2, 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).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1.

5.17.3 Состав и содержание документированных свидетельств

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 элементов инфраструктуры (компонентов 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.).

5.17.4 Критерии положительного заключения о реализации требований к процессу

5.17.4.1 Контроль зависящих от сторонних поставщиков элементов разработки выполняется.

5.17.4.2 Контроль договорных обязательств со сторонними поставщиками осуществляется.

5.17.4.3 Осуществляется выявление элементов инфраструктуры разработчика, воздействие на которые может повлиять на возникновение недекларированных возможностей в ПО. В случае отсутствия обязательств или гарантий со стороны поставщиков – принимаются соответствующие компенсирующие меры (изменение условий работы с поставщиком, пересмотр меры ответственности поставщика, штрафные санкции, отказ от поставщика, переход на альтернативного поставщика, резервирование собственных мощностей для замены поставщика и др.).

5.17.4.4 Выполняется анализ исходных текстов ПО, отчеты анализа исходных текстов ПО содержат необходимую информацию, перечень предсобранных компонентов сторонних поставщиков контролируется.

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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2, 5.3.3.2.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.18.3.1.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.18.3.1.

5.18.2.4 Регистрировать ход проведения тестирования.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.18.3.1.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.18.3.2, 5.18.3.3.

5.18.3 Состав и содержание документированных свидетельств

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 ПО, номера (идентификаторы) сборок ПО/модулей 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 регистрации хода проведения 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 конфигурации тестируемого ПО;
- 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, в котором выявлена ошибка;
- категорию ошибки;
- принятое решение;
- информацию о верификации ошибки;
- информацию об исправлении ошибки.

5.18.4 Критерии положительного заключения о реализации требований к процессу

5.18.4.1 План функционального тестирования разработан и содержит требуемую информацию.

5.18.4.2 Функциональное тестирование выполняется в соответствии с планом тестирования, по результатам тестирования разрабатываются отчеты, содержащие требуемую информацию.

5.18.4.3 По результатам тестирования разрабатываются отчеты, позволяющие идентифицировать выполненные функциональные тесты ПО на уровне модулей, компонентов, подсистем, всего ПО в целом.

5.18.4.4 Журналы регистрации хода проведения функционального тестирования содержат требуемую информацию.

5.18.4.5 Процесс исправления выявленных в ходе тестирования ошибок с использованием системы управления ошибками организован.

5.19 Нефункциональное тестирование

5.19.1 Цели

5.19.1.1 Установление регламента проведения нефункционального тестирования.

5.19.1.2 Верификация поверхности атаки, модели угроз и архитектуры безопасности.

5.19.1.3 Обнаружение ошибок и уязвимостей в коде ПО путем выполнения нефункциональных тестов, в том числе, имитирующих действия потенциального нарушителя.

5.19.2 Требования к реализации

Обязательных требований по проведению нефункционального тестирования не предъявляется. Разработчик может выполнять действия, указанные в пп. 5.19.2.1 – 5.19.2.4, и иные действия, направленные на реализацию нефункционального тестирования.

5.19.2.1 Разработать и утвердить регламент нефункционального тестирования.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2, 5.19.3.1.

5.19.2.3 Осуществлять верификацию поверхности атаки, модели угроз и архитектуры безопасности.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2, 5.19.3.1.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2, 5.19.3.1.

Примечание – В рамках нефункционального тестирования могут выполняться следующие проверки:

- сетевых взаимодействий ПО;
- локальных интерфейсов взаимодействия ПО;
- выполняемых привилегированных операций;
- работы с конфиденциальными данными;
- возможности нарушения функционирования ОС при выполнении своих функций;
- корректности выполнения файловых операций;
- безопасной реализации бинарных файлов;
- реализации системы управления секретами;
- реализации криптографических протоколов;
- работы системы развертывания продукта;
- реализации мер в ПО в соответствии с моделированием угроз;
- поведения ПО при подмене пользователя;
- возможности нарушения бизнес-логики программы;
- безопасности реализации функциональных возможностей аутентификации и авторизации;

- безопасности обработки данных, полученных от потенциального нарушителя;

- безопасности реализации клиентской и серверной частей ПО.

5.19.3 Состав и содержание документированных свидетельств

5.19.3.1 Регламент нефункционального тестирования может содержать следующие сведения:

- критерии выбора версий 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;

- описание шаблонов проведения атак для типовых сценариев работы ПО (модулей ПО).

5.19.3.2 Описание объекта нефункционального тестирования может включать следующие сведения:

- архитектуру ПО (модулей ПО);

- состав инфраструктуры развертывания;

- фактическую поверхность атаки;

- доступные и потенциально доступные интерфейсы;

- потенциальные уязвимости.

5.19.3.3 Результаты сравнения архитектуры ПО, модели угроз и описания поверхности атаки с полученными фактическими результатами, перечень необходимых изменений в указанных свидетельствах.

5.19.3.4 Отчет по результатам нефункционального тестирования может содержать следующую информацию:

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

5.19.4 Критерии положительного заключения о реализации требований к процессу

5.19.4.1 Регламент нефункционального тестирования разработан, утвержден и содержит требуемую информацию.

5.19.4.2 Разработано описание объекта нефункционального тестирования.

5.19.4.3 Верификация и уточнение поверхности атаки, модели угроз и архитектуры безопасности путем нефункционального тестирования выполняются.

5.19.4.4 Нефункциональное тестирование выполняется в соответствии с регламентом, по результатам тестирования разрабатываются отчеты, содержащие требуемую информацию.

5.20 Обеспечение безопасности при выпуске готовой к эксплуатации версии программного обеспечения

5.20.1 Цели

5.20.1.1 Организация независимой приемки ПО с целью недопущения ошибок и уязвимостей в коде ПО перед его предоставлением пользователям.

5.20.2 Требования к реализации

5.20.2.1 Разработать и утвердить 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.).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.20.3.1, 5.9.3.2, 5.10.3.4, 5.11.3.5, 5.16.3.5, 5.18.3.2.

5.20.2.3 Разработать и утвердить регламент обеспечения целостности ПО, передаваемого пользователям.

5.20.2.4 Обеспечивать возможность проверки пользователями целостности ПО.

5.20.3 Состав и содержание документированных свидетельств

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 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.

Примечание – В качестве меры, реализующей возможность проверки целостности ПО, рекомендуется применение цифровой подписи ПО.

5.20.3.4 Результаты проверки выполнения мер по обеспечению целостности ПО.

5.20.4 Критерии положительного заключения о реализации требований к процессу

5.20.4.1 Регламент приемки ПО разработан, утвержден и содержит требуемую информацию.

5.20.4.2 Анализ степени влияния на безопасность ПО неустраненных ошибок выполняется.

5.20.4.3 Регламент обеспечения целостности ПО, передаваемого пользователям, разработан, утвержден и содержит требуемую информацию.

5.20.4.4 Возможность проверки пользователями целостности ПО обеспечивается.

5.21 Безопасная доставка программного обеспечения пользователям

5.21.1 Цели

5.21.1.1 Обеспечение защиты ПО, в том числе документации ПО, от угроз, связанных с нарушением целостности, в процессе передачи ПО пользователю.

5.21.2 Требования к реализации

5.21.2.1 Разработать документацию безопасной доставки ПО пользователям.

5.21.2.2 Фиксировать версии поставляемого пользователям ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.21.3.1.

5.21.2.3 Организовать хранение копий версий поставляемого пользователям ПО.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.21.3.1.

5.21.2.4 Поставлять ПО вместе с эксплуатационной документацией.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.21.3.1.

5.21.3 Состав и содержание документированных свидетельств

5.21.3.1 Документация безопасной доставки ПО пользователям должна содержать следующие сведения:

- процедуры хранения копий версий поставляемого пользователям ПО;
- процедуры снятия копий (тиражирования) поставляемого пользователям ПО;

- процедуры доставки ПО (обновлений ПО, включая обновления безопасности);

- процедуры проверки подлинности ПО (обновлений ПО) пользователем.

5.21.3.2 Сведения о версии поставляемого пользователям ПО должны быть зафиксированы в 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).

5.21.3.4 Сведения о наличии поставляемой эксплуатационной документации на ПО должны быть зафиксированы в документации (в электронном виде или на физическом носителе).

5.21.4 Критерии положительного заключения о реализации требований к процессу

5.21.4.1 Документация безопасной доставки ПО пользователям разработана и содержит требуемую информацию.

5.21.4.2 Версии поставляемого пользователям ПО фиксируются.

5.21.4.3 Место хранения копий версий поставляемого пользователям ПО определено и идентифицируемо.

5.21.4.4 Эксплуатационная документация на ПО поставляется вместе с ПО.

5.22 Обеспечение поддержки программного обеспечения на этапе эксплуатации пользователями

5.22.1 Цели

5.22.1.1 Обеспечение технической поддержки ПО на этапе его эксплуатации с целью устранения выявляемых при эксплуатации ПО недостатков и уязвимостей, а также обновления ПО.

5.22.2 Требования к реализации

5.22.2.1 Разработать документацию технической поддержки.

5.22.2.2 Организовать работу службы технической поддержки.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.22.3.1.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.22.3.1.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 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.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.22.3.1.

5.22.3 Состав и содержание документированных свидетельств

5.22.3.1 Документация технической поддержки должна 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 Свидетельство наличия технической поддержки может определяться наличием должностей сотрудников технической поддержки в штатном расписании организации, документированными фактами оказания конкретных услуг по технической поддержке пользователей, иными фактами и документированными свидетельствами.

5.22.3.3 Свидетельство оповещения пользователей должно содержать информацию об используемых каналах взаимодействия с пользователями при выпуске обновлений (включая обновления безопасности) и необходимости их установки.

5.22.3.4 Свидетельство обучения специалистов службы технической поддержки может содержать информацию о пройденных семинарах, вебинарах, курсах или иную информацию, подтверждающую умения и знания специалистов службы технической поддержки работе с поставляемым ПО, его особенностям установки и функционирования, ограничениями и указаниями по эксплуатации.

5.22.3.5 Свидетельство информирования пользователей ПО о выявленных уязвимостях должно содержать информацию об используемых каналах взаимодействия с пользователями и способах реализации мер по их нейтрализации до разработки обновлений безопасности, устраняющих уязвимость.

5.22.4 Критерии положительного заключения о реализации требований к процессу

5.22.4.1 Документация технической поддержки разработана и содержит требуемую информацию.

5.22.4.2 Служба технической поддержки организована и функционирует.

5.22.4.3 Организовано оповещение пользователей о выпуске обновлений.

5.22.4.4 Организовано обучение специалистов службы технической поддержки.

5.22.4.5 Организована процедура информирования пользователей ПО о выявленных уязвимостях и способах реализации мер по их нейтрализации до разработки обновлений безопасности, устраняющих уязвимость.

5.23 Обеспечение реагирования на информацию об уязвимостях

5.23.1 Цели

5.23.1.1 Обеспечение выявления и устранения уязвимостей на этапе эксплуатации ПО.

5.23.2 Требования к реализации

5.23.2.1 Разработать и утвердить регламент реагирования на информацию об уязвимостях.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2.

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).

Входными данными требования к реализации являются свидетельства, указанные в пп. 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 применимости информации о найденных уязвимостях в ПО на предмет подтверждения наличия/отсутствия уязвимостей.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.23.3.2.

5.23.2.5 Осуществлять оценку актуальности и критичности потенциальной уязвимости с точки зрения безопасности ПО (в случае получения информации о потенциальных уязвимостях ПО из внешнего источника).

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.23.3.3.

5.23.3 Состав и содержание документированных свидетельств

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 поиска известных (подтвержденных) уязвимостей в общедоступных источниках информации об уязвимостях ПО, его программных компонентов и сред его функционирования;
- правила реагирования на информацию об уязвимостях;
- правила оценки актуальности и критичности потенциальной уязвимости с точки зрения безопасности ПО;
- периодичность проведения поиска известных (подтвержденных) уязвимостей в общедоступных источниках информации об уязвимостях ПО.

5.23.3.2 Свидетельство получения и обработки запросов от пользователей должно 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 Свидетельство анализа применимости информации о найденных уязвимостях в ПО должно 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).

5.23.3.4 Свидетельство оценки актуальности и критичности потенциальной уязвимости с точки зрения безопасности должно содержать следующие сведения:

- информацию об оценке актуальности уязвимости;

- информацию об оценке уровня критичности уязвимости ПО;

- вердикт по результатам анализа актуальности и критичности потенциальной уязвимости.

5.23.4 Критерии положительного заключения о реализации требований к процессу

5.23.4.1 Регламент реагирования на информацию об уязвимостях разработан, утвержден и содержит требуемую информацию.

5.23.4.2 Обработка поступающих запросов от пользователей с последующим анализом ошибок функционирования на предмет наличия уязвимостей выполняется, свидетельство получения и обработки запросов от пользователей содержит требуемые сведения.

5.23.4.3 Анализ применимости информации о найденных уязвимостях в ПО на предмет подтверждения наличия/отсутствия уязвимостей проводится, свидетельство анализа применимости информации о найденных уязвимостях в ПО содержит требуемые сведения.

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 конфигураций ПО до того, как они будут выявлены сторонними лицами и повлекут инциденты информационной безопасности.

5.24.2 Требования к реализации

Требования пп. 5.24.2.1 – 5.24.2.2 являются обязательными. Разработчику следует выполнять действия, указанные в требованиях пп. 5.24.2.3 – 5.24.2.4.

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

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.5.3.1, 5.7.3.1, 5.7.3.2.

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, его сторонних компонентов и сред функционирования.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.16.3.2, 5.24.3.1.

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

При реализации процесса поиска уязвимостей ПО при его эксплуатации разработчику ПО следует включать в состав проверок:

– статический анализ исходного кода, проводимый с привлечением специализированных методов статического анализа и/или другими настройками конфигурации;

– динамический анализ кода ПО, проводимый с расширенным набором датчиков срабатывания ошибок, более длительным и ресурсоемким фаззинг-тестированием.

Входными данными требования к реализации являются свидетельства, указанные в пп. 5.24.3.1.

Примечание 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.

Примечание 2 – Проверки кода ПО и 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 Состав и содержание документированных свидетельств

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.

5.24.4 Критерии положительного заключения о реализации требований к процессу

5.24.4.1 Для каждой версии эксплуатирующегося ПО на протяжении всего срока действия технической поддержки проверки выполняются в соответствии с определенным составом и периодичностью.

5.24.4.2 При выявлении уязвимостей обеспечивается реагирование в соответствии с утвержденным регламентом реагирования на поступающую информацию об уязвимостях.

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 Зафиксировать описание условий, при которых ПО (версию ПО) необходимо выводить из эксплуатации.

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

5.25.3 Состав и содержание документированных свидетельств

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

5.25.4 Критерии положительного заключения о реализации требований к процессу

5.25.4.1 Регламент вывода ПО из эксплуатации разработан и содержит требуемую информацию.

Приложение А

(справочное)

Сопоставление требований к процессам разработки безопасного программного обеспечения с мерами из ГОСТ Р 56939-2016

С целью обеспечения преемственности настоящей редакции стандарта редакции ГОСТ Р 56939-2016 в таблице А.1 представлено сопоставление мер по разработке безопасного программного обеспечения из ГОСТ Р 56939-2016 и требований к процессам настоящего стандарта. Сопоставление мер по разработке безопасного программного обеспечения из ГОСТ Р 56939-2016 и требований к процессам настоящего стандарта имеет справочный характер и не является основанием для невыполнения требований настоящего стандарта при наличии у разработчика реализованных мер из ГОСТ Р 56939-2016, т.к. настоящий стандарт имеет большую детализацию требований.

Таблица А.1 – Сопоставление мер по разработке безопасного программного обеспечения из ГОСТ Р 56939-2016 и требований к процессам настоящего стандарта

Требования к процессам разработки безопасного программного обеспечения (настоящая редакция стандарта)	Реализуемые меры по разработке безопасного обеспечения (по ГОСТ Р 56939-2016)
5.1 Планирование процессов разработки безопасного программного обеспечения	Меры отсутствуют
5.2 Обучение сотрудников	5.9 Меры по разработке безопасного программного обеспечения, реализуемые при процессе менеджмента людскими ресурсами
5.3 Формирование и предъявление требований безопасности к программному обеспечению	5.1 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении анализа требований к программному обеспечению
5.4 Управление конфигурацией программного обеспечения	5.7 Меры по разработке безопасного программного обеспечения, реализуемые в процессе менеджмента документацией и конфигурацией программы
5.5 Управление недостатками и запросами на изменение программного обеспечения	5.7 Меры по разработке безопасного программного обеспечения, реализуемые в процессе менеджмента документацией и конфигурацией программы
5.6 Разработка, уточнение и анализ архитектуры программного обеспечения	5.2 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении проектирования архитектуры программы

Окончание таблицы А.1

Требования к процессам разработки безопасного программного обеспечения (настоящая редакция стандарта)	Реализуемые меры по разработке безопасного обеспечения (по ГОСТ Р 56939-2016)
5.7 Моделирование угроз и разработка описания поверхности атаки	5.2 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении проектирования архитектуры программы
5.8 Формирование и поддержание в актуальном состоянии правил кодирования	5.3 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении конструирования и комплексирования программного обеспечения
5.9 Экспертиза исходного кода	5.3 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении конструирования и комплексирования программного обеспечения
5.10 Статический анализ исходного кода	
5.11 Динамический анализ кода программы	
5.12 Использование безопасной системы сборки программного обеспечения	5.3 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении конструирования и комплексирования программного обеспечения
5.13 Обеспечение безопасности сборочной среды программного обеспечения	5.8 Меры по разработке безопасного программного обеспечения, реализуемые при процессе менеджмента инфраструктурой среды разработки программного обеспечения
5.14 Обеспечение целостности кода при разработке программного обеспечения	5.8 Меры по разработке безопасного программного обеспечения, реализуемые при процессе менеджмента инфраструктурой среды разработки программного обеспечения
5.15 Обеспечение безопасности используемых секретов	Меры отсутствуют
5.16 Использование инструментов композиционного анализа	Меры отсутствуют
5.17 Проверка кода на предмет внедрения вредоносного кода через цепочки поставок	Меры отсутствуют
5.18 Функциональное тестирование	5.4 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении квалификационного тестирования программного обеспечения
5.19 Нефункциональное тестирование	5.3 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении конструирования и комплексирования программного обеспечения
5.20 Обеспечение безопасности при выпуске готовой к эксплуатации версии программного обеспечения	5.5 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении инсталляции программы и поддержки приемки программного обеспечения
5.21 Безопасная доставка программного обеспечения пользователям	5.5 Меры по разработке безопасного программного обеспечения, реализуемые при выполнении инсталляции программы и поддержки приемки программного обеспечения
5.22 Обеспечение поддержки программного обеспечения на этапе эксплуатации пользователями	5.6 Меры по разработке безопасного программного обеспечения, реализуемые при решении проблем в программном обеспечении в процессе эксплуатации
5.23 Обеспечение реагирования на информацию об уязвимостях	
5.24 Поиск уязвимостей в эксплуатирующемся программном обеспечении	
5.25 Обеспечение безопасности при выводе программного обеспечения из эксплуатации	Меры отсутствуют

Приложение Б

(справочное)

Инициализация процессов разработки безопасного программного обеспечения

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.

Б.1 Инициализация процессов разработки безопасного программного обеспечения

Б.1.1 Цели

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

Б.1.1.2 Подготовка к внедрению процессов разработки безопасного ПО.

Б.1.2 Требования к реализации

Б.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).

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

Б.1.2.1 Определить пути изменения и совершенствования реализованных ранее процессов с учетом полученного ранее опыта и развитием технологий и формализовать их в виде плана.

Входными данными требования к реализации являются свидетельства, указанные в пп. Б.1.3.1, Б.1.3.2.

Б.1.3 Состав и содержание документированных свидетельств

Б.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 не реализованных 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.

Б.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.

Б.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.

Б.1.4 Критерии положительного заключения о реализации требований к процессу

Б.1.4.1 Анализ текущего статуса реализации процессов, которые реализованы разработчиком в области разработки безопасного ПО, выполняется.

Б.1.4.2 Анализ потребностей в ресурсах, необходимых для реализации процессов разработки безопасного ПО, выполняется.

Б.1.4.3 Определены пути изменения и совершенствования реализованных процессов.

Приложение В

(справочное)

Рекомендации по формированию совокупности процессов, подлежащих реализации разработчиком безопасного ПО в рамках научно-исследовательских и опытно-конструкторских работ

При разработке макета ПО рекомендуется реализовывать следующие процессы:

- планирование процессов разработки безопасного программного обеспечения (подраздел 5.1);
- управление недостатками и запросами на изменение программного обеспечения (подраздел 5.5);
- моделирование угроз и разработка описания поверхности атаки (подраздел 5.7);
- формирование и поддержание в актуальном состоянии правил кодирования (подраздел 5.8).

При разработке модели ПО рекомендуется реализовывать следующие процессы:

- планирование процессов разработки безопасного программного обеспечения (подраздел 5.1);
- управление недостатками и запросами на изменение программного обеспечения (подраздел 5.5);
- моделирование угроз и разработка описания поверхности атаки (подраздел 5.7);
- формирование и поддержание в актуальном состоянии правил кодирования (подраздел 5.8).

При разработке экспериментального образца ПО рекомендуется реализовывать следующие процессы:

- планирование процессов разработки безопасного программного обеспечения (подраздел 5.1);
- формирование и предъявление требований безопасности к программному обеспечению (подраздел 5.3);
- управление конфигурацией программного обеспечения (подраздел 5.4);

- управление недостатками и запросами на изменение программного обеспечения (подраздел 5.5);

- разработка, уточнение и анализ архитектуры программного обеспечения (подраздел 5.6);

- моделирование угроз и разработка описания поверхности атаки (подраздел 5.7);

- формирование и поддержание в актуальном состоянии правил кодирования (подраздел 5.8);

- статический анализ исходного кода (подраздел 5.10);

- динамический анализ кода программы (подраздел 5.11);

- использование инструментов композиционного анализа (подраздел 5.16);

- функциональное тестирование (подраздел 5.18).

Библиография

[1] ГОСТ Р ИСО 9000-2015 Системы менеджмента качества. Основные положения и словарь

[2] ГОСТ Р 51904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию

УДК 004.005.354 ОКС 35.020

Ключевые слова: безопасное программное обеспечение, уязвимость программы, защита информации