

Разработка безопасного ПО высокого уровня зрелости

Подходы и практики разработки
ОС SberLinux и системного ПО

Баженов Андрей
Щербаков Алексей





Баженов Андрей

Директор управления разработки системного ПО



Щербаков Алексей

Начальник центра КБ Платформы

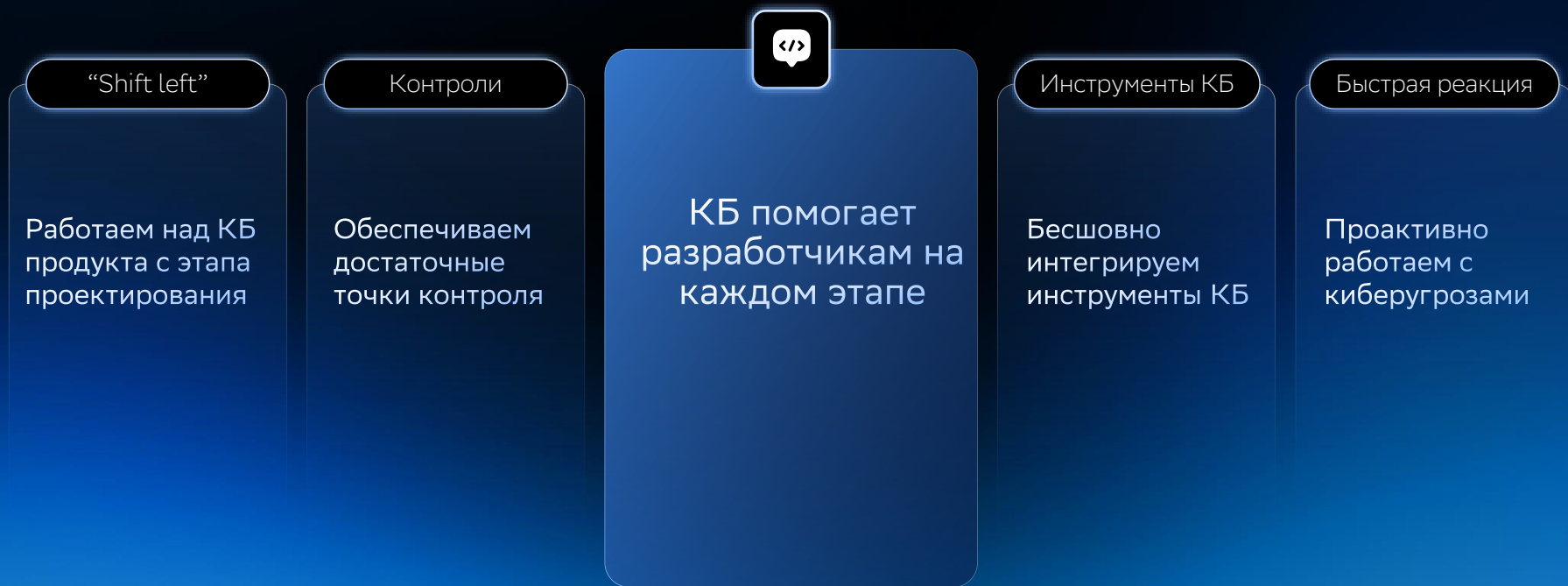
СберТех это

200+

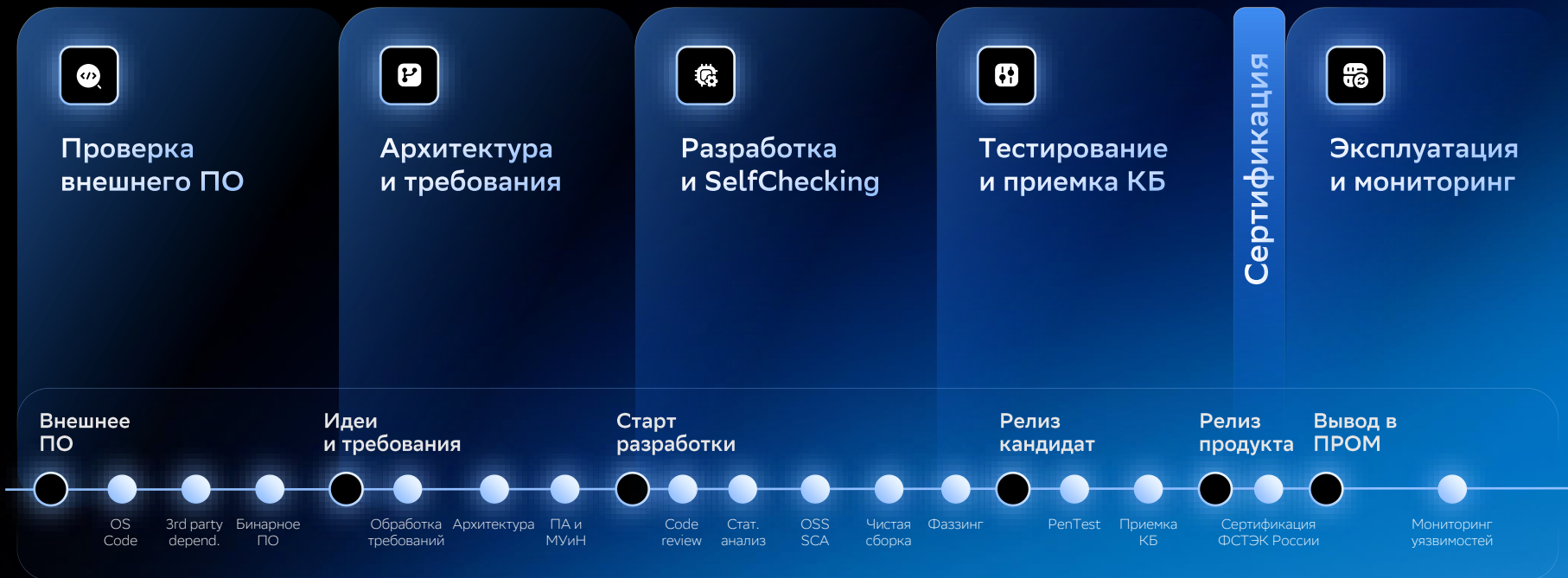
команд разработки

- ✓ Platform V, SberLinux, GitVerse
- ✓ СБЕР и Рынок
- ✓ СЗИ и системное ПО
- ✓ Цифровизация и импортозамещение

Мы стремимся к максимально эффективному взаимодействию между разработчиками и сотрудниками КБ для создания безопасного ПО



На каждом этапе разработки мы уделяем особое внимание вопросам кибербезопасности



Для всей линейки выпускаемых нами продуктов мы обеспечиваем максимальный уровень КБ

80+

продуктов

450+

релизов
в квартал

0

High и Critical
уязвимостей
в выпускаемых
релизах

100%

продуктов
проверяется
на КБ до передачи
клиенту

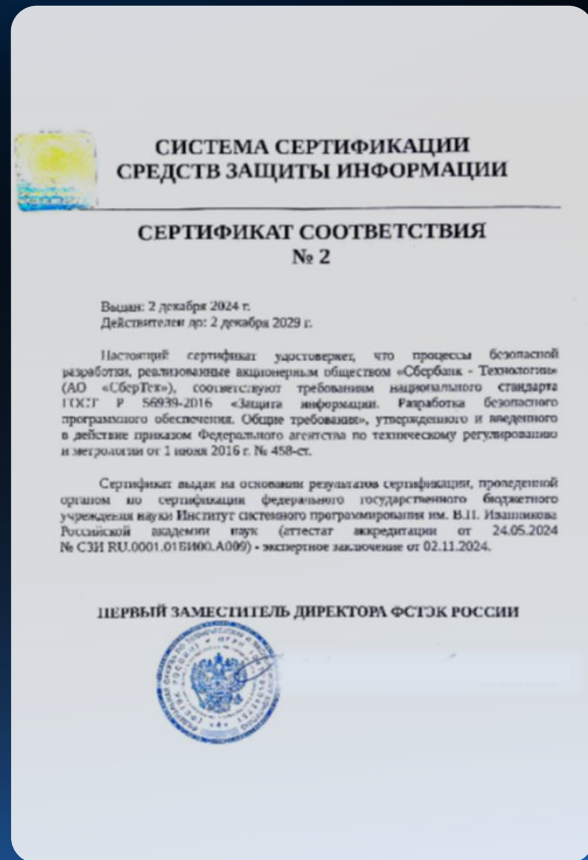
Полученный сертификат соответствия процессов РБПО ФСТЭК России подтвердил высокий уровень КБ наших процессов и продуктов

Подготовка к сертификации РБПО

- Развить все процессы РБПО до максимальной эффективности
- Проработать демо каждой меры и технически и документально
- Проводить регулярный аудит КБ для выявления «зон развития»
- Вносить вклад в деятельность ТК362 для развития стандартов КБ

Прохождение сертификации

- Выворачивать терминологию и идти от общего к частному
- Детализировать структуру производственного конвейера
- Интервью команд – необходимая часть прохождения аудита
- Предъявить доказательства реализованных процессов



Мы представляем операционную систему и платформу для автоматизации управления контейнерными приложениями

Platform V SberLinux OS Server

Российская серверная операционная система уровня enterprise, основанная на ядре Linux и системных компонентах GNU



Platform V DropApp

Российская платформа для автоматизации развертывания, масштабирования и управления контейнерными приложениями



Продукт	Аналоги	Реестр российского ПО	Сертификат ФСТЭК
PV SberLinux OS Server	RedHat Linux, Oracle Linux, Альт Линукс, РедОС, CentOS	№ 18621 от 22.08.2023	№ 4884 от 04.12.2024
PV DropApp	OpenShift, Kubernetes, Штурвал, Deckhouse	№ 18785 от 05.09.2023	№ 4883 от 04.12.2024

Создавая системное ПО, мы применяем лучшие практики, которыми хотим поделиться с сообществом

Проверка
внешнего ПО



Архитектура
и требования



Разработка
и SelfChecking



Тестирование
и приемка КБ



Сертификация



Эксплуатация
и мониторинг



Наши решения:

- Автоматизированная процедура определения ПА
- Автостендирование в динамической инфраструктуре
- Fuzzing-ферма как часть сборочного конвейера
- Автоматизированное формирование артефактов сертификации

Внешнее
ПО

Идеи
и требования

Старт
разработки

Релиз
кандидат

Релиз
продукта

Вывод
в пром.

OS
Code

3d party
depend.

Бинарное
ПО

Обработка
требований

Архитектура

ПА и
МУиН

Code
review

Стат.
анализ

OSS
SCA

Чистая
сборка

Фаззинг

PenTest

Приемка
КБ

Сертификация
ФСТЭК России

Мониторинг
уязвимостей

Определение принадлежности интерфейса к ПА автоматизировано с использованием собственной методики



Примеры критериев	Методы проверки	Способ реализации атаки
Бинарный файл имеет SUID-бит или SGID-бит	Посредством statc	Атака на локальное повышение привилегий
Имеет хотя бы одну capabilities: cap_dac_override, cap_net_admin и т.д.	Посредством getcaps Посредством поиска AmbientCapabilities	
Исполняемый файл создает сокеты (кроме AF_UNIX) и [осуществляет прием датаграмм, является демоном / выполняет функции программы-сервера]	Анализ исходного кода	Сетевая атака

Динамическая инфраструктура как сервис для автостендирования

Легко и быстро



- Среда для автотестирования создается автоматически
- Время развертывания стендов менее 3х минут

Удобно



- Есть возможность взаимодействия через API
- Авторазвертывание и тестирование прикладного ПО

Безопасно



- Запуск в изолированной среде
- Возможность тестирования без сетевого взаимодействия

Прозрачно



- Автоматическая запись экрана в момент инсталляции
- Логирование на всех этапах развертывания

Разработчик

Тестирующий

DevOps

Поддержка

КБ



Автотесты

Демостенды

Мы интегрировали фаззинг в CI/CD и создали инструмент для независимого тестирования каждого пакета по требованиям ФСТЭК России

✓ Интеграция с системой сборки

Автоматическая инструментация пакетов позволяет минимизировать необходимость правки исходных кодов

✓ Изоляция и воспроизводимость

Каждая fuzzing-задача изолирована и имеет четко определенные конфигурации, что обеспечивает легкость воспроизведения результатов, ускоряет триаж и отладку при проведении тестов

✓ Интеграция с RPM-окружением

Fuzzing-обертки предоставляются через специальный репозиторий, генерируемый в процессе сборки продукта



Мы настроили наш производственный конвейер так, чтобы получать артефакты сертификации по нажатию «одной кнопки»



Перечень артефактов

SAST

- Анализ используемых языков программирования
- Отчеты по результатам триажа
- Артефакты анализа опасных лексем, секретов в коде и полит. контента

DAST

- Отчеты Unit-тестирования
- Отчеты фаззинг-тестирования
- Отчеты о покрытии
- Инструментированные сборки для DAST

Документация

- Данные о подсистемах, модулях и интерфейсах
- Описание представления реализации
- Сведения о компонентах сборочной среды
- Описание изменений в составе пакетной базы

Автостендирование

- Стенд с развернутым ПО, прошедшим процедуры безопасной установки и настройки
- Дампы сетевого трафика
- Отчеты по итогам антивирусных проверок

Другие артефакты

- Логи сборок
- SBOM-файлы
- Контрольные суммы

И в заключении, хотелось бы отметить, что:

- ✔ **Сотрудничество разработчика и эксперта КБ**
усиливает функциональность и надежность продукта
- ✔ **Достижение зрелости процессов РБПО**
упрощает прохождение сертификации на соответствие ГОСТ-56939

SberLinux и DropApp
позволяет создать безопасную
отказоустойчивую
инфраструктуру
корпоративного уровня



Спасибо за ваше внимание!

Давайте сделаем из вопросов ответы!

Баженов Андрей

Bazhenov.A.S@sbertech.ru

Щербаков Алексей

Shcherbakov.A.V@sbertech.ru