СТАНДАРТИЗАЦИЯ ПРОЦЕССА РАЗРАБОТКИ БЕЗОПАСНЫХ ПРОГРАММНЫХ СРЕДСТВ

Барабанов Александр Владимирович, *CISSP, CISSLP*

Статья посвящена вопросам стандартизации процесса разработки безопасных программных средств, рассмотрены основные положения разрабатываемого ФСТЭК России ГОСТ Р «Защита информации. Требования по обеспечению безопасности разработки программного обеспечения».

Ключевые слова: *безопасность программ, уязвимости, дефекты безопасности, сертификация*

THE STANDARDIZATION OF THE PROCESS OF DEVELOPING A SECURITY SOFTWARE

Alex Barabanov, *CISSP, CISSLP*

The standardization process of developing secure software are discussed. National standard GOST R «Information Security. Security requirements of software development» are considered.

Keywords: *security software, vulnerability, security defects*

Анализ современных тенденций в области информационной безопасности (ИБ) показывает, что на протяжении последних лет наблюдается устойчивый рост количества нарушений ИБ, приводящих к снижению уровня целостности, доступности и конфиденциальности информационных ресурсов автоматизированных систем (рис. 1).

Следует отметить, что большинство нарушений ИБ связано, в первую очередь, с наличием уязвимостей в программных средствах (ПС), используемых в автоматизированных системах. Опыт работы аккредитованной испытательной лаборатории, проводящей сертификационные испытаний ПС в различных системах сертификации, показывает, что даже в продуктах, представляемых разработчиками на сертификацию, содержаться уязвимости и дефекты безопасности.

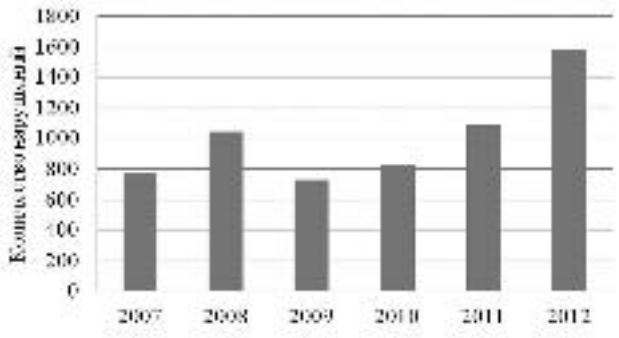

Рис.1 *Количество нарушений информационной безопасности в информационных системах*

Наиболее распространенными[1] типами уязвимостей, выявляемых в ПС, подаваемом сертификацию являются: аутентификационные данные в исходном коде ПС, межсайтовый скрптинг, внедрение SQL-кода (рис. 2).

Уязвимости обнаруживаются как с использованием методов структурного тестирования (как правило, статический сигнатурный анализ), так и с использованием функционального тестирования (в случае отсутствия доступа к исходным кодам ПС).

До недавнего времени наблюдался некий нормативный и методический вакуум, не позволяющий испытательным лабораториям, обоснованно и целенаправленно заниматься выявлением уязвимостей в ПС, подаваемых на сертификацию. Ситуация изменилась с введением ФСТЭК России нового подхода к формированию требований к сертифицируемым ПС и действиям испытательных лабораторий при проведении испытаний - в настоящее время выполнение тестирования на проникновения является обязательной процедурой, которое должна выполнить испытательная лаборатория при проведении сертификационных испытаний.

---

[1] Статистическая информация основана на результатах сертификационных испытаний, проведенных компанией НПО «Эшелон».

***Рис. 2**. Распределение дефектов безопасности программного обеспечения по типам*

Следует отметить, что тестирование на проникновение может выполняться как на основе анализа исходных текстов (если разработчик предоставляет к ним доступ), так и по методологии «черного ящика».

При сертификации по низким классам защищенности общепринятой практикой является следующая:

- изучение списка известных уязвимостей, опубликованных в открытых источниках (например, osvdb.org, securityfocus.com), и формирование списка потенциальных уязвимостей, которые могут быть в ПС;

- проверка выдвинутых гипотез относительно уязвимостей ПС (тестирование с использованием различных инструментальных средств).

При сертификации по высоким классам защищенности, когда предоставление доступа к исходным кодам ПС является обязательным, тестирование на проникновение основывается на результатах статического и динамического анализа исходного кода ПС [2, 3].

Другим важным направлением развития, нацеленным на повышение качества ПС и уменьшение числа уязвимостей и дефектов безопасности, является внедрение цикла разработки безопасных ПС. Опыт компании Microsoft показал, что внедрение цикла разработки безопасных ПС позволило сократить число уязвимостей в ПС компании в среднем на 80%.

В настоящее время ФСТЭК России ведется работа над государственным стандартом «Защита информации. Требования по обеспечению безопасности разработки программного обеспечения». Кроме этого, в соответствии с нормативным правовым актом ФСТЭК России «Состав и содержание организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных», использование в информационной системе системного и (или) прикладного программного обеспечения, разработанного с использованием методов защищенного программирования, является дополнительной мерой по обеспечению безопасности информации в случае определения в качестве актуальных угроз безопасности персональных данных 1-го и 2-го типов. В тоже время документ, определяющий содержание и порядок выполнения работ по созданию программного обеспечения с использованием методов защищенного программирования, отсутствует. Эти положения и определяют актуальность разработки ГОСТ.

При разработке первой редакции проекта ГОСТ учитывались следующие особенности:

1. Поскольку известно, что стоимость устранения уязвимостей и дефектов безопасности ПС выше на поздних стадиях проектирования, ГОСТ должен обеспечить внедрение необходимых процедур на самых ранних стадиях проектирования ПС.

2. ГОСТ должен учитывать современные тенденции разработки безопасных ПС, учитывать положения «лучших практик» (например, Microsoft SDL, Cisco SDL) [4-7].

3. ГОСТ должен быть полностью совместим с методологией разработки и сертификации ПС, используемой в настоящее время ФСТЭК России («Общие критерии») [3].

4. Разрабатываемый ГОСТ должен обеспечить возможность интеграции процедур разработки безопасных ПС с существующей на предприятии системой управления ИБ (например, построенной в соответствии с ГОСТ Р ИСО/МЭК 27001 «Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования») [1].

Анализ современных методологий (Таблица 1) проектирования безопасных ПС позволил сформулировать следующий перечень процедур, которые должны быть реализованы разработчиком ПС:

- процедуры управления конфигурацией;

- процедуры определения модели жизненного цикла;

- процедуры проектирования и реализации безопасных ПС;

- процедуры использования инструментальных средств и методов разработки;

- процедуры обеспечения безопасности разработки;

- процедуры поставки;

- процедуры обновления и устранения уязвимостей и дефектов безопасности ПС.

В таблице 1 представлены результаты сравнительного анализа следующих методологий проектирования безопасных ПС: «Общие критерии» [3], Microsoft SDL [6], OpenSAMM [7].

Следует отметить, что требования к реализации большинства процедур согласованы с требованиями доверия, предъявляемыми ГОСТ Р ИСО/МЭК 15408-3 (Таблица 2).

Таблица 1 *– Результаты сравнительного анализа методологий проектирования безопасных ПС*

Характеристика	«Общие критерии»	Microsoft SDL	Open SAMM
Обучение специалистов	-	+	+
Обеспечение физической и логической безопасности при разработке ПС	+	-	-
Управление конфигурацией	+	-	-
Моделирование угроз	+	+	+
Определение требований безопасности к разрабатываемым ПС	+	+	+
Использование принципов безопасного проектирования	+	+	+
Анализ исходных кодов ПС	-	+	+
Анализ уязвимостей	+	+	+
Использование методов проектирования безопасных ПС	-	+	+
Обеспечение безопасности поставки	+	-	+

Таблица 2– *Соответствие требований разрабатываемого стандарта требованиям ГОСТ Р ИСО/МЭК 15408-3*

Требование разрабатываемого стандарта	Требование ГОСТ Р ИСО/МЭК 15408-3
Требования к функциональным возможностям системы управления конфигурацией	ACM_CAP.1, ACM_CAP.2, ACM_CAP.3, ACM_CAP.4, ACM_CAP.5
Требования к области действия системы управления конфигурации	ACM_SCP.1, ACM_SCP.2, ACM_SCP.3
Требования к средствам автоматизации процесса управления конфигурацией	ACM_AUT.1, ACM_AUT.2
Требования к процедурам определения модели жизненного цикла	ALC_LCD.1, ALC_LCD.2, ALC_LCD.3
Требования к процедурам проектирования и реализации безопасных ПС	ATE_FUN.1, ATE_FUN.2, AVA_VLA, ADV_FSP, ADV_HLD, ADV_LLD, ADV_RCR
Требования к процедурам использования инструментальных средств и методов разработки	ALC_TAT.1, ALC_TAT.2, ALC_TAT.3
Требования к обеспечению безопасности разработки	ALC_DVS.1, ALC_DVS.2
Требования к процедуре поставки	ADO_DEL.1, ADO_DEL.2, ADO_DEL.3, ADO_ISG, AGD_ADM, AGD_USR
Требования к реализации процедур обновления и устранения недостатков	ALC_FLR.1, ALC_FLR.2, ALC_FLR.3

Отдельно рассмотрим перечень основных требований к процедурам проектирования и реализации безопасных ПС, сформулированные по результатам анализа «лучших практик» и обобщения опыта работы аккредитованных испытательных лабораторий систем сертификации средств защиты информации.

1. Должно проводиться периодическое обучение сотрудников разработчика ПС с целью повышения их осведомленности в области разработки безопасных ПС. Выполнение данного требования позволяет обеспечить повышение осведомленности в области безопасной разработки ПС сотрудников, связанных с разработкой ПС и, как следствие, повышение уровня безопасности разрабатываемых ПС. В программу обучения сотрудников могут входит курсы безопасного программирования, инспекции кода, тестирования на проникновение, статического анализа, динамического анализа, функционального тестирования и другие.

2. Разработчиком ПС должны быть определены и документированы требования безопасности к разрабатываемому ПС. Например, могут быть определены следующие классы требований: требования к обеспечению конфиденциальности, требования к обеспечению идентификации и аутентификации, требования к реализации разграничения доступа, требования к обработке ошибок и исключений ПС.

3. Проектирование ПС должно выполняется с использованием принципов проектирования безопасных ПС. При выполнении проектирования ПС разработчиком могут, например, использоваться следующие принцип проектирования безопасных ПС: принцип эшелонированной защиты, принцип минимальных привилегий, принцип модульного проектирования, принцип разделения обязанностей.

4. При проектировании ПС разработчиком ПС должно выполняться моделирование угроз с целью выявления уязвимостей ПС этапа проектирования, результаты моделирования угроз должны документироваться.

5. Разработка ПС должна выполняться с использованием методов защищенного программирования. Например, в ходе разработки ПС разработчик должен избегать использования скомпрометированных (небезопасных) функций в исходных кодах ПС. В качестве источника небезопасных функций могут использоваться, например, списки «Security Development Lifecycle Banned Function Calls» компании Microsoft.

6. Должен проводиться периодический статический анализ исходных кодов ПС с целью выявления уязвимостей и дефектов кода, результаты анализа должны документироваться.

7. Должна проводиться периодическая инспекция кода с целью выявления уязвимостей и дефектов кода, результаты инспекции должны документироваться.

8. Должно проводиться и документироваться тестирование (функциональное, нагрузочное) ПС.

9. Должен проводиться динамический анализ исходных кодов ПС с целью выявления уязвимостей и дефектов кода, результаты анализа должны документироваться.

10. Должно проводиться тестирование на проникновение с целью выявления уязвимостей ПС, результаты тестирования должны документироваться.

11. Должны быть разработаны функциональная спецификация ПС, проектная документация (проект верхнего уровня, проект нижнего уровня) и документация, демонстрирующая соответствие между всеми смежными парами имеющихся представлений ПС.

Планируется, что будет введена некоторая градация, которая позволит предъявлять требования к организациям-разработчикам в зависимости от критичности создаваемого ПС.

Внедрение подобных процедур в российские организации-разработчики ПС, на наш взгляд, повысить уровень защищенности создаваемых ПС и, как следствие, значительно уменьшить число инцидентов информационной безопасности.

Литература

1. Дорофеев А.В., Шахалов И.Ю. Основы управления информационной безопасностью современной организации // Правовая информатика. 2013. № 3. С. 4-14.
2. Марков А.С., Цирлов В.Л. Сертификация программ: мифы и реальность // Открытые системы. СУБД. 2011. № 6. С. 26-29.
3. Марков А.С., Цирлов В.Л., Барабанов А.В. Методы оценки несоответствия средств защиты информации. М.: Радио и связь, 2012. 192 с.

4. Building Security into Your Software Development Lifecycle. Coverity White Paper. 2012. 14 p.
5. Cisco Secure Development Lifecycle. Cisco White Paper. 2013. 4 p.
6. Howard M., Lipner S. The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software. Microsoft Press. 2006. 352 p.
7. Software Assurance Maturity Model: A guide to building security into software development. Ver. 1.0. OWASP. 2013. 96 p.

References

1. Dorofeyev A.V., Shakhalov I.Yu. Osnovy upravleniya informatsionnoy bezopasnostyu sovremennoy organizatsii, (Basics of Information Security Management modern organization), Pravovaya informatika, 2013, No 3, pp. 4-14.
2. Markov A.S., Tsirlov V.L. Sertifikatsiya programm: mify i realnost, (Certification programs: myths and reality), Otkrytyye sistemy. SUBD, (Open Systems Journal), 2011, No 6, pp. 26-29.
3. Markov A.S., Tsirlov V.L., Barabanov A.V. Metody otsenki nesootvetstviya sredstv zashchity informatsii, Moscow, Radio i Svyaz, 2012, pp. 1-192.
4. Building Security into Your Software Development Lifecycle. Coverity White Paper. 2012. 14 p.
5. Cisco Secure Development Lifecycle. Cisco White Paper. 2012. 14 p.
6. Howard M., Lipner S. The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software. Microsoft Press. 2006. 352 p.
7. Software Assurance Maturity Model: A guide to building security into software development. Ver. 1.0. OWASP. 2013. 96 p.

Рецензент: ***Гарбук Сергей Владимирович****, к.т.н, доцент*