XXXI НАУЧНО-ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ

Комплексная защита информации

Модель процесса безопасной разработки программных средств криптографической защиты информации

Герасимов Вячеслав Александрович

НИИ ТЗИ, Минск, Республика Беларусь

Москва · 7 октября 2026

Эволюция организации разработки

Без формального процесса	SDLC (Software Development Life Cycle) - жизненный цикл разработки программного обеспечения (ПО)	SSDLC (Secure Software Development Life Cycle) - безопасный жизненный цикл разработки ПО
Решения зависят от отдельных специалистов	Этапы, роли и результаты определены	Безопасность встроена в каждый этап
Готовность определяется ситуативно	Готовность по требованиям и испытаниям	Готовность включает подтверждённую безопасность
Состав выпуска трудно восстановить	Конфигурация и изменения управляются	С выпуском связаны проверяемые доказательства

Использование открытых исходных текстов и доверие к выпуску

ДОСТУПНО ДЛЯ АНАЛИЗА

Исходный код
История изменений
Состав зависимостей

ТРЕБУЕТ ПОДТВЕРЖДЕНИЯ

Происхождение компонента
Поведение при сборке
Соответствие кода поставке

Для СКЗИ под угрозой оказываются ключи, ГСЧ и доверие к обновлению.

XZ Utils: закладка в сборочной цепочке

2024

ПРЕДНАМЕРЕННОЕ ВМЕШАТЕЛЬСТВО

МЕХАНИЗМ

Закладка в XZ 5.6.0 / 5.6.1, различия архива и Git

РИСК

Вмешательство в SSH-аутентификацию при определённых условиях

ПОСЛЕДСТВИЯ

Срочный отказ от затронутых сборок и откат версий

Проверка репозитория не охватывает весь путь до бинарного файла.

Post Office Horizon: ложные недостачи

2019

СУДЕБНОЕ РЕШЕНИЕ ПО HORIZON

СИСТЕМА

Учёт операций в почтовых отделениях Великобритании

ДЕФЕКТЫ

Ошибки ПО могли искажать балансы отделений

ПОСЛЕДСТВИЯ

Взыскания, банкротства и ошибочные обвинительные приговоры

Проверяемость данных и расследование жалоб входят в безопасность процесса.

event-stream: зависимость кошелька Copay

2018

ЦЕЛЕВАЯ АТАКА ЧЕРЕЗ ЗАВИСИМОСТЬ

МЕХАНИЗМ

event-stream получил вредоносную зависимость flatmap-stream

ЦЕЛЬ

Данные и закрытые ключи кошелька Copay

ПОСЛЕДСТВИЯ

Удаление пакетов, исправление приложения, перенос средств

Контроль должен охватывать транзитивные зависимости и сборочную среду.

Debian OpenSSL: предсказуемые ключи

2008

ОШИБКА В КРИПТОГРАФИЧЕСКОМ КОМПОНЕНТЕ

ДЕФЕКТ

Ослабленный генератор случайных чисел в Debian OpenSSL

РИСК

Предсказуемые ключи и упрощение их подбора

ПОСЛЕДСТВИЯ

Обновление ПО и замена недостоверных ключей

Исправленная программа не восстанавливает безопасность ранее созданных ключей.

Patriot: потеря точности расчёта времени

1991

ПВО
Дахран, 25 февраля

ПРИЧИНА

Усечение двоичного представления 0,1 при пересчёте времени

УСЛОВИЯ

Более 100 часов работы, ошибка времени около 0,34 с

ПОСЛЕДСТВИЯ

Батарея не перехватила Scud, 28 погибших

Точность вычислений и длительность работы входят в условия испытаний.

Нормативная основа модели в Беларуси

Основание	Значение для разработки
ТР 2013/027/BY	Применимая форма подтверждения соответствия
Приказ ОАЦ № 77 и СТБ 34.101	Назначение, уровень безопасности, алгоритмы и протоколы

Шесть этапов предлагаемой модели

Этап	Результат контрольной точки
1. Требования	Матрица применимости и проверяемые требования
2. Проектирование	Модель угроз и архитектура защиты
3. Реализация и поставки	Проверенный код, зависимости и сборка
4. Верификация	Протоколы и решения по замечаниям
5. Выпуск	Идентифицированный образец и комплект доказательств
6. Сопровождение	Оценка изменений и управляемые обновления

Состав и глубина контроля зависят от вида СКЗИ, уровня безопасности и среды эксплуатации.

Требования и безопасное проектирование

ТРЕБОВАНИЯ

Ключи и требования к ГСЧ
Алгоритмы и протоколы
Обновление и восстановление
Ответственный и метод проверки

ПРОЕКТНЫЕ РЕШЕНИЯ

Криптографическая граница
Минимальная доверенная база
Безопасное состояние при отказе
Защита продукта и среды разработки

Реализация и цепочка поставок

КОД

Независимый обзор критичных изменений
Статический анализ и поиск секретов
Защита репозитория и прав доступа

КОМПОНЕНТЫ И СБОРКА

SBOM и проверка зависимостей
Источник, версия, хэш
Изолированная воспроизводимая сборка

Верификация безопасности СКЗИ

КРИПТОГРАФИЧЕСКИЕ СВОЙСТВА

Тесты известных ответов
Генерация и жизненный цикл ключей

ПОВЕДЕНИЕ ПРИ ОШИБКАХ

Негативные тесты и фаззинг
Отказ, обновление и восстановление
Единый реестр замечаний

Выпуск как фиксируемая конфигурация

ЗАФИКСИРОВАННЫЕ ВХОДНЫЕ ДАННЫЕ

Исходная версия и зависимости
Компилятор и параметры
Сценарии сборки и документация

ДОКАЗАТЕЛЬСТВА ДЛЯ ОБРАЗЦА

Бинарные файлы и хэш-значения
Подписанный манифест и состав
Связанные протоколы испытаний

Сопровождение после выпуска

СИГНАЛЫ ДЛЯ АНАЛИЗА

Сообщения об уязвимостях
Изменения зависимостей и стандартов
Инциденты и условия эксплуатации

РЕЗУЛЬТАТ РЕАГИРОВАНИЯ

Оценка затронутых версий и ключей
Регрессия и корректирующий выпуск
Архив воспроизводимой конфигурации

Исправление оценивается по влиянию на продукт, доказательства и ключевой материал.

Международные практики

Область	Примеры	Роль в модели
Безопасная разработка	NIST SSDF 1.1 ISO/IEC 27034-1:2011 ГОСТ Р 56939-2024	Организация работ и управление уязвимостями
Криптографический модуль	ISO/IEC 19790:2025	Требования к свойствам модуля
Испытания модуля	ISO/IEC 24759:2025	Методы проверки и подтверждающие материалы

Универсальные практики адаптируются к национальным требованиям и назначению СКЗИ.

Результаты работы и выводы

ЧТО МЫ СДЕЛАЛИ

Объединили нормативные требования
и практики безопасной разработки
Определили контрольные точки
на этапах жизненного цикла

ЧТО ПОЛУЧИЛИ

Модель SSDLC из шести этапов
Связь требований с испытаниями
и конкретным выпуском
Критерии готовности продукта

Безопасная разработка ПСКЗИ требует контроля на всём жизненном цикле.

Модель процесса безопасной разработки программных средств криптографической защиты информации

Герасимов Вячеслав Александрович

НИИ ТЗИ, Минск, Республика Беларусь

Москва · 7 октября 2026