УДК 004.056:006.06

ПРАКТИЧЕСКИЕ АСПЕКТЫ РЕАЛИЗАЦИИ МЕР ПО РАЗРАБОТКЕ БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**Гришин М.И., Марков А.С., Цирлов В.Л.**

АО «НПО «Эшелон», 107023, Россия, г. Москва, ул. Электрозаводская, 24, e-mail: mail@npo-echelon.ru

В работе рассмотрены вопросы стандартизации процедур и мер по разработке программного обеспечения, связанные с уменьшением количества уязвимостей и оперативным их исправлением в случае обнаружения. Указаны основные угрозы процессу разработки программ. Приведены данные по гармонизации стандартов по линии ГОСТ Р 56939. Обоснован концептуальный подход по реализации мер по разработке безопасного программного обеспечения. Представлены основные требования и детальные рекомендации. Приведен краткий пример. Отмечены направления развития стандартизации в области безопасности программ

Ключевые слова: безопасность программ, процедуры разработки программ, процессы жизненного цикла программ, меры безопасности информации, угрозы, уязвимости, зрелость компании

PRACTICAL ASPECTS OF THE IMPLEMENTATION OF MEASURES TO DEVELOP SAFE SOFTWARE

**Grishin M.I., Markov A.S., Tsirlov V.L.**

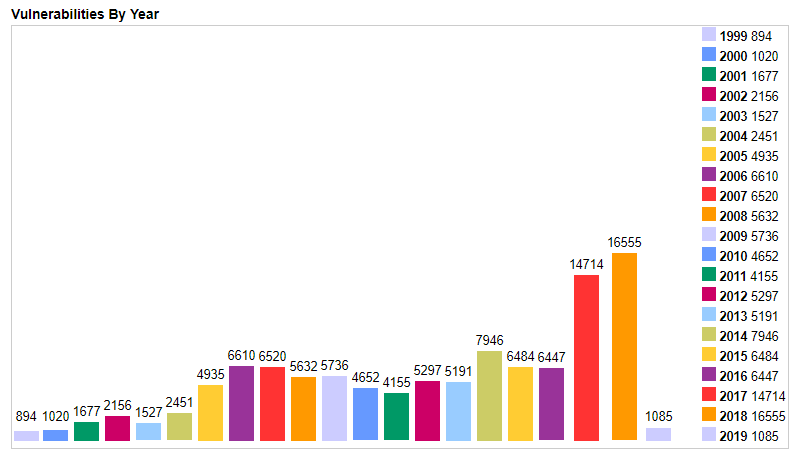
JSC NPO Echelon, 107023, Russia, Moscow, st. Electrozavodskaya, 24, e-mail: mail@npo-echelon.ru

The paper discusses the issues of standardization of procedures and measures for software development related to reducing the number of vulnerabilities and their prompt correction in case of detection. The main threats to the program development process are indicated. The data on harmonization of standards in accordance with GOST R 56939 are presented. A conceptual approach to the implementation of measures for the development of secure software is substantiated. Basic requirements and detailed recommendations are presented. A short example is given. The directions of development of standardization in the field of program security are marked

Keywords: program security, program development procedures, program life cycle processes, information security measures, threats, vulnerabilities, company maturity

**Введение**

Одним из базовых направлений противодействия росту количества уязвимостей (рис. 1) в программных ресурсах информационных систем является создание систем менеджмента безопасной разработки программной продукции [1, 2, 15].



*Источник: CVEDetails*

Рис. 1. Рост числа опубликованных уязвимостей

Востребованность развития стандартов в указанной области обусловлена следующим:

1. Испытательные (сертификационные) лаборатории (ИЛ) отмечают высокий уровень ошибок и уязвимостей в программных продуктах, разработанных организациями, в которых отсутствуют системы менеджмента качества и т.п.;

2. Современные международные стандарты и руководства («хорошие практики») не обеспечивают полноту регламентации всех процессов и процедур разработки программ с учетом современного спектра угроз информационной безопасности (ИБ).

На рис. 2 проиллюстрирован эмпирический закон обратной пропорциональности степени уязвимости программ от «зрелости» компании (по COBIT), который выведен на основе статистики, полученной в ряде ИЛ [1].

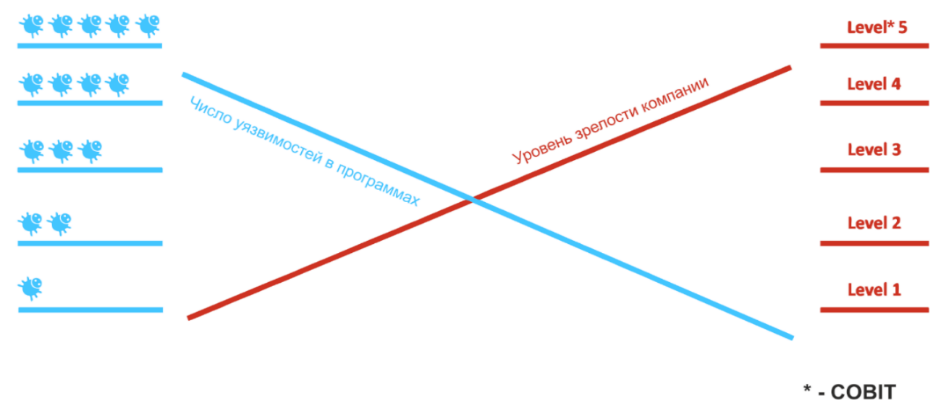


Рис. 2. Закон 5х5 – обратной пропорциональности зрелости компании и количества уязвимостей в программной продукции

В табл. 1 и 2 показаны фрагменты анализа международных нормативных документов, что демонстрирует отсутствие полноты зарубежных изысканий [14, 15].

Таблица 1. Каталоги угроз, связанные с жизненным циклом программ

|  |  |  |
| --- | --- | --- |
| Каталог угроз, связанных с жизненным циклом программам | Учет угроз, специфичных среде разработки | Учет непреднамеренных угроз |
| MITRE&DSE [16] | + | - |
| NIST SP 800-30 rev.1 | - | - |
| NIST IR 8144 | + | - |
| MITRE CAPEC | + | - |
| БДУ ФСТЭК России | - | + |

Таблица 2. Нормативные документы по разработке безопасных программ

| Характеристика, наличие мер по разработке безопасного программного обеспечения | Стандарт/руководство | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| ISO/IEC 15408 | Microsoft SDL | Open SAMM | OWASP CLASP | ISO/IEC TR 24772 | ISO/IEC 27034-1 |
| Обучение сотрудников | - | + | + | - | - | - |
| Обеспечение безопасности инфраструктуры | + | - | - | - | - | - |
| Управление конфигурацией разрабатываемых программ | + | - | - | - | - | - |
| Моделирование угроз безопасности информации, источником которых являются программы | + | + | + | + | - | - |
| Определение требований в части разработки безопасных программ | + | + | + | + | - | - |
| Использование стандарта оформления исходного кода | - | + | + | + | + | - |
| Проведение статического анализа исходного кода | - | + | + | + | - | - |
| Проведение динамического анализа кода | - | + | + | + | - | - |
| Проведение экспертизы исходного кода программ в ручном режиме | - | + | + | + | - | - |
| Проведение анализа уязвимостей | -[[1]](#footnote-1) | + | + | + | - | - |
| Обеспечение безопасности поставки | + | + | + | - | - | - |
| Устранение выявляемых при эксплуатации уязвимостей | + | + | + | + | - | - |
| Возможность использования документов при сертификации | + | - | - | - | - | - |
| Наличие методики выбора подмножества мер разработки | + | + | + | - | - | + |
| Согласованность с процессами жизненного цикла программ (согласно ISO/IEC 12207) | - | - | - | - | - | + |

В настоящее время в России разработан и вступил в действие национальный стандарт по разработке безопасных программ - ГОСТ Р 56939-2016, а также инициирована разработка линейки связанных с ним стандартов. Гармонизация указанного стандарта с международными стандартами проиллюстрирована на рис. 3.

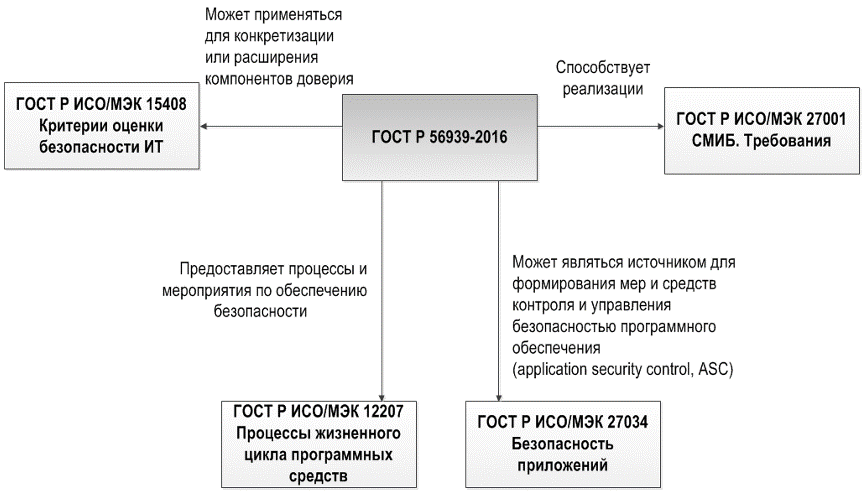


Рис. 3. Гармонизация стандартов

Несмотря на то, что указанный национальный стандарт вступил в силу полтора года назад и на него уже представлены ссылки во многих новых и перспективных НПА ФСТЭК России, вопросы же практической реализации данного стандарта находятся на начальной стадии. Обсуждение этого составляет основной предмет исследования данной публикации.

**Процессы и меры разработки безопасных программ**

Целью разработки линейки стандартов по разработке безопасных программ является предотвращение появления и устранение уязвимостей программ. При этом прослеживается связь между **процессами** разработки (что регламентировано ГОСТ Р ИСО/МЭК 12207) и **мерами** по безопасной разработке, которые выбираются с учетом риск-ориентированного подхода, т.е. с учетом актуальных **угроз**. Основные группы мер проиллюстрированы на рис. 4 и 5.

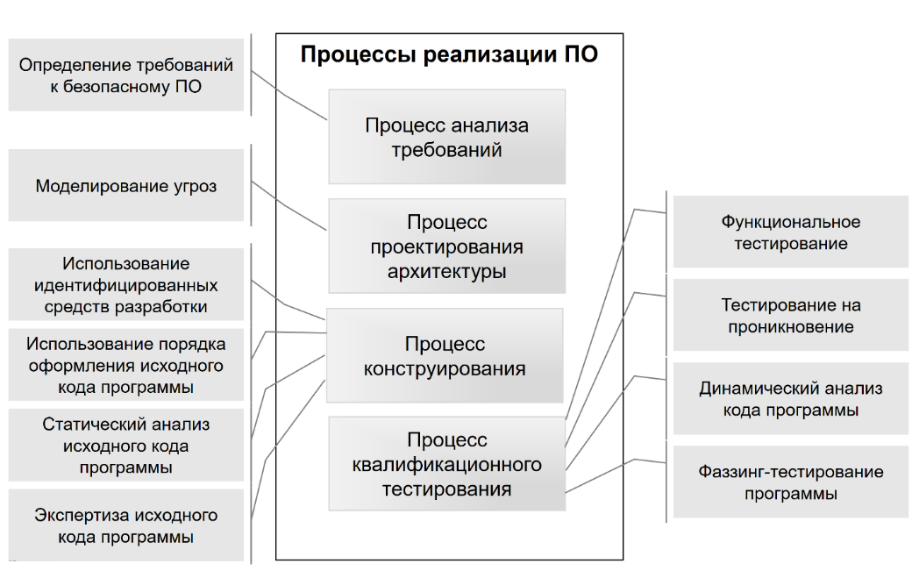


Рис. 4 Меры по реализации безопасных программ

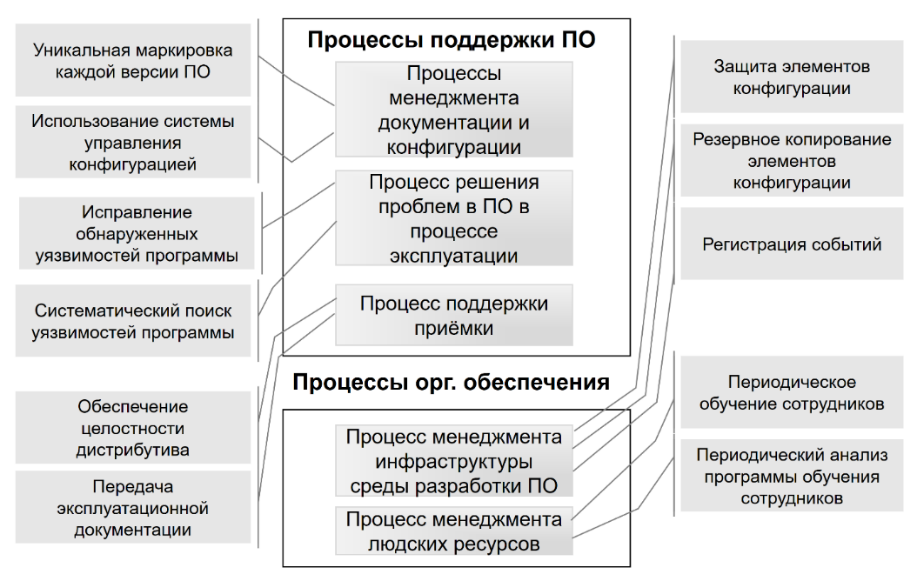


Рис. 5 Меры по поддержке и обеспечению безопасных программ

**Рекомендации по практической реализации мер**

Концептуальная модель разработки практических рекомендаций представлена на рис. 6.

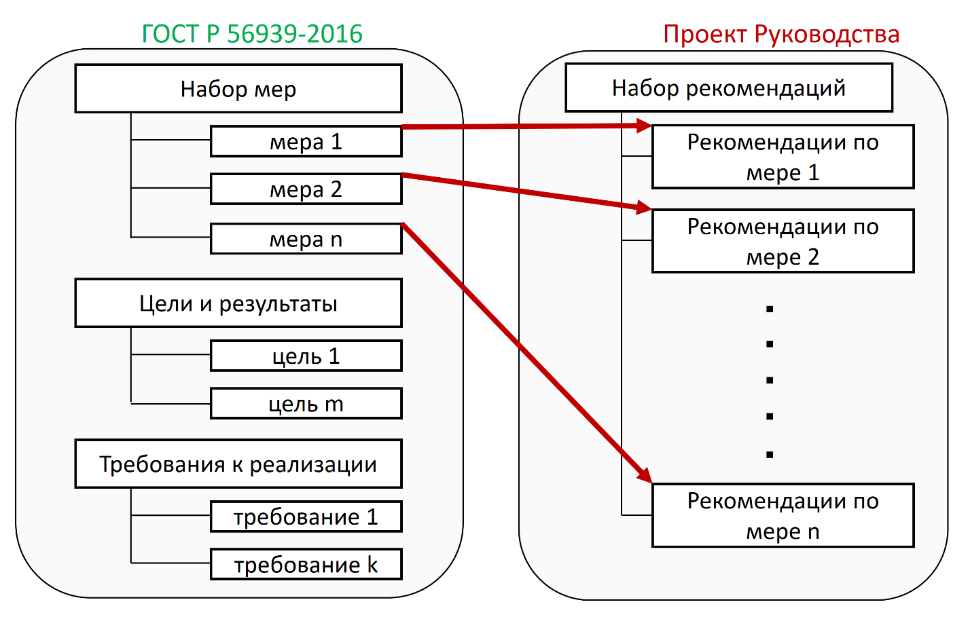


Рис. 6. Соответствие рекомендаций и мер

Общий методический подход к представлению практических рекомендаций может включать в себя:

- основные требования к процессу разработки рекомендаций;

- собственно рекомендации по каждой мере.

Требования могут вытекать из ряда «хороших практик» [15] и включать в себя:

- получение предварительного одобрения руководства;

- определение области действия мер;

- оценку существующих процессов с точки зрения выполнимости требований к мерам;

- создание и реализацию плана внедрения мер;

- реализацию внутренней проверки мер.

При составлении рекомендации по конкретной мере следует представить: краткое описание меры, описание рекомендации, перечень документации, а также распределение ролей и обязанностей. На наш взгляд, полезен будет иллюстративный материал в виде полуформальных моделей (рис. 7), где можно отметить вход-выход и условия.

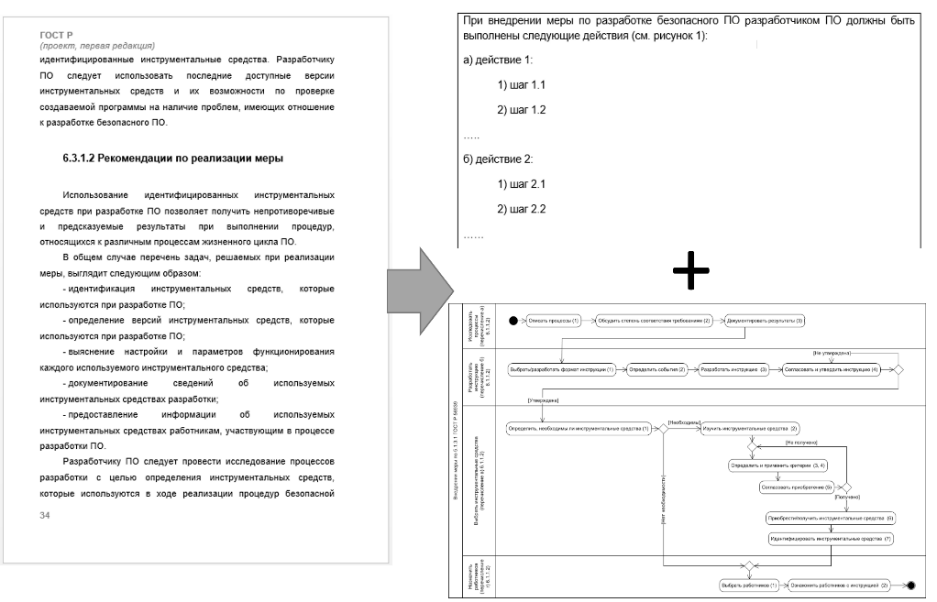


Рис. 7. Пример использования полуформальных моделей

По причине того, что стандарт определяет основных 25 мер, в данной публикации мы ограничимся краткими комментариями по одной из них - «Статический анализ исходного кода» (см. рис. 4).

**Статический анализ исходного кода**

Напомним, что под статическим анализом исходного кода программы понимают вид работ по инструментальному исследованию программы, основанный на анализе исходного кода программы с использованием специализированных инструментальных средств (статических анализаторов) в режиме, не предусматривающем реального выполнения кода [1, 3-13]. На рис. 8 представлен вариант методики практической реализации указанной меры, которая включает в себя ряд ниже представленных шагов.

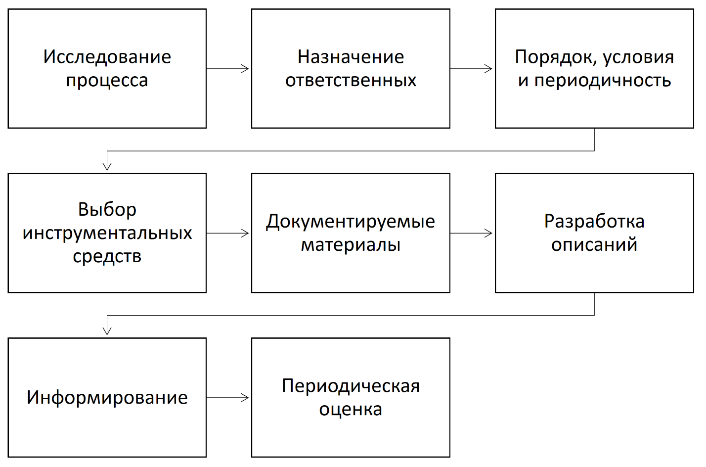


Рис. 8. Общая последовательность реализации меры

1. В ходе проведения первичного исследования идентифицируются различные особенности соответствующего процесса разработки, его части, их последовательность, данные, ресурсы и пр. Результаты исследования могут быть представлены в виде простых диаграмм, схем или кратких описаний (см., например, рис.9).

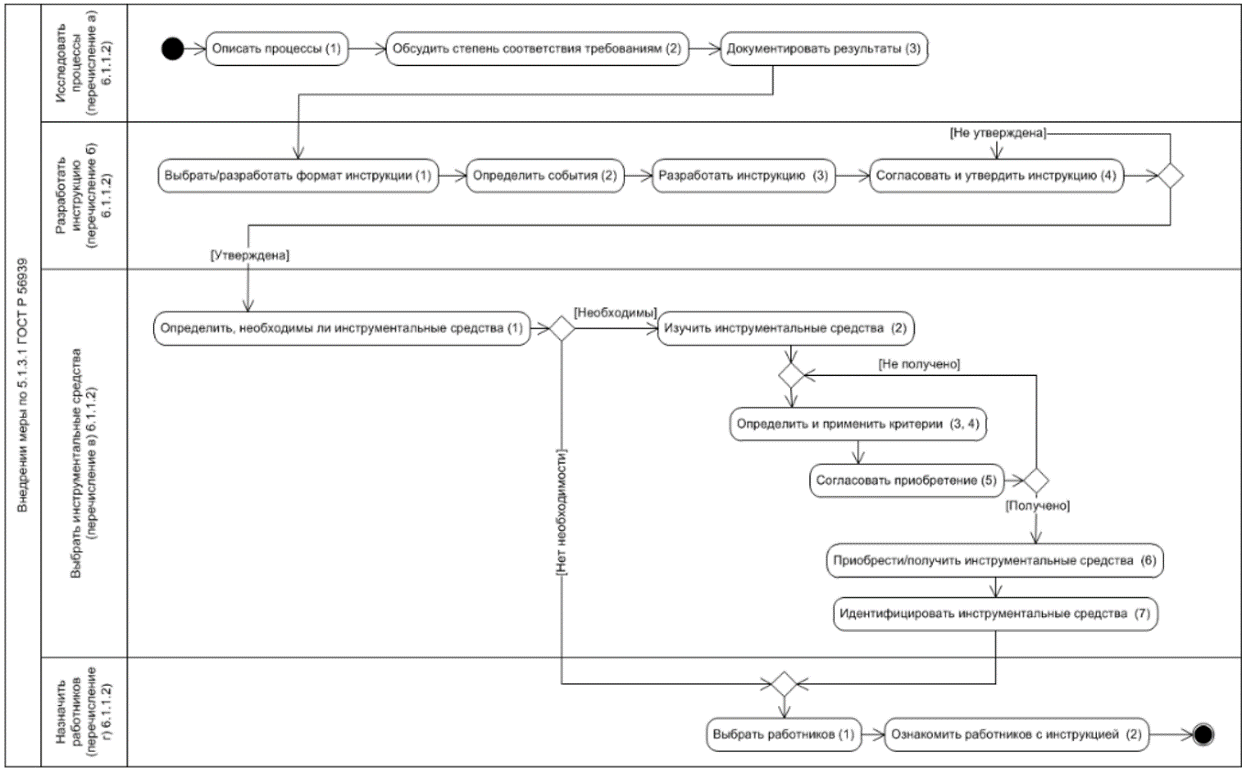


Рис. 9. Полуформальная модель

2. Исполнителями меры могут быть сотрудники организации или внешних организаций, обладающие компетенцией в области выявления уязвимостей программы, для актуальной версии исходного кода программы. Предпочтения по назначению ответственных указаны на рис. 9.

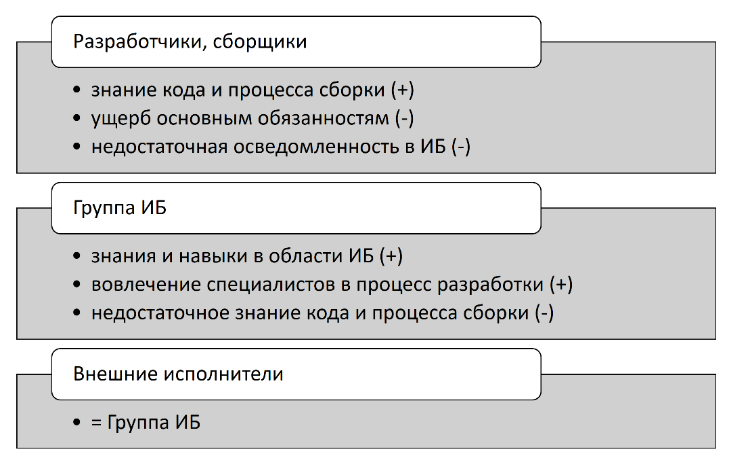


Рис. 10. Предпочтения по выбору ответственных лиц

3. Периодичность и условия выполнения действий по статическому анализу определяются исходя из особенностей процесса разработки и комплексирования, степени его автоматизации, применяемого ПО среды разработки, временных параметров процесса разработки и статического анализа и пр. Статический анализ может проводиться при выполнении одного или нескольких из следующих условий:

- любое изменение в исходном коде (т.е. в процессе генерации кода);

- любое преобразование исходного кода в исполняемый;

- выпуск любых промежуточных версий программ;

- выпуск версий ПО, предназначенных для передачи пользователю;

- наступление определенных временных условий, переход процесса разработки на новую стадию, другие условия.

4. Так как статический анализ проводится с использованием инструментальных средств, то, после идентификации, их выбор может быть выполнен на основании ряда технических критериев (языки программирования, типы выявляемых дефектов, методы и уровень статического анализа, среда интеграции, наличие документации, отчетность, производительность, ресурсоемкость, лицензии и пр.) и др. [10].

5. Документирование процедур статического анализа может содержать следующие сведения:

- перечень лиц, ответственных за проведение статического анализа, с указанием их обязанностей;

- описание порядка, условий и периодичности проведения статического анализа;

- идентификационные данные инструментальных средств для проведения статического анализа;

- ссылки на руководства по использованию инструментальных средств для проведения статического анализа;

- описание объема и содержимого документируемых материалов, являющихся результатами проведения статического анализа, и порядка обработки результатов и др.

6. Завершаемыми шагами являются определение порядка информирования и систематического (периодического) контроля.

После описания реализации меры, как отмечалось выше, следует определить перечень документации, который в данном случае включает в себя:

- инструкции по реализации меры (разработанные с учетом п. 6.3.4.2 ГОСТ Р 56939);

- документы разработчика ПО (соответствующие требованиям 5.3.3.4 ГОСТ Р 56939).

Рекомендуемое распределение ролей и обязанностей, связанных с реализацией меры по разработке безопасного ПО, выглядит следующим образом:

- архитектор безопасности ПО, руководитель разработки ПО определяют границы проведения статического анализа и устанавливают условия и способы проведения статического анализа;

- руководители групп разработки ПО, специалисты группы разработки ПО обеспечивают проведение статического анализа исходного кода и обеспечивают анализ результатов проведения статического анализа;

- специалисты группы разработки безопасного ПО обеспечивают анализ результатов проведения статического анализа.

**Благодарности**

Авторы выражают благодарность кандидату технических наук Барабанову А.В. и доценту Шахалову И.Ю. за важные рекомендации при подготовке материала.

**Выводы**

Концептуальные вопросы разработки безопасных программ в настоящее время нашли свое решение в рамках выхода национального стандарта ГОСТ Р 56939, однако выбор собственно мер безопасности целесообразно выполнять при учёте актуальных угроз, значимых рисков, а также имеющихся программных сред, платформ, сетевых топологий, ИТ- архитектор и т.п.

Практические аспекты внедрения указанных мер подлежат развитию в виде перспективного перечня рекомендаций. Практические рекомендации удобно сочетать с иллюстративным материалом, представленным разного рода концептуальными (семантическими) моделями.

Перспективным направлением стандартизации может быть разработка руководства по оценке соответствия соответствующих систем менеджмента, а также производств защищённых программных изделий.

Данное исследование проводилось в рамках работы Технического комитета по стандартизации ТК 362 «Защита информации».

Список литературы

1. Барабанов А.В., Марков А.С., Цирлов В.Л. Актуальные вопросы выявления уязвимостей и недекларированных возможностей в программном обеспечении // Системы высокой доступности. 2018. Т. 14. № 3. С. 12-17.
2. Барабанов А.В., Марков А.С., Цирлов В.Л. Методический аппарат анализа и синтеза комплекса мер разработки безопасного программного обеспечения// Программные продукты и системы № 4 (112), 2015. – C.166 – 175.
3. Бегаев А.Н., Тарасюк М.В. Контроль безопасности программного кода в составе объекта информатизации//Защита информации. Инсайд. 2013. № 5 (53). С. 63-67.
4. Бородакий Ю.В., Жуков И.Ю., Зубарев И.В., Костогрызов А.И., Родионов В.Н. и др. Методическое руководство по оценке качества функционирования информационных систем. М.: Изд-во 3 ЦНИИ МО РФ, 2003. 352 с.
5. Жидков И.В., Кадушкин И.В. О признаках потенциально опасных событий в информационных системах//Вопросы кибербезопасности. 2014. № 1 (2). С. 40-48.
6. Козачок А.В., Кочетков Е.В. Обоснование возможности применения верификации программ для обнаружения вредоносного кода//Вопросы кибербезопасности. 2016. № 3 (16). С. 25-32.
7. Костогрызов А.И., Липаев В.В. Сертификация функционирования автоматизированных информационных систем. М.: Изд. «Вооружение. Политика. Конверсия», 1996. 280 с.
8. Ломако А. Г., Еремеев М. А., Новиков В. А. Метод выявления дефектов и недокументированных возможностей программ//Информационное противодействие угрозам терроризма. -2010. -№ 14. -С. 46-49.
9. Марков А.С., Матвеев В.А., Фадин А.А., Цирлов В.Л. Эвристический анализ безопасности программного кода//Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия: Приборостроение. 2016. № 1 (106). С. 98-111 DOI: 10.18698/0236-3933-2016-1-98-111.
10. Марков А.С., Фадин А.А., Швец В.В. Сравнение статических анализаторов безопасности программного кода // Защита информации. Инсайд. 2015. № 6 (66). С. 38-47.
11. Невлюдов И.Ш., Андрусевич А.А., Евсеев В.В. Анализ жизненного цикла разработки программного обеспечения для корпоративных информационных систем // Восточ.-Европ. журн. передовых технологий. 2010. Т. 6. № 8 (48). С. 25–27.
12. Петренко С. А. Концепция поддержания работоспособности киберсистем в условиях информационно-технических воздействий//Труды ИСА РАН. -2009. -Т. 41. Управление рисками и безопасностью. -C. 175-193.
13. Рибер Г., Малмквист К., Щербаков А. Многоуровневый подход к оценке безопасности программных средств // Вопросы кибербезопасности. 2014. № 1 (2). С. 36–39.
14. Barabanov A., Grishin M., Markov A., Tsirlov V. Current Taxonomy of Information Security Threats in Software Development Life Cycle. In: 2018 IEEE 12th International Conference Application of Information and Communication Technologies (AICT). IEEE (17-19 Oct 2018, Almaty, Kazakhstan). 2018, pp. 356-361.
15. 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: 10.1145/2799979.2799998
16. Reed M., Miller J.F., and Popick P. Supply Chain Attack Patterns: Framework and Catalog. Office of the Deputy Assistant Secretary of Defense for Systems Engineering, 2014. 88 p. URL: <https://www.acq.osd.mil/se/docs/Supply-Chain-WP.pdf>.

References

1. Barabanov A.V., Markov A.S., Cirlov V.L. Aktual'nye voprosy vyyavleniya uyazvimostej i nedeklarirovannyh vozmozhnostej v programmnom obespechenii // Sistemy vysokoj dostupnosti. 2018. T. 14. № 3. S. 12-17.

2. Barabanov A.V., Markov A.S., Cirlov V.L. Metodicheskij apparat analiza i sinteza kompleksa mer razrabotki bezopasnogo programmnogo obespecheniya// Programmnye produkty i sistemy № 4 (112), 2015. – C.166 – 175.

3. Begaev A.N., Tarasyuk M.V. Kontrol' bezopasnosti programmnogo koda v sostave ob"ekta informatizacii//Zashchita informacii. Insajd. 2013. № 5 (53). S. 63-67.

4. Borodakij YU.V., ZHukov I.YU., Zubarev I.V., Kostogryzov A.I., Rodionov V.N. i dr. Metodicheskoe rukovodstvo po ocenke kachestva funkcionirovaniya informacionnyh sistem. M.: Izd-vo 3 CNII MO RF, 2003. 352 s.

5. ZHidkov I.V., Kadushkin I.V. O priznakah potencial'no opasnyh sobytij v informacionnyh sistemah//Voprosy kiberbezopasnosti. 2014. № 1 (2). S. 40-48.

6. Kozachok A.V., Kochetkov E.V. Obosnovanie vozmozhnosti primeneniya verifikacii programm dlya obnaruzheniya vredonosnogo koda//Voprosy kiberbezopasnosti. 2016. № 3 (16). S. 25-32.

7. Kostogryzov A.I., Lipaev V.V. Sertifikaciya funkcionirovaniya avtomatizirovannyh informacionnyh sistem. M.: Izd. «Vooruzhenie. Politika. Konversiya», 1996. 280 s.

8. Lomako A. G., Eremeev M. A., Novikov V. A. Metod vyyavleniya defektov i nedokumentirovannyh vozmozhnostej programm//Informacionnoe protivodejstvie ugrozam terrorizma. -2010. -№ 14. -S. 46-49.

9. Markov A.S., Matveev V.A., Fadin A.A., Cirlov V.L. Evristicheskij analiz bezopasnosti programmnogo koda//Vestnik Moskovskogo gosudarstvennogo tekhnicheskogo universiteta im. N.E. Baumana. Seriya: Priborostroenie. 2016. № 1 (106). S. 98-111 DOI: 10.18698/0236-3933-2016-1-98-111.

10. Markov A.S., Fadin A.A., SHvec V.V. Sravnenie staticheskih analizatorov bezopasnosti programmnogo koda // Zashchita informacii. Insajd. 2015. № 6 (66). S. 38-47.

11. Nevlyudov I.SH., Andrusevich A.A., Evseev V.V. Analiz zhiznennogo cikla razrabotki programmnogo obespecheniya dlya korporativnyh informacionnyh sistem // Vostoch.-Evrop. zhurn. peredovyh tekhnologij. 2010. T. 6. № 8 (48). S. 25–27.

12. Petrenko S. A. Koncepciya podderzhaniya rabotosposobnosti kibersistem v usloviyah informacionno-tekhnicheskih vozdejstvij//Trudy ISA RAN. -2009. -T. 41. Upravlenie riskami i bezopasnost'yu. -C. 175-193.

13. Riber G., Malmkvist K., SHCHerbakov A. Mnogourovnevyj podhod k ocenke bezopasnosti programmnyh sredstv // Voprosy kiberbezopasnosti. 2014. № 1 (2). S. 36–39.

14. Barabanov A., Grishin M., Markov A., Tsirlov V. Current Taxonomy of Information Security Threats in Software Development Life Cycle. In: 2018 IEEE 12th International Conference Application of Information and Communication Technologies (AICT). IEEE (17-19 Oct 2018, Almaty, Kazakhstan). 2018, pp. 356-361.

15. 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: 10.1145/2799979.2799998

16. Reed M., Miller J.F., and Popick P. Supply Chain Attack Patterns: Framework and Catalog. Office of the Deputy Assistant Secretary of Defense for Systems Engineering, 2014. 88 p. URL: <https://www.acq.osd.mil/se/docs/Supply-Chain-WP.pdf>.

1. Имеется стандарт ISO/IEC 20004, который определяет подход к выявлению уязвимостей при оценке соответствия ИТ-продуктов в учетом ISO/IEC 15408. [↑](#footnote-ref-1)