28 МАГИЧЕСКИХ МЕР РАЗРАБОТКИ БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Барабанов А.В.[1], Марков А.С.[2], Цирлов В.Л.[3]

Представлены результаты исследования, посвященного созданию аппарата обоснованного формирования множества мер разработки безопасного программного обеспечения. Сформулировано понятие безопасного программного обеспечения, на основе обобщения стандартов и руководств в области разработки безопасного программного обеспечения сформулирован набор требований к процессу разработки безопасного программного обеспечения, рекомендуемых к реализации в жизненном цикле программного обеспечения (базовый набор требований к разработке безопасного программного обеспечения). Предложена оригинальная методика обоснованного формирования множества мер разработки безопасного программного обеспечения. Отражены практические аспекты использования сформулированного базового набора требований в процессе подготовки проекта национального стандарта в области разработки и производства безопасного программного обеспечения. Показана целесообразность гармонизации разрабатываемых нормативных требований и практических механизмов безопасности с методологиями международных стандартов серии ГОСТ Р ИСО 15408 и ГОСТ Р ИСО 12207, представлена информация о гармонизация разработанного проекта национального стандарта с существующими национальными стандартами. Проиллюстрирована взаимосвязь между разработанным базовым набором требований к разработке безопасного программного обеспечения и требований доверия к безопасности, установленных методологией «Общие критерии».

***Ключевые слова**: безопасное программное обеспечение, уязвимость программного обеспечения, Общие критерии, жизненный цикл разработки безопасного программного обеспечения, ИСО 15408, ИСО 12207*

Введение

Современные тенденции в области информационной безопасности характеризуется устойчивым ростом количества компьютерных атак, приводящих к снижению уровня защищенности ресурсов автоматизированных систем. В большинстве случаев основной причиной успешности компьютерной атаки является наличие уязвимостей программного обеспечения (ПО), используемого в составе таких систем [12]. Одним из направлений повышения уровня безопасности ПО является внедрение в рамках жизненного цикла ПО различных процедур, касающихся снижения числа ошибок и уязвимостей [5, 6, 7, 8, 13, 14].

Все это определяет актуальную задачу формирования требований к процессам создания ПО, выполнение которых позволит сократить число уязвимостей ПО и обеспечить их оперативное устранение в случае обнаружения.

Для дальнейшего изложения введем следующие определения. *Недостатком программы* будем называть любую ошибку, допущенную в ходе проектирования или реализации программы, которая, в случае оставления её неисправленной, может являться причиной уязвимости программы. *Под уязвимостью программы* будем понимать недостаток программы, который может быть использован для реализации угроз безопасности информации. Следует отметить, что уязвимость программы может быть результатом её реализации без учета требований по обеспечению безопасности информации или результатом наличия ошибок проектирования или кодирования. Надо понимать, что для реализации уязвимости программы необходимо наличие субъекта, воздействующего на программу (нарушителя) и способного эксплуатировать уязвимость. *Безопасным программным обеспечением* будем называть ПО, разработанное с использованием совокупности мер, направленных на предотвращение появления и устранение уязвимостей программы [9].

Постановка задачи исследования

К настоящему времени в области создания безопасного ПО известно определенное количество документов, оформленных в виде корпоративных, отраслевых и международных

1 Барабанов Александр Владимирович, кандидат технических наук, ЗАО «НПО «Эшелон», Москва, *ab@cnpo.ru*

2 Марков Алексей Сергеевич, доктор технических наук, старший научный сотрудник, МГТУ им. Н.Э.Баумана, *a.markov@bmstu.ru*

3 Цирлов Валентин Леонидович, кандидат технических наук, МГТУ им. Н.Э.Баумана, *v.tsirlov@bmstu.ru*

стандартов и содержащих указания для разработчиков ПО и «лучшие практики», которые рекомендуется внедрять в жизненном цикле ПО с целью создания ПО с минимально возможным количеством уязвимостей и формирования среды обеспечения оперативного устранения выявленных пользователем ПО проблем (уязвимостей ПО) [17, 18, 19, 20, 21, 23]. Предлагаемая документами номенклатура мер разработки безопасного ПО, как правило, является стандартной и содержит меры, связанные, например, с риск-анализом архитектуры ПО (моделирование угроз безопасности информации), проведением статического анализа исходного кода программы, тестирования на проникновение. Вместе с тем следует отметить, что существующие документы не содержат четко определенного аппарата, который мог бы использоваться для независимой оценки реализованных разработчиком мер требованиям разработки безопасного ПО. Известно, что методология «Общие критерии» в настоящее время широко используется при проведении оценки ПО по требованиям безопасности информации, но использование положений только этого документа для оценки мер разработки безопасного ПО не является достаточным, поскольку:

- «Общие критерии» применяются только для ПО с функциями безопасности (иначе говоря, средств защиты информации) - для ПО, в котором функции безопасности не реализованы, не может быть написано требуемое задание по безопасности и, соответственно, проведена оценка;

- предлагаемая «Общими критериями» номенклатура мер разработки ПО, выраженная в форме требований доверия, не обладает свойством полноты: в частности, отсутствуют меры, связанные с проведением статического и динамического анализа [1], обучением сотрудников и др.

Отдельно следует отметить, что требования к свойствам безопасного ПО, как правило, никогда не формулируются заказчиками/пользователями в терминах «Общих критериев» (например, с использованием понятия «оценочный уровень доверия»). Направленность «Общих критериев» иная – с помощью документа подтверждают, что ПО функционирует в соответствии со спецификацией, и определяют степень доверия к правильному функционированию ПО.

Таким образом, цель проведенного исследования состояла в создании аппарата обоснованного формирования множества мер разработки безопасного ПО, подлежащих реализации разработчиком ПО и оцениванию с привлечением независимых организаций.

В соответствии с целью исследования были поставлены и решены следующие частные задачи исследования:

- анализ существующих механизмов, направленных на уменьшение количества уязвимостей в разрабатываемом ПО, и их применимости при проведении оценки соответствия ПО;

- формирование базового набора требований к разработке безопасного ПО, позволяющего проводить оценку соответствия процессов данным требованиям;

- разработка методики обоснованного формирования множества мер разработки безопасного ПО.

В качестве ограничений области исследования была определена необходимость гармонизации получаемых решений с современной нормативной базой оценки соответствия («Общие критерии») [4, 15] и организации жизненного цикла ПО [16].

Результаты исследования

Формирование базового набора требований к разработке безопасного ПО было выполнено на основе проведенного анализа существующих мер, направленных на уменьшение количества уязвимостей в разрабатываемом ПО. В рамках проведенного анализа были систематизированы и обобщены сведения о различных мерах разработки безопасного ПО, используемых отечественными и зарубежными организациями [2, 3]. С целью обеспечения проведения оценки соответствия процессов разработки ПО требованиям при формировании базового набора были учтены аспекты, связанные с формированием требований к:

- к документальным свидетельствам (выполнения требования);

- действиям оценщика (например, эксперта испытательной лаборатории), выполняемым в ходе оценки соответствия (сертификации).

В ходе выполнения исследования с целью учета выявленных особенностей было предложено использование при описании требования по разработке безопасного ПО набора параметров (рис. 1).

Краткое описание разработанного базового набора требований представлено в таблице 1. Достоверность и обоснованность представленного набора требований была подтверждена в рамках экспертизы, проведенной Техническим комитетом по стандартизации ТК-362 «Защита информации», насчитывающего более 100 организаций.

Требование по разработке безопасного ПО
+ уникальный идентификатор требования + название + ссылка на процесс жизненного цикла ПО + достигаемая цель + элементы действий разработчика + элементы содержания и представления документированных свидетельств + элементы действий оценщика

Рис. 1. *Описание требования по разработке безопасных программ*

Таблица 1

Предлагаемый базовый набор требований по разработке безопасного ПО

Идентификатор	Краткое содержание требования	Процесс жизненного цикла ПО
АТ-1	Разработчик ПО должен определить требования в части разработки безопасного ПО, предъявляемые к разрабатываемому ПО	Процесс анализа требований к ПО
ПА-1	Разработчик ПО должен выполнить моделирование угроз безопасности информации, источником которых является ПО, с целью выявления потенциальных угроз безопасности информации и обоснования требований к ПО	Процесс проектирования архитектуры ПО
ПА-2	Архитектура ПО (логическая структура ПО, в которой идентифицированы компоненты, их интерфейсы и концепция взаимодействия между ними) должна уточняться с учетом необходимости нейтрализации потенциальных угроз безопасности информации, которые были выявлены на этапе моделирования угроз безопасности информации, и выполнения требований в части разработки безопасного ПО, установленных в процессе анализа требований к ПО	
КК-1	Разработчик ПО должен идентифицировать каждое инструментальное средство, используемое при разработке ПО	Процессы конструирования и комплексирования ПО
КК-2	Разработчик ПО должен создавать программу на основе архитектуры ПО, определенной в ходе выполнения процессов проектирования и детального проектирования архитектуры ПО	
КК-3	Разработчик ПО должен создать или выбрать стандарт оформления исходного кода программы, содержащий перечень правил и рекомендаций, направленных на минимизацию количества потенциально уязвимых конструкций в исходном коде программы	
КК-4	Разработчик ПО должен проводить статический анализ исходного кода программы с целью выявления потенциально уязвимых конструкций в исходном коде программы	
КК-5	Разработчик ПО должен проводить периодическую экспертизу исходного кода программы с целью выявления потенциально уязвимых конструкций в исходном коде программы	
КТ-1	Разработчик ПО должен проводить функциональное тестирование ПО для того, чтобы определить, выполняются ли требования, идентифицированные в процессе анализа требований к ПО	Процесс квалификационного тестирования ПО
КТ-2	Разработчик ПО должен проводить тестирование на проникновение в отношении программы с целью выявления уязвимостей ПО	
КТ-3	Разработчик ПО должен проводить динамический анализ кода программы с целью выявления уязвимостей ПО	
КТ-4	Разработчик ПО должен обеспечить проведение фаззинг-тестирования программы с целью выявления уязвимостей программы. Тесты, выполняемые в рамках фаззинг-тестирования программы, должны быть разработаны с учетом:	

ИП-1	Разработчик ПО должен применять технические и организационные меры, необходимые для обнаружения модификации ПО или любого расхождения между оригиналом и версией, полученной пользователем	Процессы инсталляции ПО и поддержки приемки ПО
ИП-2	В состав поставляемого ПО должна быть включена эксплуатационная документация в объеме, достаточном для правильной настройки и безопасного применения ПО	
РП-1	Разработчик ПО должен реализовать процедуру, позволяющую выполнять отслеживание и исправление обнаруженных уязвимостей ПО	Процесс решения проблем в ПО
РП-2	Разработчик ПО должен быть способен предложить пользователю решение проблемы в ситуации, когда неизвестная ранее уязвимость ПО используется для проведения компьютерной или сетевой атаки на информационную систему пользователя	
РП-3	В экстренных ситуациях разработчик ПО должен быть способен к выпуску обновлений ПО в обход стандартной процедуры выпуска новых версий ПО. Если экстренный выпуск обновлений ПО невозможен, разработчик ПО должен быть способен предложить альтернативные способы временного решения проблемы, включая использование пользователем дополнительных средств защиты.	
РП-4	Разработчик ПО должен проводить систематический поиск уязвимостей программы.	
РП-5	При выполнении модернизации ПО (выпуска обновления ПО) в отношении измененного ПО должны выполняться меры по разработке безопасного ПО.	
МДК-1	Разработчиком ПО должна быть реализована процедура, позволяющая выполнять уникальную маркировку каждой версии ПО	Процесс менеджмента документации и конфигурации ПО
МДК-2	Разработчик ПО должен использовать систему управления конфигурацией ПО, позволяющую уникально идентифицировать элементы конфигурации, имеющие отношение к разрабатываемому ПО	
МДК-3	Система управления конфигурацией ПО должна идентифицировать элементы конфигурации, которые связаны с реализацией функций безопасности ПО	
МДК-4	Система управления конфигурацией ПО должна предоставлять средства для определения всех элементов конфигурации, на которые воздействует модифи кация данного элемента конфигурации	
МИ-1	Разработчиком ПО должны применяться технические и организационные меры, обеспечивающие защиту от несанкционированного доступа к элементам конфигурации	Процесс менеджмента инфраструктуры
МИ-2	Разработчиком ПО должны применяться технические и организационные меры, обеспечивающие резервное копирование и восстановление элементов конфигурации	
МИ-3	Разработчиком ПО должны применяться технические и организационные меры, обеспечивающие регистрацию всех событий, связанных с фактами изменения элементов конфигурации, в журнале регистрации событий	
МЛР-1	Разработчик ПО должен проводить периодическое обучение сотрудников с целью повышения их осведомленности в области разработки безопасного ПО	Процесс менеджмента людских ресурсов
МЛР-2	Разработчик ПО должен проводить периодический анализ программы обучения сотрудников для установления её пригодности, адекватности и результативности для достижения установленных целей в области разработки безопасного ПО	

В ходе проведения исследования была разработана методика обоснованного формирования множества мер разработки безопасного ПО (рис. 2). Поскольку современное развитие отечественной системы сертификации средств защиты информации по требованиям безопасности информации связано с апробацией методологии «Общих критериев» [22], то в основу методики положен подход, используемый данными документами при формировании требований безопасности к объектам оценки. Следует отметить, что при разработке методики предполагалось, что все требования не являются функциональными, иначе говоря, требования являются аналогами требований доверия «Общих критериев».

Практические аспекты использования базового набора требований

Созданный базовый набор требований разработки безопасного ПО использовался при создании проекта национального стандарта ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. Общие требования», прошедшего экспертизу в рамках работы Технического комитета по стандартизации ТК-362 «Защита информации». На рис. 3 показана предполагаемая взаимосвязь разработанного проекта с другими национальными стандартами. Следует отметить, что проект стандарта может применяться в качестве источника для формирования мер и средств контроля и управления безопасностью ПО в соответствии с ГОСТ Р ИСО/МЭК 27034-1. Кроме этого, стандарт может применяться для конкретизации или расширения компонентов доверия из ГОСТ Р ИСО/МЭК 15408-3.

Если разработчиком ПО планируется проведение оценки ПО в соответствии с требованиями «Общих критериев» (ГОСТ Р ИСО/МЭК 15408), то подготовка ПО к оценке может осуществляться в рамках существующих процессов, в которых реализованы меры по разработке безопасного ПО, путем выполнения дополнительных мер. Таблица 2 иллюстрирует взаимосвязь между разработанным базовым набором требований и требований доверия к безопасности «Общих критериев» и может использоваться при подготовке ПО к оценке.

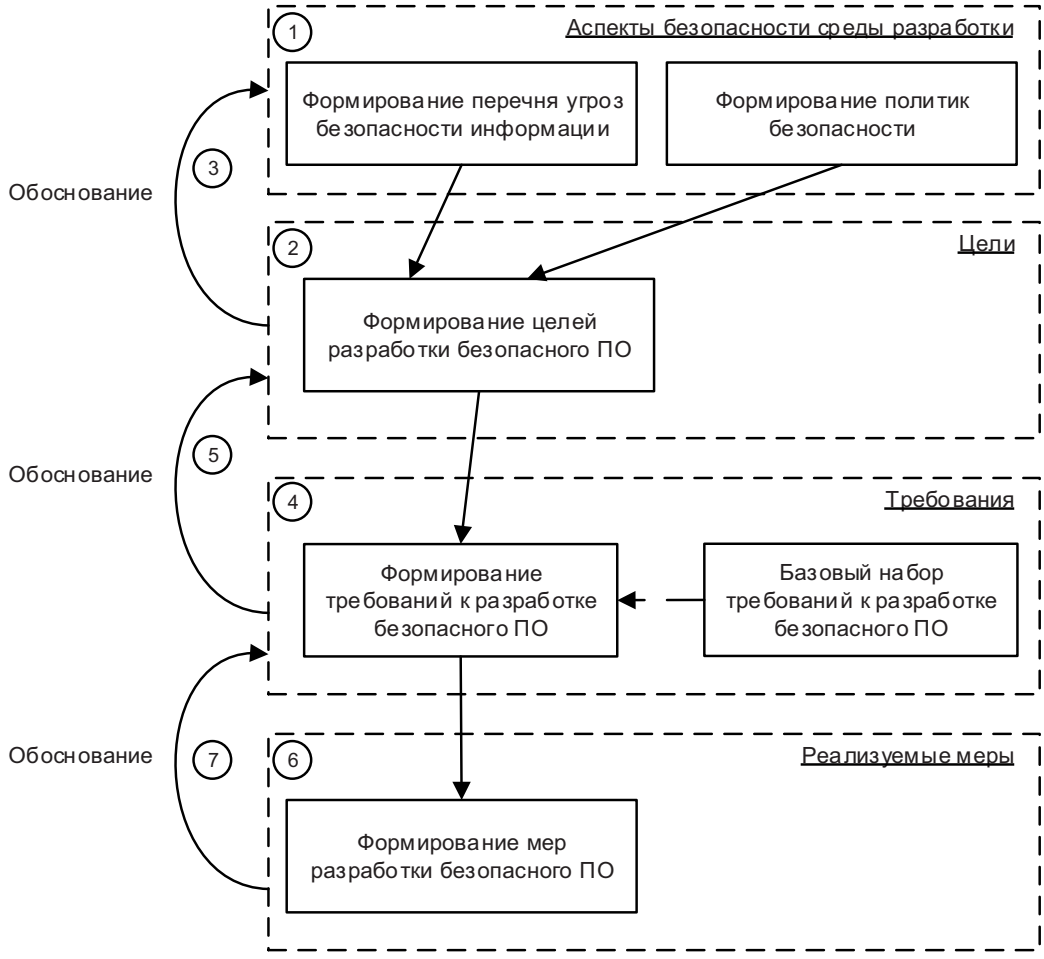

Рис. 2. *Методика обоснованного формирования множества мер разработки*

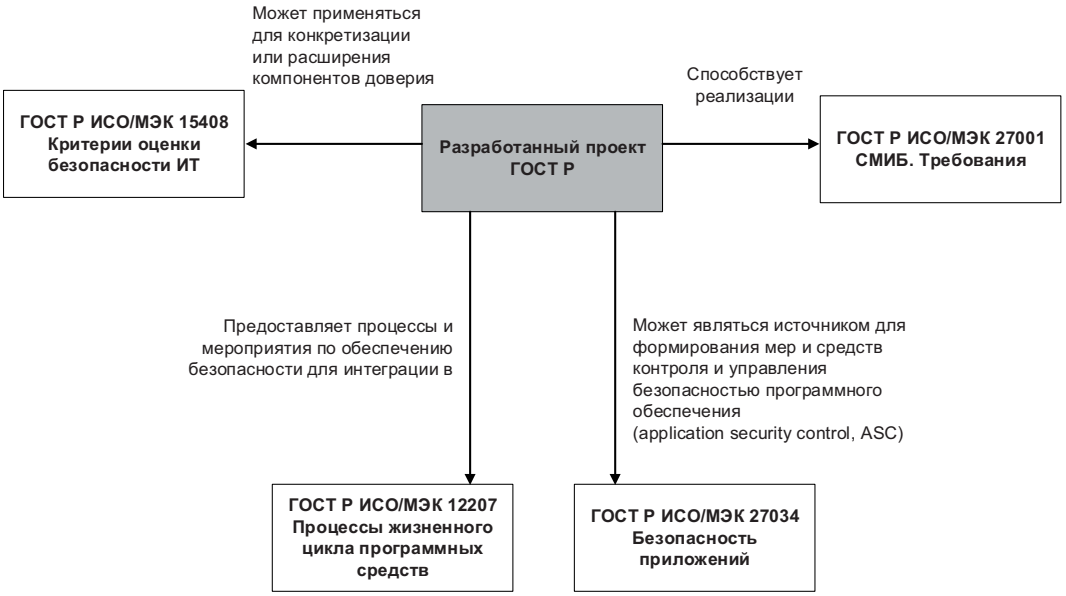

Рис. 3. *Гармонизация разработанного проекта национального стандарта*

Таблица 2

Взаимосвязь между разработанным базовым набором требований и требований доверия к безопасности «Общих критериев»

Меры по разработке безопасного ПО	**Семейство (класс) требований доверия к безопасности по ГОСТ Р ИСО/МЭК 15408-3**
АТ-1	ASE «Оценка задания по безопасности»
ПА-1	ADV_ARC «Архитектура безопасности»
ПА-2	ADV_FSP «Функциональная спецификация», ADV_TDS «Проект ОО», ADV_ARC «Архитектура безопасности»
КК-1	ALC_TAT «Инструментальные средства и методы»
КК-2	ADV_IMP «Представление реализации»
КК-3	ALC_TAT «Инструментальные средства и методы» (в части ALC_TAT.2)
КК-4	ALC_TAT «Инструментальные средства и методы» ALC_CMC «Возможности управления конфигурацией» (в части ALC_CMC.5)
КК-5	ALC_TAT «Инструментальные средства и методы» (в части ALC_TAT.2), ALC_CMC «Возможности управления конфигурацией» (в части ALC_CMC.5), ALC_CMS «Область управления конфигурацией» (в части ALC_CMS.4)
КТ-1	ATE_COV «Покрытие», ATE_DPT «Глубина», ATE_FUN «Функциональное тестирование»
КТ-2	AVA_VAN «Анализ уязвимостей»
КТ-3	отсутствует
КТ-4	отсутствует
ИП-1	ALC_DEL «Поставка»
ИП-2	AGD_OPE «Руководство пользователя по эксплуатации», AGD_PRE «Подготовительные процедуры»
РП-1	ALC_FLR «Устранение недостатков»
РП-2	ALC_FLR «Устранение недостатков» (в части ALC_FLR.2)

Меры по разработке безопасного ПО	Семейство (класс) требований доверия к безопасности по ГОСТ Р ИСО/МЭК 15408-3
РП-3	ALC_FLR «Устранение недостатков» (в части ALC_FLR.2)
РП-4	отсутствует
РП-5	ALC_TAT «Инструментальные средства и методы», ALC_CMC «Возможности управления конфигурацией, ALC_CMS «Область управления конфигурацией»
МДК-1	ALC_CMC «Возможности управления конфигурацией», ALC_CMS «Область управления конфигурацией»
МДК-2	ALC_CMC «Возможности управления конфигурацией», ALC_CMS «Область управления конфигурацией»
МДК-3	ALC_CMC «Возможности управления конфигурацией»
МДК-4	ALC_CMC «Возможности управления конфигурацией»
МИ-1	ALC_CMC «Возможности управления конфигурацией»
МИ-2	ALC_DVS «Безопасность разработки»
МИ-3	ALC_CMC «Возможности управления конфигурацией»
МЛР-1	отсутствует
МЛР-2	отсутствует

Выводы

К основным результатам исследования можно отнести следующие:

1. Проанализированы механизмы разработки безопасного ПО, используемые в настоящее время ведущими отечественными и зарубежными разработчиками. На основе систематизации и анализа публикаций и документов в области создания безопасного ПО разработан базовый набор из 28-и требований по разработке безопасного ПО, который обладает следующими особенностями:

- позволяет установить связь с процессами жизненного цикла ПО, регламентированные ИСО 12207, в результате чего обеспечивается уменьшение времени интеграции необходимых мер в уже существующие в организации системы менеджмента;

- позволяет установить связь с требованиями доверия «Общих критериев», что обеспечивает сокращение затрат на подготовку ПО, разработанного с использованием мер разработки безопасного ПО, к процедуре оценки по «Общим критериям»;

- определяет документированные свидетельства выполнения требований и действий оценщика, что обеспечивает детерминированность процесса независимой оценки.

2. Предложена методика, которая может использоваться разработчиками ПО для обоснованного формирования множества мер разработки безопасного ПО, подлежащих реализации в инфраструктуре разработки. Отличительной особенностью разработанной методики является ее согласованность с подходом формирования и обоснования функций безопасности и мер разработки «Общих критериев». В рамках исследования также была предложена концептуальная модель выбора мер безопасной разработки [3].

Исследование показало, что выполнение даже отдельных установленных требований приводит к существенному уменьшению количества уязвимостей в разрабатываемом ПО [10]. Созданный базовый набор использовался при разработке проекта национального стандарта ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. Общие требования», прошедшего экспертизу в рамках работы Технического комитета по стандартизации ТК-362 «Защита информации» (включающего экспертов от 105 организаций).

Определение организационно-технических мер разработки безопасного ПО позволяет также регламентировать вопросы проверки серийного производства при проведении сертификации [11].

Рецензент: доктор технических наук, профессор Петренко Сергей Анатольевич.

Литература:

1. Аветисян А.И., Белеванцев А.А., Чукляев И.И. Технологии статического и динамического анализа уязвимостей программного обеспечения // Вопросы кибербезопасности. 2014. № 3 (4). С. 20-28.
2. Барабанов А.В. Стандартизация процесса разработки безопасных программных средств // Вопросы кибербезопасности. 2013. № 1 (1). С. 37-41.
3. Барабанов А.В., Марков А.С., Цирлов В.Л. Методический аппарат анализа и синтеза комплекса мер разработки безопасного программного обеспечения // Программные продукты и системы. 2015. № 4 (112). С. 177-186.
4. Барабанов А.В., Марков А.С., Цирлов В.Л. Оценка соответствия средств защиты информации «Общим критериям» // Информационные технологии. 2015. Т. 21. № 4. С. 264-270.
5. Еремеев М.А., Беляков И.А. Интеллектуальные технологии при сертификации программ // Автоматика, связь, информатика. 2012. № 2. С. 23-25.
6. Жидков И.В., Шубенин А.А., Хабибуллин И.В., Поздняков С.Ю. Испытания систем защиты информации автоматизированных систем управления // Решетневские чтения. 2013. Т. 2. № 17. С. 281-282.
7. Зефиров С.Л., Колобанов А.Ю. Процесс выбора оптимальных методов разработки безопасного программного обеспечения // Труды международного симпозиума Надежность и качество. 2011. Т. 2. С. 409-413.
8. Казарин О.В., Кондаков С.Е., Троицкий И.И. Подходы к количественной оценке защищенности ресурсов автоматизированных систем // Вопросы кибербезопасности. 2015. № 2 (10). С. 31-35.
9. Марков А.С., Фадин А.А. Систематика уязвимостей и дефектов безопасности программных ресурсов // Защита информации. Инсайд. 2013. №3. С. 56-61.
10. Марков А.С., Цирлов В.Л. Опыт выявления уязвимостей в зарубежных программных продуктах // Вопросы кибербезопасности. 2013. № 1(1). С.42-48.
11. Марков А.С., Цирлов В.Л. Сертификация программ: мифы и реальность // Открытые системы. СУБД. 2011. № 6. С. 26-29.
12. Михайлов Д.М., Жуков И.Ю., Шеремет И.А. Защита автоматизированных систем от информационно-технологических воздействий. М.: НИЯУ МИФИ, 2014. 184 с.
13. Михаленок В.В. Создание безопасного программного обеспечения на базе платформы.net // Электронное периодическое издание Информационная среда образования и науки. 2011. № 1. С. 51-58.
14. Рибер Г., Малмквист К., Щербаков А. Многоуровневый подход к оценке безопасности программных средств // Вопросы кибербезопасности. 2014. № 1 (2). С. 36-39.
15. Barabanov A., Markov A. Modern Trends in The Regulatory Framework of the Information Security Compliance Assessment in Russia Based on Common Criteria. In Proceedings of the 8th International Conference on Security of Information and Networks (Sochi, Russian Federation, September 08-10, 2015). SIN '15. ACM New York, NY, USA, 2015, pp. 30-33. DOI = http://dx.doi.org/10.1145/2799979.2799980.
16. Barabanov A., Markov A., Fadin A., Tsirlov V., Shakhalov I. Synthesis of Secure Software Development Controls. In Proceedings of the 8th International Conference on Security of Information and Networks (Sochi, Russian Federation, September 08-10, 2015). SIN '15. ACM New York, NY, USA, 2015, pp. 93-97. DOI = http://dx.doi.org/10.1145/2799979.2799998.
17. Biró M., Molnár B. Synergies Between the Common Criteria and Process Improvement. LNCS, 2007, No 4764, pp. 31-45. DOI = http://dx.doi.org/10.1007/978-3-540-75381-0_4.
18. De Win B., Scandariato R., Buyens K., Grégoire J., Joosen W. On the Secure Software Development Process: CLASP, SDL and Touchpoints Compared. Information and Software Technology, 2009, Vol. 51, No 7 (Jul. 2009), pp. 1152-1171. DOI= http://dx.doi.org/10.1016/j.infsof.2008.01.010.
19. Gary McGraw. 2015. Software security and the building security in maturity model (BSIMM). J. Comput. Sci. Coll. 30, 3 (January 2015), 7-8.
20. Howard M., Lipner S. The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software. Microsoft Press, 2006. 352 p.
21. Kara M. Review on Common Criteria as a Secure Software Development Model. IJCSIT, 2012, V. 4, No 2 (Apr. 2012), pp. 83-94. DOI = http://dx.doi.org/10.5121/ijcsit.2012.4207.
22. Markov A., Luchin D., Rautkin Y., Tsirlov V. Evolution of a Radio Telecommunication Hardware-Software Certification Paradigm in Accordance with Information Security Requirements, In Proceedings of the 11th International Siberian Conference on Control and Communications (Omsk, Russia, May 21-23, 2015). SIBCON-2015. IEEE, 2015, pp. 1-4. DOI = http://dx.doi.org/10.1109/SIBCON.2015.7147139.
23. Viega J., McGraw G. Building Secure Software: How to Avoid Security Problems the Right Way. Addison-Wesley Professional, 2001. 528 p.

THE 28 MAGIC PRACTICES FOR SECURE SOFTWARE DEVELOPMENT

A.V.Barabanov[1], A.S.Markov[2], V.L.Tsirlov[3]

1 Aleksandr Baranov, Ph.D., NPO Echelon, Moscow, ab@cnpo.ru

2 Aleksey Markov, Dr.Sc., Associate Professor, Bauman MSTU, Moscow, a.markov@bmstu.ru

3 Valentin Tsirlov, Ph.D., Bauman MSTU, Moscow, v.tsirlov@bmstu.ru

The article deals with creation of method for secure software development practice selection. The conceptual definition of secure software is proposed. It is formulated a set of secure software requirements to integrated into software development lifecycle (the basic set of secure software requirements). The basic set of secure software requirements were developed based on existing research papers review. The original methods for selection of security controls for developing secure software based on a set of requirements generated is developed. Practical aspects related to using a proposed basic set of secure software requirements during development of Russian standard (draft) on secure software development is shown. The expediency of the harmonization of regulatory requirements developed and practical measures with the methodologies of international standards like ISO 15408 and ISO 12207 series is shown. Information related to harmonization between a standard on secure software development and other Russian standard is given.

***Keywords**: secure software, vulnerability, Common Criteria, secure software development lifecycle, ISO 15408, ISO 12207*

References:

1. Avetisyan A.I., Belevantsev A.A., Chuklyaev I.I. Tekhnologii staticheskogo i dinamicheskogo analiza uyazvimostey programmnogo obespecheniya, Voprosy kiberbezopasnosti, 2014, No 3 (4), pp. 20-28.
2. Barabanov A.V. Standartizatsiya protsessa razrabotki bezopasnykh programmnykh sredstv, Voprosy kiberbezopasnosti [Cybersecurity issues], 2013, No 1 (1), pp. 37-41.
3. Barabanov A.V., Markov A.S., Tsirlov V.L. Metodicheskiy apparat analiza i sinteza kompleksa mer razrabotki bezopasnogo programmnogo obespecheniya, Programmnye produkty i sistemy, 2015, No 4 (112), pp. 177-186.
4. Barabanov A.V., Markov A.S., Tsirlov V.L. Otsenka sootvetstviya sredstv zashchity informatsii "Obshchim kriteriyam", Informatsionnye tekhnologii, 2015. T. 21, No 4, pp. 264-270.
5. Eremeev M.A., Belyakov I.A. Intellektual'nye tekhnologii pri sertifikatsii programm, Avtomatika, svyaz', informatika, 2012, No 2, pp. 23-25.
6. Zhidkov I.V., Shubenin A.A., Khabibullin I.V., Pozdnyakov S.Yu. Ispytaniya sistem zashchity informatsii avtomatizirovannykh sistem upravleniya, Reshetnevskie chteniya, 2013. T. 2, No 17, pp. 281-282.
7. Zefirov S.L., Kolobanov A.Yu. Protsess vybora optimal'nykh metodov razrabotki bezopasnogo programmnogo obespecheniya, Trudy mezhdunarodnogo simpoziuma Nadezhnost' i kachestvo, 2011. V. 2, pp. 409-413.
8. Kazarin O.V., Kondakov S.E., Troitskiy I.I. Podkhody k kolichestvennoy otsenke zashchishchennosti resursov avtomatizirovannykh sistem, Voprosy kiberbezopasnosti [Cybersecurity issues], 2015, No 2 (10), pp. 31-35.
9. Markov A.S., Fadin A.A. Sistematika uyazvimostey i defektov bezopasnosti programmnykh resursov, Zashchita informatsii. Insayd, 2013, No3, pp. 56-61.
10. Markov A.S., Tsirlov V.L. Opyt vyyavleniya uyazvimostey v zarubezhnykh programmnykh produktakh, Voprosy kiberbezopasnosti, 2013, No 1(1), pp. 42-48.
11. Markov A.S., Tsirlov V.L. Sertifikatsiya programm: mify i real'nost', Otkrytye sistemy. SUBD, 2011, No 6, pp. 26-29.
12. Mikhaylov D.M., Zhukov I.Yu., Sheremet I.A. Zashchita avtomatizirovannykh sistem ot informatsionno-tekhnologicheskikh vozdeystviy. M.: NIYaU MIFI, 2014. 184 p.
13. Mikhalenok V.V. Sozdanie bezopasnogo programmnogo obespecheniya na baze platformy.net, Elektronnoe periodicheskoe izdanie Informatsionnaya sreda obrazovaniya i nauki, 2011, No 1, pp. 51-58.
14. Riber G., Malmkvist K., Shcherbakov A. Mnogourovnevyy podkhod k otsenke bezopasnosti programmnykh sredstv, Voprosy kiberbezopasnosti. 2014, No 1 (2), pp. 36-39.
15. Barabanov A., Markov A. Modern Trends in The Regulatory Framework of the Information Security Compliance Assessment in Russia Based on Common Criteria. In Proceedings of the 8th International Conference on Security of Information and Networks (Sochi, Russian Federation, September 08-10, 2015). SIN '15. ACM New York, NY, USA, 2015, pp. 30-33. DOI = http://dx.doi.org/10.1145/2799979.2799980.
16. Barabanov A., Markov A., Fadin A., Tsirlov V., Shakhalov I. Synthesis of Secure Software Development Controls. In Proceedings of the 8th International Conference on Security of Information and Networks (Sochi, Russian Federation, September 08-10, 2015). SIN '15. ACM New York, NY, USA, 2015, pp. 93-97. DOI = http://dx.doi.org/10.1145/2799979.2799998.
17. Biró M., Molnár B. Synergies Between the Common Criteria and Process Improvement. LNCS, 2007, No 4764, pp. 31-45. DOI = http://dx.doi.org/10.1007/978-3-540-75381-0_4.
18. De Win B., Scandariato R., Buyens K., Grégoire J., Joosen W. On the Secure Software Development Process: CLASP, SDL and Touchpoints Compared. Information and Software Technology, 2009, Vol. 51, No 7 (Jul. 2009), pp. 1152-1171. DOI= http://dx.doi.org/10.1016/j.infsof.2008.01.010.
19. Gary McGraw. 2015. Software security and the building security in maturity model (BSIMM). J. Comput. Sci. Coll. 30, 3 (January 2015), 7-8.
20. Howard M., Lipner S. The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software. Microsoft Press, 2006. 352 p.
21. Kara M. Review on Common Criteria as a Secure Software Development Model. IJCSIT, 2012, V. 4, No 2 (Apr. 2012), pp. 83-94. DOI = http://dx.doi.org/10.5121/ijcsit.2012.4207.
22. Markov A., Luchin D., Rautkin Y., Tsirlov V. Evolution of a Radio Telecommunication Hardware-Software Certification Paradigm in Accordance with Information Security Requirements, In Proceedings of the 11th International Siberian Conference on Control and Communications (Omsk, Russia, May 21-23, 2015). SIBCON-2015. IEEE, 2015, pp. 1-4. DOI = http://dx.doi.org/10.1109/SIBCON.2015.7147139.
23. Viega J., McGraw G. Building Secure Software: How to Avoid Security Problems the Right Way. Addison-Wesley Professional, 2001. 528 p.