МЕТОДЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СЛОЖНЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

Михалевич И.Ф.
Институт проблем управления им. В. А. Трапезникова РАН
mif-orel@mail.ru

Аннотация: В статье рассмотрены вопросы обеспечения безопасности программного обеспечения сложных информационно-управляющих систем, которые составляют основу национальных цифровых экономик и критических информационных инфраструктур. Предложена расширенная система критериев безопасности программного обеспечения, рассмотрены вопросы доверия к его безопасности. Предложен метод обеспечения безопасности программного обеспечения, основанный на интеграции требований информационной безопасности и технологической независимости программного обеспечения. Представлен пример реализации предложенной методологии обеспечения безопасности программного обеспечения объектов цифровых экономик и критических информационных инфраструктур.

Ключевые слова: безопасность программного обеспечения, доверие к безопасности программного обеспечения, информационно-управляющая система, критическая информационная инфраструктура, технологическая независимость программного обеспечения, цифровая экономика.

Введение

Сложные информационно-управляющие системы (ИУС), в том числе государственные, муниципальные и иные важные информационные (автоматизированные) системы, автоматизированные системы управления, информационно-телекоммуникационных сети, образуют основу национальных цифровых экономик и критических информационных инфраструктур (далее – цифровые экономики) [1-6]. Построение и развитие цифровых экономик связано с использованием ранее созданных (наследуемых) ИУС и созданием новых ИУС, их интеграцией в единые функциональные комплексы. Такие интегрированные ИУС цифровых экономик характеризует не только более сложная архитектура, но и разнородный состав программного обеспечения (ПО) и программно-аппаратно-программных средств (ПАС). Это создает дополнительные условия для появления и проявления уязвимостей ПО (ПАС) ИУС и ИУС в целом, требует большего внимания к вопросам обеспечения безопасности ПО (ПАС) АИУС. Нередки случаи, когда функции безопасности начинают встраиваться в ПО (ПАС) на конечных этапах разработки при тестировании и исправлении ошибок, а также, когда информация о ПО (ПАС) распространяется без контроля, необходимого для обеспечения безопасности ИУС. Такие ПО и ПАС нельзя признать безопасными, а методология их разработки и применения требует совершенствования. В частности, методология должна основываться на обновленных принципах безопасности, учитывающих изменение условий функционирования ИУС.

Условия, влияющие на безопасность ПО (ПАС) ИУС, можно описать следующим образом: «Жизненно важные встроенные системы - будь то медицинские устройства, автомобили, подключенные к Интернету, диспетчерский контроль и сбор данных (SCADA), промышленные системы управления (ICS) или другие системы - играют решающую роль в современном мире. По мере того, как все больше и больше этих систем подключаются к Интернету вещей, необходимость обеспечения надлежащей защиты этих систем от хакеров и кибератак становится все более очевидной» [7].

При таких обстоятельствах методология обеспечения безопасности ПО и ПАС (далее - ПО) должна быть чувствительна к опасным изменениям условий функционирования ИУС, что влечет уточнение критериев безопасности и подходов к оценке доверия к безопасности. Уточненная методология должна также предусматривать возможность управления безопасностью на всех стадиях жизненного цикла ПО, комплексного применения технических и организационных мер обеспечения безопасности ПО, в том числе совершенствования системы профессиональной подготовки разработчиков ПО, а также лиц, участвующих в развертывании, применении и сопровождении (технической поддержке) ПО и ИУС.

1 Развитие системы критериев безопасности программного обеспечения

Исходя из анализа терминологии [8-13], в общем смысле под безопасным ПО ИУС понимают ПО, применение которого не несет угрозы безопасности информации в ИУС [14]. Безопасность

информации в ИУС определяет триада критериев, которая характеризуют конфиденциальность, доступность и целостность информации в ИУС.

Безопасность ИУС может быть нарушена за счет эксплуатации уязвимостей как самой ИУС, так и ее ПО. Под уязвимостью ПО понимают любые недостатки или ошибки ПО, которые могут быть использованы для реализации угроз безопасности информации [8]. До последнего времени безопасным считалось ПО, которое было разработано с соблюдением мер безопасности. Такой подход неполно учитывал условия, при которых возникновение или проявление уязвимостей стало возможно и на последующих этапах (стадиях) жизненного цикла ПО. Подход был уточнен. Было принято, что для реализации угроз безопасности информации в ИУС могут быть использованы уязвимости, возникающие не только в связи с недостатками в процессах разработки ПО (уязвимости кода, архитектуры, многофакторные), но также в процессах развертывания ПО (уязвимости конфигурации, организационные, многофакторные), а также сопровождения ПО, его доработки и обновления (уязвимости кода, архитектуры, конфигурации, многофакторные). Поэтому теперь под безопасным понимают ПО, которое разработано с использованием совокупности мер, направленных на предотвращение возникновения уязвимостей и их устранение [14].

Сведения о уязвимостях и рекомендации по их устранению могут быть предоставлены компаниями, осуществляющими разработку и/или продажи ПО, сообществами разработчиков ПО, государственными организациями. Так, например, в РФ общедоступные сведения о уязвимостях публикуются в банке данных угроз безопасности информации ФСТЭК России [15]. Банк данных ФСТЭК России предоставляет возможность поиска уязвимостей по основным и дополнительным признакам. К основным признакам отнесены область происхождения уязвимости, тип недостатка ПО, место возникновения (проявления) уязвимости. Дополнительными признаками являются наименование ПО и его версия, тип аппаратной платформы, степень опасности уязвимости, язык программирования и другие.

Однако опубликованные сведения неполно отражают реальную картину о уязвимостях ПО, а меры по их устранению не всегда могут быть общедоступны в силу особенностей ИУС, национальных и коммерческих интересов. Это наглядно иллюстрирует многообразие существующих открытых баз данных уязвимостей [16-24], ни одна из которых не является полной. Некоторые, ранее популярные базы данных закрываются, как это произошло, например, с базой данных уязвимостей с открытым исходным кодом OSVDB (Open Source Vulnerability Database) [25].

Неполноту баз данных можно объяснить, в том числе национальными требованиями, обязывающими обеспечить конфиденциальность сведений о выявленных уязвимостях ПО [14]. На неполноту баз данных также влияют коммерческие интересы компаний, которые не всегда заинтересованы в распространении информации, которая может повлиять на рынок ПО.

На рис. 1 и 2 приведены сведения об общем количестве и количестве критических уязвимостей соответственно в ПО различных производителей (на 11.05.2019 г. согласно [15]).

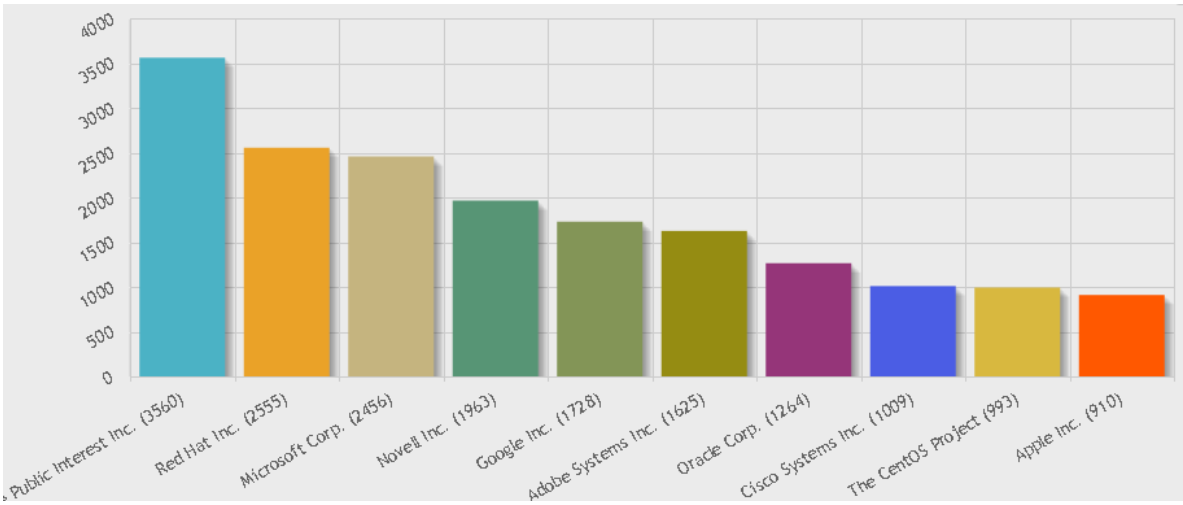

Рис. 1. Количество уязвимостей в программном обеспечении производителей

Методы обеспечения безопасности ПО должны учитывать национальные особенности, цели и приоритеты, а также подчиненные им меры по реализации внутренней и внешней национальной политики в сфере применения информационных и коммуникационных технологий. Так, например,

до 2030 г. реализация внутренней и внешней политики РФ в сфере применения информационных и коммуникационных технологий будет направлена на развитие информационного общества, формирование национальной цифровой экономики, обеспечение национальных интересов и реализацию стратегических национальных приоритетов [1].

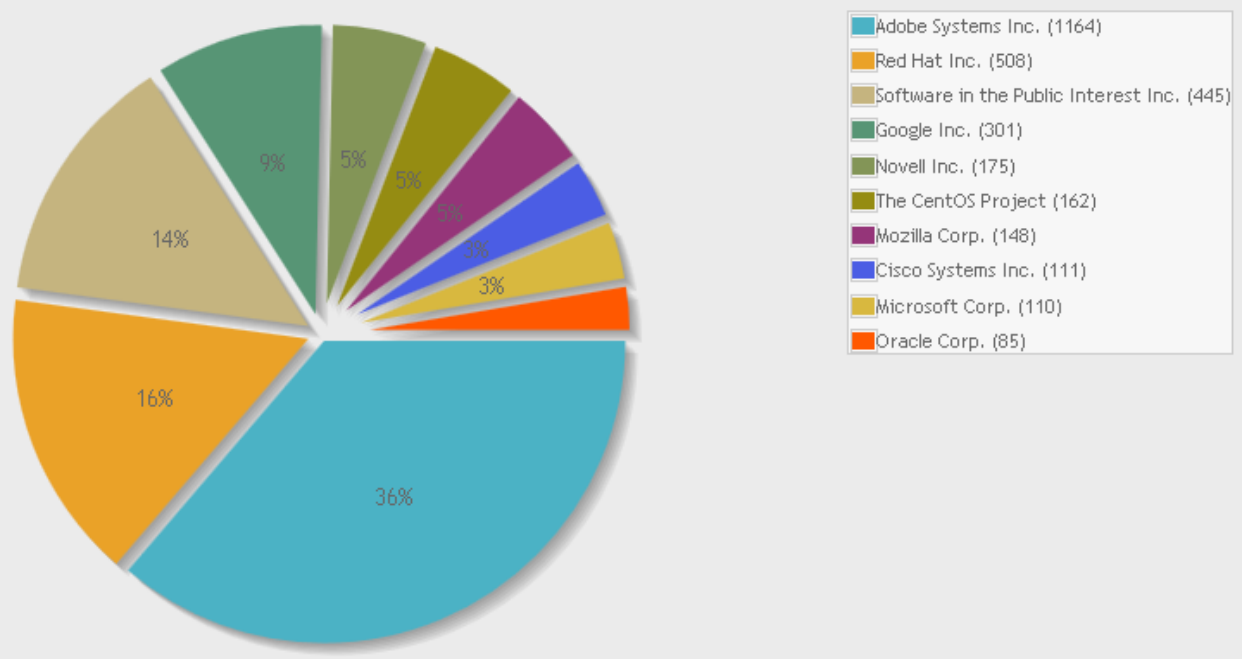

Рис. 2. Количество критических уязвимостей в программном обеспечении производителей

Изменения национальных целей и приоритетов неизбежно приводит к изменению (уточнению) критериев безопасности ПО, методов разработки и применения ПО, которое должно удовлетворять уточненным критериям и их показателям. Об этом свидетельствуют, например, Стратегия развития информационного общества в Российской Федерации на 2017 - 2030 годы, Указ Президента США от 11.05.2017 г. «Об усилении кибербезопасности федеральных сетей и критической инфраструктуры», другие многочисленные нормативные акты РФ, США, ЕС, Китая, других стран и союзных государств (союзов).

В частности, сформулированные в [1] национальные цели и приоритеты повлекли уточнение термина «безопасное ПО». В РФ безопасным признается ПО, сертифицированное на соответствие требованиям к информационной безопасности, устанавливаемым уполномоченными федеральными органами исполнительной власти РФ в области обеспечения безопасности противодействия техническим разведкам и технической защиты информации.

Более того, сегодня понятие «безопасность ПО» необходимо рассматривать совместно с понятием «технологическая независимость». В РФ технологически независимым является ПО, которое может быть использовано на всей территории РФ, обеспечено гарантийной и технической поддержкой российских организаций, не имеет принудительного обновления и управления из-за рубежа, модернизация которых осуществляется российскими организациями на территории РФ и которые не осуществляют несанкционированную передачу информации, в том числе технологической [1].

Совместное применение приведенных понятий приводит к расширенной системе критериев безопасности ПО сложных ИУС, образующих элементы цифровой экономики РФ. Несоблюдение требований расширенного вектора критериев исключает возможность признания ПО безопасным, его применение в составе сложных ИУС, таких, например, как значимые объекты критической информационной инфраструктуры, в РФ запрещено [26-28].

2 Уточнение условий доверия к безопасности программного обеспечения

Сложные ИУС подвержены риску нарушения безопасности вследствие многих факторов, возникающих в течение их жизненного цикла на фоне роста типов и видов атак. Увеличение числа потенциальных уязвимостей, сбоев и нарушений безопасности ИУС может быть вызвано, недооценкой угроз, неудовлетворительной организацией процессов разработки и эксплуатации ПО.

Угрозы, уязвимости и риски безопасности не статичны. Это требует управления ими в течение жизненного цикла ПО, иначе доверие к ПО может быть утрачено. В связи с этим методы разработки ПО необходимо интегрировать с методами менеджмента рисков безопасности и выявления уязвимостей и угроз, что обеспечит сохранение заданного уровня доверия к безопасности ПО, уверенность в том, что ПО реализует и не нарушает принятую политику безопасности ИУС при разработке ПО, его применении, доработке и технической поддержке (установке обновлений).

Владельцы ИУС должны быть не просто уверенными, а определенным образом доверять тому, что использование или развертывание ПО является безопасным, функционирование ПО создает надежные результаты, а меры и средства контроля и управления безопасностью ПО установлены и функционируют должным образом в динамике изменения условий функционирования ИУС. Безопасным считают ПО фактический уровень доверия которого равен целевому уровню доверия, который определяется организацией, использующей ПО [31]. Поэтому новым фактором, который необходимо дополнительно учитывать при принятии решения о доверии к безопасности ПО, является повышение риска нарушения безопасности ИУС при бесконтрольном распространении сведений о разработке и функционировании ПО за границы страны (союза стран).

Риски, связанные с трансграничным распространением данных, вызывают все большее беспокойство. Этим объясняется поставленная в РФ цель достижения технологической независимости в сфере информационных и телекоммуникационных технологий [1]. На всеобъемлющий характер рисков, связанных с трансграничной передачей данных указывает, в том числе, беспокойство Всемирного экономического форума. В [29] это выражено следующим образом: «Четвертая промышленная революция повлияет на масштаб конфликтов и их характер. Размывается грань между войной и миром, а также между тем, кто участвует в боевых действиях, а кто - нет. … Кибернетическая война представляет собой одну из самых серьезных угроз нашего времени. Киберпространство становится таким же театром военных действий, как в прошлом были земля, моря и воздух. … В результате этого не только снизится порог критериев наличия войны, но также станет менее выраженной грань между войной и миром, поскольку любые сети или подключенные устройства, от военных систем до гражданской инфраструктуры, такие как источники энергии, электрические сети, системы управления здравоохранением, движением или водоснабжением, могут быть взломаны и подвергнуты нападению. … Остается открытым вопрос о том, будет ли создан комплекс общих норм в отношении кибервойны, наподобие тех договоренностей, которые разработаны в отношении ядерных, биологических и химических вооружений. У нас отсутствует даже классификация, позволяющая нам прийти к согласию в отношении того, что считать нападением, а что - адекватным на него реагированием, какими способами эти действия могут производиться и кем. Один из показателей, который требуется учитывать для управления таким сценарием, состоит в том, какие именно данные пересекают границы».

Концепция определения целевого уровня доверия к безопасности ПО была основана на триаде критериев информационной безопасности (конфиденциальность, целостность и доступность) [30,31], что в современных условиях представляется недостаточным. При определении уровня доверия необходимо ввести дополнительные процедуры по выявлению и исключению из ПО конструкций и объектов, вызывающих непосредственные обращения к зарубежным компаниям (международным сообществам разработчиков) за подтверждениями и/или обновлениями ПО, отправку в их адрес технологической информации о ПО в процессе разработки, развертывания и применения.

До последнего времени требования по обеспечению безопасности ПО чаще всего рассматривались в контексте создания ИУС специального назначения, в которых реализовывалась дополнительная защита [32, 33]. Сегодня, в условиях IoT, Big Date, гибридных войн и других особенностей среды функционирования ИУС, создающих дополнительные возможности для нарушения безопасности ИУС, требования к безопасности ПО должны стать обязательными и трактоваться таким же образом, как и требования функциональных возможностей, качества и удобства в эксплуатации. Сами же требования к безопасности непременно должны быть согласованы с допустимыми пределами остаточного риска уязвимостей ПО, как это рассмотрено, например в [34,35].

Уязвимости приводят к неприемлемому риску для ИУС, являются следствием отсутствия или недостаточности мер и средств контроля и управления уязвимостями, которые проистекают от [8]:

– действующих субъектов, таких как программисты, которые создают плохие программы, пользователи, которые делают ошибки при использовании ПО, технические специалисты и разработчики, которые делают ошибки в процессе доработки и поддержки ПО;
– процессов, таких как неадекватные процедуры тестирования, плохой менеджмент проектов, недостаточное внимание к безопасности в течение процессов жизненного цикла, непредвиденное взаимодействие между программными средствами, пользователями и операторами, неадекватные процессы менеджмента изменений;
– технологического контекста, такого как плохо выбранная технологическая инфраструктура или продукты;
– особенностей, таких как неадекватное проектирование, уязвимости, обусловленные взаимодействиями в системе или ошибками в интерфейсах компонентов.

На безопасность ПО влияет целевая среда, поэтому вид и масштаб требований к безопасности ПО должны определяться с учетом рисков, которым оно подвергается исходя из бизнес, регулятивного и технологического контекстов [31].

Бизнес-контекст описывает конкретные риски, связанные со сферой деятельности владельца ИУС (финансовая организация, транспортная компания, государственное учреждение и т. д.).

Регулятивный контекст определяет конкретные риски, проистекающие из местоположения сферы действия владельца ИУС (права на интеллектуальную собственность и лицензирование, ограничения на криптографическую защиту, авторское право и т. д.).

Технологический контекст указывает на конкретные риски, проистекающие из технологий, используемых в деятельности владельца ИУС (реинжениринг, безопасность встроенных инструментальных средств, защита исходного кода программы, использование программы, заранее скомпилированной третьей стороной, тестирование безопасности, тестирование на проникновение, граничная проверка, проверка кода программы, среда информационно-коммуникационной технологии (ИКТ), в которой работает ПО, конфигурационные файлы и некомпилированные данные, привилегии операционной системы для инсталляции и/или функционирования, техническое обслуживание, безопасное распространение и т. д.). Технологический контекст охватывает технические спецификации ПО (функциональные возможности безопасности, безопасные компоненты, онлайновые платежи, надежные контрольные журналы, криптография, управление полномочиями и т. д.).

Зависимость от контекста определяет, что доверие к безопасности ПО не может быть безусловным. Так, например, на основе прежних оценок владелец ИУС может быть убежден, что используемое в ИУС ПО безопасно. Но это действительно только для конкретной ИУС в ее особом сочетании бизнес, регулятивного и технологического контекстов. Если меняется инфраструктура ПО или оно используется в другой ИУС, то изменившийся контекст может повлиять на требования к безопасности, а также целевой уровень доверия к безопасности ПО. При этом реализованные в ИУС меры, примененные средства контроля и управления безопасностью ПО могут уже неадекватно учитывать новые требования безопасности, а ПО может уже не быть безопасность.

3 Методы обеспечения безопасности программного обеспечения

Обеспечение безопасности ПО должно предусматривать применение мер и средств контроля, управления и измерений к ПО с целью осуществления менеджмента риска, возникающего в результате использования ПО в ИУС. Меры и средства контроля, управления и измерения должны применяться к самому ПО (его процессам, компонентам, программным средствам и результатам), его данным (конфигурационным данным, данным пользователей, данным организации) и ко всей технологии, процессам и действующим субъектам, вовлеченным в жизненный цикл ПО.

Безопасность ПО должна обеспечивать защиту критических данных, вычисляемых, используемых, хранимых и передаваемых ПО так, как определил владелец ИУС. Эта защита должна обеспечить уверенность не только в доступности, целостности и конфиденциальности данных, но также в том, что посредством передачи сведений о ПО не могут быть созданы условия для нарушения безопасности ИУС. Исходный, двоичный и исполняемый коды ПО относятся к критическим данным ИУС и нуждаются в защите наряду с другими данными ИУС.

Сфера действия безопасности ПО шире, чем сфера действия самого ПО. Сфера действия безопасности ПО показана на рис. 3 [31].

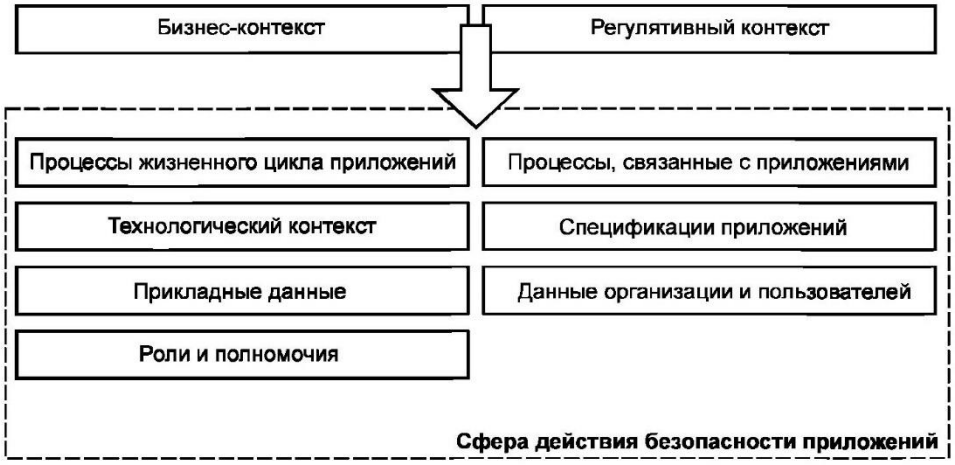

Рис. 3. Сфера действия безопасности ПО

Сравнение сфер действия ПО и безопасности ПО представлено в табл. 1 [31].

Таблица 1. Сферы действия ПО и безопасности ПО

Сферы	Действие	
	ПО	безопасности ПО
Данные организации и пользователей		+
Прикладные данные	+	+
Роли и полномочия	+	+
Спецификации приложений	+	+
Технологический контекст		+
Процессы, связанные с приложениями		+
Процессы жизненного цикла приложений		+
Бизнес-контекст		+
Регулятивный контекст		+

Для обеспечения безопасности ПО требования к безопасности должны быть определены и проанализированы для каждого этапа жизненного цикла ПО, подробно рассмотрены и управляемы на постоянной основе. Требования к безопасности ПО должны быть [36]:

- необходимыми;
- обобщенными;
- точно выраженными;
- последовательными;
- полными;
- лаконичными;
- осуществимыми;
- прослеживаемыми;
- поддающимися проверке.

Безопасность ПО должна обеспечиваться с этапа его разработки. Поэтому недопустимо включение в задания на разработку ПО нечетких требований, например, что «Разработчик должен обнаруживать все значимые риски безопасности для ПО». Встраивание функций безопасности целесообразно проводить не в конце разработки, а на каждом его шаге. Это обеспечивает тестирование функций безопасности совместно с основными функциями ПО и устранение совместно возникающих ошибок. На рис. 2 в общем виде представлен жизненный цикл разработки безопасного ПО (Security Development Lifecycle - SDL) согласно [31]. Сочетая целостный и практичный подход, данный метод вводит безопасность во все этапы процесса разработки ПО.

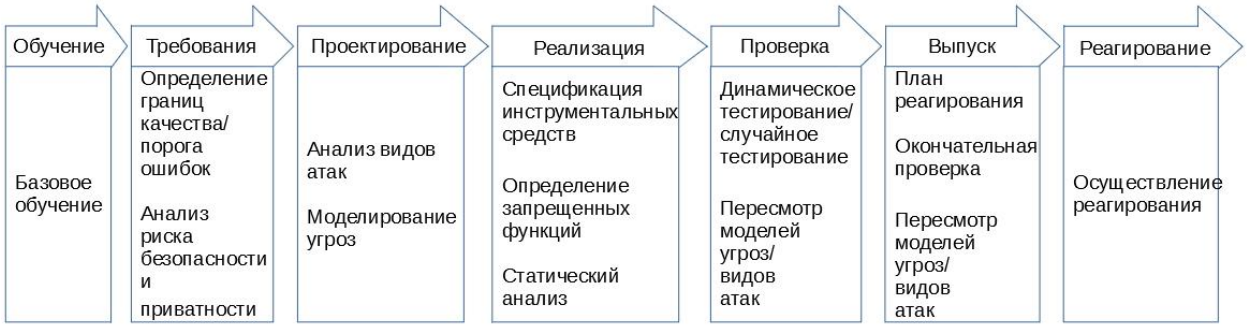

Рис. 4. Жизненный цикл разработки безопасного ПО

Развитие метода касается реализации процедур, направленных на выявление и исключение из ПО объектов, вызывающих его обращения к зарубежным компаниям (организациям) и международным сообществам разработчиков, а также устранение возможного влияния ПО на функции безопасности (проверки невлияния ПО на встроенные механизмы безопасности и средства защиты информации, взаимодействующие с ПО). Перечень групп мер по обеспечению безопасности ПО представлен в табл. 2 [14]. Описанные выше дополнительные процедуры должны быть применены при реализации каждой из приведенных в табл. 2 группах мер.

Таблица 2. Группы мер по обеспечению безопасности ПО

№	Меры, реализуемые:
1	при выполнении анализа требований к ПО
2	при выполнении проектирования архитектуры программ
3	при выполнении конструирования и комплексирования ПО
4	при выполнении квалификационного тестирования ПО
5	при выполнении инсталляции программы и поддержки приемки ПО
6	при решении проблем в ПО в процессе эксплуатации
7	в процессе менеджмента документацией и конфигурацией программы
8	в процессе менеджмента инфраструктурой среды разработки ПО
9	в процессе менеджмента людскими ресурсами

4 Пример реализации усовершенствованного метода обеспечения безопасности программного обеспечения

Усовершенствованный метод обеспечения безопасности ПО основан на сочетании требований качества, информационной безопасности и технологической независимости ПО. Схема реализации метода с применением технологии «воздушного зазора» представлена на рис. 5.

Рис. 5. Реализация метода обеспечения безопасности ПО с применение «воздушного зазора»

Метод был успешно применен при разработке аппаратно-программной платформы «Синтез-АПП» [35]. Разработка была завершена до принятия в 2015 г. в РФ требований к качеству ПО [37].

Но она не только соответствует этим требованиям, но и реализует расширенный набор характеристик качества ПО, включая технологическую независимость.

Уточненная модель качества безопасного ПО ИУС цифровой экономики представлена в табл. 3.

Таблица 3. Модель качества безопасного ПО

Характеристика (подхарактеристика)	
Функциональная пригодность	Готовность
Функциональная полнота	Отказоустойчивость
Функциональная корректность	Восстанавливаемость
Функциональная целесообразность	Защищенность
Уровень производительности	Конфиденциальность
Временные характеристики	Целостность
Использование ресурсов	Неподдельность
Потенциальные возможности	Отслеживаемость
Совместимость	Подлинность
Сосуществование	Сопровождаемость
Функциональная совместимость (интероперабельность)	Модульность
Удобство использования	Возможность многократного использования
Определимость пригодности	Анализируемость
Изучаемость	Модифицируемость
Управляемость	Тестируемость
Защищенность от ошибки пользователя	Переносимость
Эстетика пользовательского интерфейса	Адаптируемость
Доступность	Устанавливаемость
Надежность	Взаимозаменяемость
Завершенность	Технологическая независимость

Реализацию метода иллюстрирует рис. 6, на котором представлен состав основных компонент, включенных в состав четырех версий безопасных операционных систем семейства «Синтез-ОС».

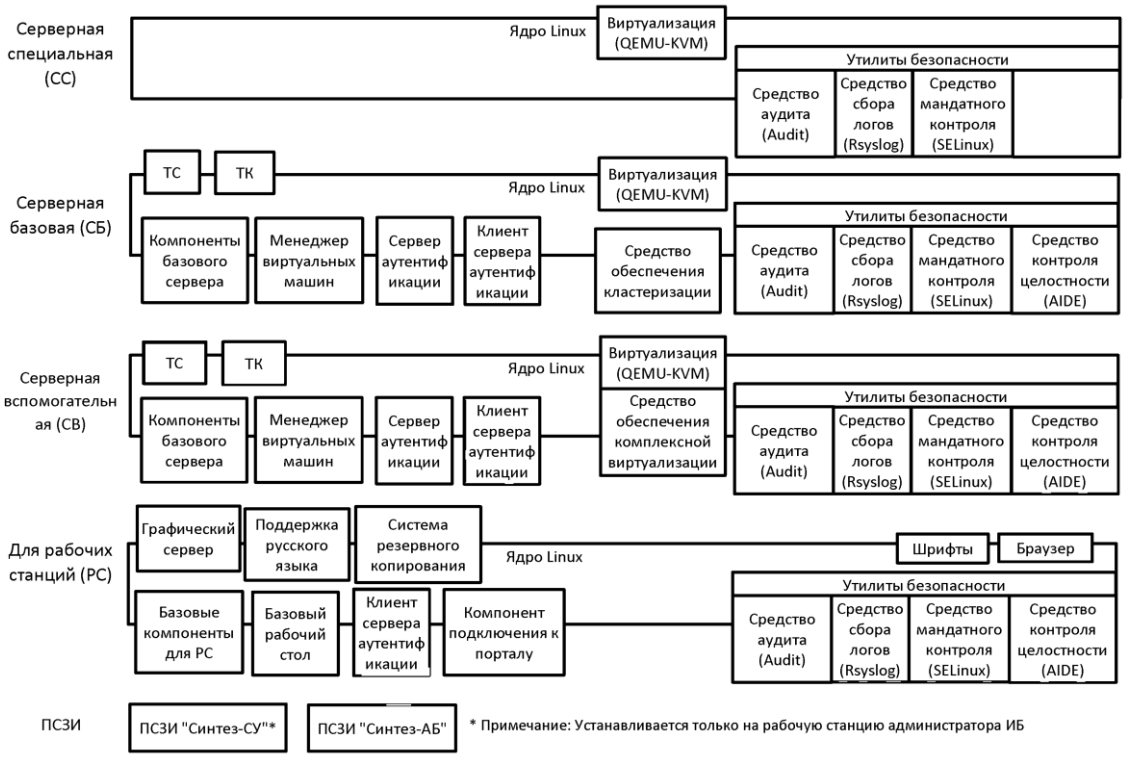

Рис. 6. Реализация метода на примере семейства безопасных операционных систем

Заключение

Вопросы обеспечения безопасности ПО требуют непрерывного внимания. При этом должны учитываться растущие риски, связанные с функционированием ПО в открытом киберпространстве. Обоснованную тревогу вызывает трансграничная передача данных о ПО, сведения о котором может использоваться для нарушения безопасности ИУС. В работе предложена расширенная система критериев безопасности ПО ИУС национальных цифровых экономик и критических информационных инфраструктур, рассмотрены вопросы доверия к его безопасности, разработан метод обеспечения безопасности ПО, основанный на интеграции требований информационной безопасности и технологической независимости ПО. Представлен пример реализации предложенной методологии обеспечения безопасности ПО ИУС цифровых экономик и критических информационных инфраструктур.

Литература

1. Стратегия развития информационного общества в Российской Федерации на 2017 - 2030 годы (утв. Указом Президента РФ от 09.05.2017 г. № 203).
2. Федеральный закон от 26.07.2017 г. № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации».
3. Executive Order No. 13636 of “Improving Critical Infrastructure Cybersecurity” of the USA President, February 12,2013.
4. Presidential Policy Directive 21 “Critical Infrastructure Security and Resilience” (PPD-21) of the USA President, February 12,2013.
5. National cyber strategy of the United States of America. - President of USA. - September 2018 - *https://www.whitehouse.gov/wp-content/uploads/2018/09/National-Cyber-Strategy.pdf* (дата обращения 02.06.2019).
6. Critical Information Infrastructures Protection approaches in EU. - Enisa.- July 2015. *https://resilience.enisa.europa.eu/enisas-ncss-project/CIIPApproachesNCSS.pdf* (дата обращения: 26.02.2018).
7. Security tenets for life critical embedded systems. - Department of Homeland Security of USA. - November 20, 2015. - *https://www.dhs.gov/sites/default/files/publications/security-tenets-lces-paper-1* (дата обращения 18.05.2019).
8. ГОСТ Р 56546-2015. Защита информации. Уязвимости информационных систем. Классификация уязвимостей информационных систем. - М.: Стандартинформ, 2015.
9. ГОСТ Р ИСО/МЭК 27000-2012. Информационная технология (ИТ). Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Общий обзор и терминология. - М.: Стандартинформ, 2012.
10. Банк данных угроз безопасности информации. Термины. - *http://bdu.fstec.ru/terms.* - (дата обращения 02.06.2019).
11. National Information Assurance (IA). Glossary. - CNSS Instruction No. 4009. - Committee on National Security Systems 26 April 2010. - *https://www.dni.gov/files/NCSC/documents/nittf/CNSSI-4009_National_Information_Assurance.pdf.* - (дата обращения 02.06.2019).
12. FISMApedia. Term: Vulnerability. - *http://fismapedia.org/index.php/Term:Vulnerability*. - (дата обращения 02.06.2019).
13. Threat and Risk Management. Glossary. - Enisa. - https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory/glossary#G52 (дата обращения 02.06.2019).
14. ГОСТ Р 56939-2016. Защита информации. Разработка безопасного программного обеспечения. Общие требования. Меры по разработке безопасного программного обеспечения. - М.: Стандартинформ, 2016.
15. База данных уязвимостей. - *http://bdu.fstec.ru/vul*. - (дата обращения 02.06.2019).
16. Common Vulnerabilities and Exposures (CVE). - https://cve.mitre.org (дата обращения 11.05.2019).
17. National Vulnerabilities Database (NVD). - http://nvd. nist.gov (дата обращения 11.05.2019).
18. United States Computer Emergency Readiness Team (US-CERT). - http://www.us-cert.gov (дата обращения 11.05.2019).
19. Symantec. SecurityFocus. BugTraq. - http://securityfocus.com (дата обращения 11.05.2019).

20. Secunia Research Community. https://secuniaresearch.flexerasoftware.com/community/research/ (дата обращения 11.01.2019).
21. Common Vulnerability Reporting Framework (CVRF). - http://www.icasi.org/cvrf. (дата обращения 11.05.2019).
22. Common Vulnerability Scoring System (CVSS). - http://www. first.org/cvss (дата обращения: 11.05.2019).
23. Common Platform Enumeration (CPE). - http://cpe.mitre.org (дата обращения: 11.05.2019).
24. Complete Vulnerability DataBase &Security Scanner. - https://vulners.com/kitploit/search?query=order: published. - (дата обращения 02.06.2019).
25. Open Source Vulnerabilities Data Base (OSVDB): FIN. - https://blog.osvdb.org (дата обращения 11.05.2019).
26. Требования по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации (утв. приказом ФСТЭК России № 239 от 25.12.2017).
27. Требования о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах (утв. приказом ФСТЭК России от 11.02.2013 г. № 17).
28. Требования к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды природной среды (утв. приказом ФСТЭК России № 31 от 14.03.2014).
29. Шваб. К. Четвертая промышленная революция. – М: Издательство «Эксмо», 2016. – 138 с.
30. ГОСТ Р 54581-2011/ISO/IEC/TR 15443-1:2005. Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 1. Обзор и основы. - - М.: Стандартинформ, 2012.
31. ГОСТ Р ИСО/МЭК 27034-1-2014. Информационная технология. Методы и средства обеспечения безопасности. Безопасность приложений. Часть 1. Обзор и общие понятия. - М.: Стандартинформ, 2015.
32. Проблемы безопасности программного обеспечения. Под ред. П.Д. Зегжда. - СПб.: Издательство СПбГТУ, 1995.
33. *Казарин О. В.* Безопасность программного обеспечения компьютерных систем. - М.: МГУЛ, 2003. - 212 с.
34. *Барабанов А.В., Марков А. С., Цирлов В.Л.* 28 магических мер разработки безопасного программного обеспечения // Вопросы кибербезопасности. 2015. № 5(13). - С. 2-10.
35. *Михалевич И.Ф.* Требования, принципы, практика создания отечественных аппаратно-программных платформ для автоматизированных систем в защищенном исполнении критической информационной инфраструктуры Российской Федерации // Интеллектуальные системы. Теория и приложения. - 2018. Том 22, вып. 4. – с. 11- 30.
36. ISO/IEC/IEEE 29148:2018. Systems and software engineering - Life cycle processes - Requirements engineering.
37. ГОСТ Р ИСО/МЭК 25010-2015. Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов. – М.: Стандартинформ, 2015.