

Тематика безопасной разработки ПО является сегодня одной из наиболее актуальных в сфере обеспечения информационной безопасности. Проектирование компьютерных программ с учетом требований безопасности – это процесс устранения ошибок на протяжении всего жизненного цикла разработки ПО, начиная с ранней его стадии. Подобный подход позволяет значительно снизить риски, связанные с последующей эксплуатацией готового продукта, поэтому методы безопасной разработки программ сейчас внедряются практически во всех известных продуктовых компаниях.

В пособии изложены концептуальные положения и определения, связанные с созданием безопасного программного обеспечения, выделены некоторые из лучших практик в данной области, которым следуют известные мировые компании. Кроме того, в документе представлены передовые российские решения в сфере безопасной разработки ПО, описываются организационные ресурсы и процессы планирования, необходимые для внедрения этих передовых практик и перечисляются некоторые ключевые ресурсы, рекомендованные компаниями-лидерами индустрии кибербезопасности и рядом международных объединений. Помимо этого в материале дан обзор полезных программ для практики безопасной разработки, в Приложениях приведена нормативная документация по рассматриваемой тематике.

В первую очередь, пособие представляет интерес для компаний-разработчиков программного и аппаратного обеспечения, облачных и системных решений, но будет полезно и для других учреждений и организаций, регулирующих органов, а также для студентов и преподавателей вузов, в которых готовят программистов и специалистов по информационной безопасности.

## Введение

### Глава 1. Лучшие практики безопасной разработки

- 1.1. Безопасные концепции и терминология проектирования
  - 1.1.1. Безопасность при проектировании
  - 1.1.2. Безопасность по умолчанию
  - 1.1.3. Жизненный цикл разработки системы безопасности
  - 1.1.4. Надежность
- 1.2. Основные элементы безопасной разработки
  - 1.2.1. Моделирование угроз
  - 1.2.2. Безопасность цепочки поставок и третьих сторон
  - 1.2.3. Безопасная разработка и развертывание
  - 1.2.4. Процессы и поддержка курса на снижение уязвимостей
  - 1.2.5. Оценка эффективности
  - 1.2.6. Доверие и сотрудничество
  - 1.2.7. Исправление и поддержка
- 1.3. Корректировка мышления и внутренних процессов

### Глава 2. Технология безопасной разработки программного обеспечения DevSecOps

- 2.1. О технологии DevSecOps
- 2.2. Развитие SSDLC
- 2.3. Автоматизация (DevSecOps)
- 2.4. ИБ-тестирование
- 2.5. Архитектурные проверки
- 2.6. Поддержка Bug Bounty
- 2.7. Обучение сотрудников
- 2.8. Работа с инцидентами ИБ
- 2.9. Безопасность как движущая сила бизнеса
- 2.10. Факторы, усложняющие обеспечение безопасности программного обеспечения
- 2.11. Защита от различных угроз
- 2.12. Проблемы, связанные с построением надежной системы DevSecOps
- 2.13. Преодоление сложности и вариативности конвейера DevOps

- 2.14. Управление распределенной разработкой и тестированием
- 2.15. Проблемы с операционными и организационными нагрузками
- 2.16. Интеграция инструментов обеспечения безопасности
- 2.17. Использование политик и автоматизации для установки защитных механизмов
- 2.18. Предоставление информации о рисках безопасности непосредственно рабочим процессам разработчиков
- 2.19. Рекомендации по реализации интегрированного DevSecOps
  - 2.19.1. Обеспечение безопасности в среде IDE
  - 2.19.2. Трансформация функциональных тестов в тесты безопасности
- 2.20. Централизация анализа рисков безопасности и координация тестирования
  - 2.20.1. Масштабирование тестирования безопасности приложений без увеличения нагрузки

**Глава 3. Инструменты DevSecOps, реализующие лучшие практики обеспечения безопасности ПО и процессов DevOps**

- 3.1. Acunetix
- 3.2. Aqua Security
- 3.3. Checkmarx
- 3.4. Fortify Webinspect
- 3.5. GitHub Actions
- 3.6. OWASP Zed Attack Proxy (ZAP)
- 3.7. Snyk
- 3.8. SonarQube
- 3.9. ThreatModeler
- 3.10. Trivy
- 3.11. Veracode

**Глава 4. Концепция разработки безопасного программного обеспечения на единой цифровой платформе Российской Федерации «ГосТех»**

- 4.1. Перечень терминов, сокращений и обозначений
  - 4.1.1. Сокращения
  - 4.1.2. Термины и определения
- 4.2. Общие положения
- 4.3. Правовая основа концепции
- 4.4. Цели и задачи разработки безопасного программного обеспечения на платформе «ГосТех»
- 4.5. Требования по разработке безопасного программного обеспечения на платформе «ГосТех»
- 4.6. Требования к субъектам и объектам платформы «ГосТех»
- 4.7. Процессы и организационные меры по разработке безопасного ПО
- 4.8. Технические меры по разработке безопасного ПО на платформе «ГосТех»
- 4.9. Повышение и поддержание на должном уровне компетенций в области разработки безопасного ПО
- 4.10. Подтверждение соответствия программного обеспечения требованиям по разработке безопасного ПО
- 4.11. Порядок пересмотра концепции

**Глава 5. Технологии для безопасной разработки ПО от ИСП РАН**

- 5.1. Анализ исходного кода, верификация, тестирование
  - 5.1.1. ASTRAYER TOOLSET: система верификации ключевых компонентов
  - 5.1.2. KLEVER: система верификации моделей промышленного ПО
  - 5.1.3. MASIW: набор инструментов для проектирования ответственных систем
  - 5.1.4. MICROTesk: генератор тестовых программ
  - 5.1.5. SAFEC: безопасный компилятор
  - 5.1.6. SVACE: статический анализатор исходного кода
  - 5.1.7. TESTOS: окружение для тестирования ПО

## 5.2. Анализ бинарного кода, фаззинг

- 5.2.1. БЛЕСНА: инструмент динамического анализа помеченных данных
- 5.2.2. Инструмент диверсификации: комплекс защиты от эксплуатации уязвимостей
- 5.2.3. Платформа для анализа программ на основе эмулятора QEMU
- 5.2.4. ИСП CRUSHER: комплекс динамического и статического анализа бинарного кода
- 5.2.5. BINSIDE: статический анализатор бинарного кода
- 5.2.6. CASR: инструмент формирования отчетов об ошибках
- 5.2.7. NATCH: инструмент определения поверхности атаки
- 5.2.8. SYDR + SYDRFUZZ: комплекс гибридного фаззинга и динамического анализа

## 5.3. Анализ сетевого трафика

- 5.3.1. PROTOSPHERE: система анализа сетевого трафика

## 5.4. Управление требованиями

- 5.4.1. Requality: инструмент управления требованиями

## Глава 6. Технологии для безопасной разработки ПО от Сбера и VI.Zone

### 6.1. Позиция Сбербанка в области информационной безопасности

### 6.2. Технологии VI.Zone

- 6.2.1. Платформа для непрерывного контроля безопасности разрабатываемых приложений

## Глава 7. Российские продукты Start X для безопасной разработки ПО

### Заключение

### Литература

**Приложение А.** Источники нормативных требований РФ по безопасности проектирования и эксплуатации информационных продуктов систем и услуг

**Приложение Б.** Рекомендуемые информационные ресурсы (аннотированный перечень)