Вартан Падарян
Заведующий лабораторией ИСП РАН
vartan@ispras.ru
Андрей Белеванцев
Ведущий научный сотрудник ИСП РАН
abel@ispras.ru

27 февраля 2025 г.

Цифровая устойчивость промышленных систем

РБПО для АСУТП
Как улучшить безопасность, когда все против тебя

ИСП РАН

Ошибки в ПО – основная причина уязвимости ПАК

ИСП РАН

Принципиальное наличие уязвимостей* в ПО и аппаратуре:

функциональные, архитектурные, программного кода/микрокода.

***Размыты границы между ошибками программиста, закладками и НДВ**

Утечка информации о уязвимостях

Хакеры «для интереса»

- Взломы ради известности
- Ограничены ресурсы
- Только известные эксплойты

Хакеры «для ущерба»

- Вандализм
- Ограничены ресурсы

Преступность

- Взлом ради прибыли
- Существенные ресурсы
- Синдикаты
- Специально разработанные программы для кражи данных

Спецслужбы

- Атаки на критические информационные инфраструктуры, промышленный шпионаж
- Практически неограниченные ресурсы
- Сложное ВПО и программы для взлома
- Постоянные угрозы

Рост ресурсов и сложности атак

Разработка безопасного ПО: стандарты и требования

ИСП РАН

США: Разработка стандартов Common Criteria (NIST: National Institute of Standards and technology), 1999
Жизненный цикл безопасного ПО, Microsoft, 2004

Россия: ГОСТ Р 56939-2024

ГОСТы по процессам и инструментам (статический анализ, безопасный компилятор), введены в действие 2024

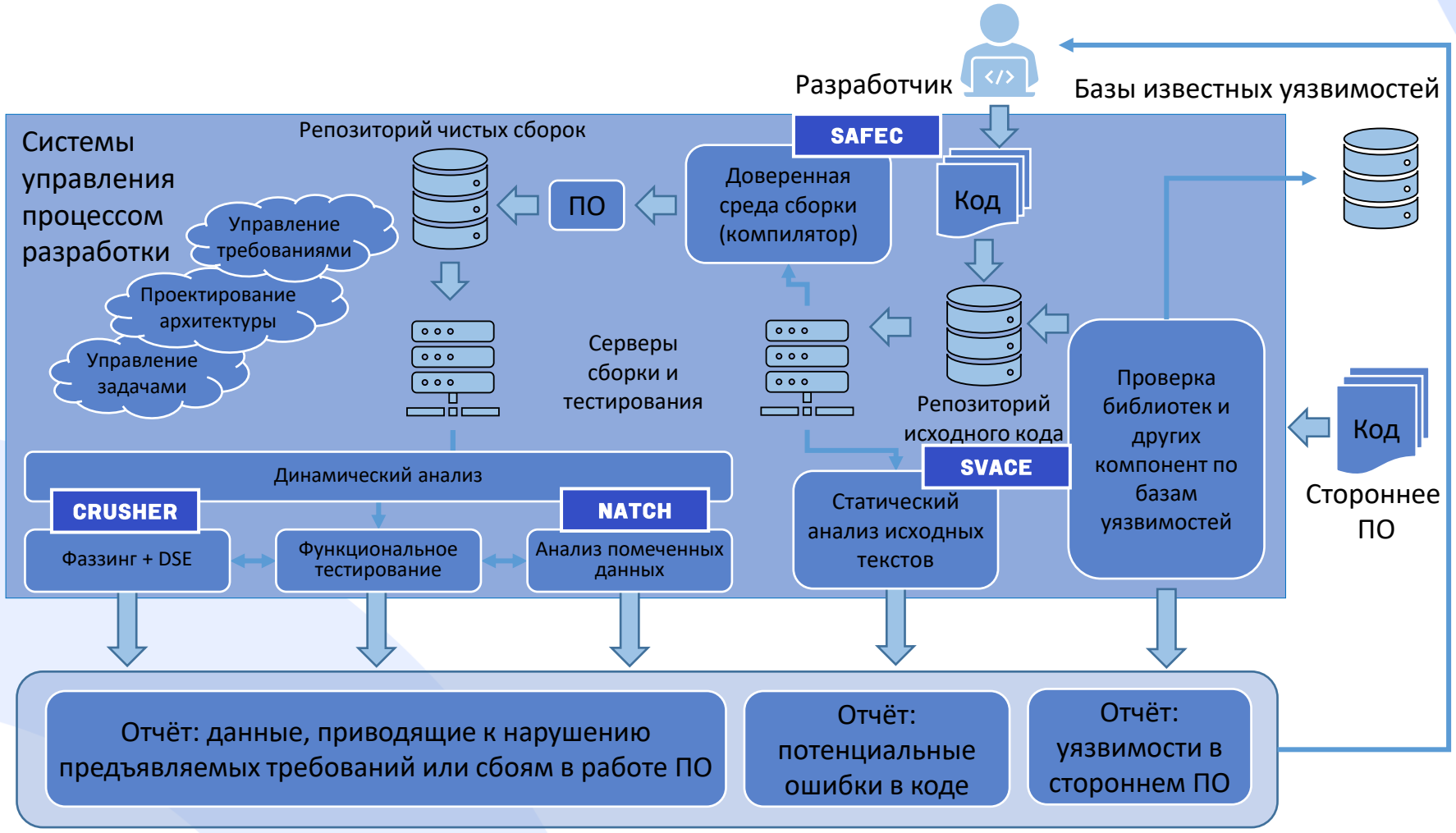
ЕС: The Cybersecurity Act (EU 881 / 2019), система сертификации ПО, сервисов и процессов

Китай: стандарты по кибербезопасности от национального комитета ТК260, 19 стандартов в 2023

- **Недостаточно использовать классические методы защиты (защита по периметру, проверка доступа, антивирусы и др.)**
- **Необходима разработка новых моделей, методов и технологий в области анализа и трансформации программ**

Найти и устранить максимальное количество уязвимостей в исполняемом коде во время **жизненного цикла разработки безопасного ПО**	**Обеспечить устойчивость** программной системы, затруднив эксплуатацию существующих уязвимостей или смягчив последствия их эксплуатации

Типовой конвейер РБПО



РБПО специализируется в каждой индустрии (авиация, автомобили, космос, госсектор) под нормативные документы и специфику ПО отрасли.

Технологический стек ИСП РАН развивается с 2002 г. на базе результатов фундаментальных научных исследований по контрактам с зарубежными и российскими компаниями.

Показательная ситуация: РБПО в электроэнергетике*

ИСП РАН

Требования регуляторов:

- Приказ №239 – внедрение, как минимум, отдельных процессов РБПО для ПО КИИ
  - проведение статического анализа исходного кода программы;
  - проведение фаззинг-тестирования программы, направленного на выявление в ней уязвимостей;
  - проведение динамического анализа кода программы
- Необходимо одновременно выполнять требования многих регуляторов: отраслевых, ФСТЭК России и ФСБ

Отраслевые особенности:

- Отсутствие отраслевых сообществ, в которых бы велась коллективная работа над анализом безопасности Open Source компонент
- Особое внимание к защите коммерческой тайны, сильные ограничения доступа к исходным кодам

Технологические сложности:

- Необходимость импортозамещать среды функционирования, средства разработки и т.д.
- Инструменты динамического анализа зачастую недоступны
- В инструментах нужна поддержка промышленных протоколов Modbus, IEC 61850 GOOSE, IEC 104 — IEC 60870–5-104, …
- Какие отечественные процессоры выбирать на перспективу

** Вопросы обсуждались на круглом столе «Безопасность, качество, электроэнергетика. Взгляд отрасли» Открытой конференции ИСП РАН 11 декабря 2024 г.*

Инструменты РБПО: технологически независимый стек

ИСП РАН

Безопасный компилятор

SAFEC

kaspersky, Postgres PROFESSIONAL, ASTRA LINUX

Аналоги: компоненты компиляторов GCC/Clang, исследовательские компиляторы и рандомизаторы (CompCert C, kcc, Selfrando, Oxymoron, Shuffler)

Инструмент статического анализа

SVACE

SAMSUNG, Postgres PROFESSIONAL, HUAWEI, ..., kaspersky

Аналоги: Klocwork (Perforce, США), Coverity (Synopsys, США), Fortify (MicroFocus (прежде HP), США).
Открытые инструменты: Clang Static Analyzer, SpotBugs

Комплекс динамического анализа

CRUSHER

Postgres PROFESSIONAL, КОД безопасности, kaspersky, ASTRA LINUX

Аналоги: Peach Fuzzer (США, Peach Tech), Synopsys Defensics (Synopsys, США), MAYHEM (ForAllSecure, США).
Открытые инструменты: angr, American Fuzzy Lop, Driller (США)

Инструмент определения поверхности атаки

NATCH

kaspersky

Аналоги: инфраструктуры Panda, Decaf (без интроспекции), Panorama (исследовательский инструмент для Windows)

Поддержка стандартов кодирования MISRA / AUTOSAR

ИСП РАН

В ИСП РАН разработаны **статические детекторы** для программ на языках C/C++, предназначенные для проверки соответствия стандартам безопасного кодирования

MISRA C 2012, MISRA C++ 2008, AUTOSAR C++14.

Основа реализации:

- компиляторная инфраструктура LLVM
- механизм сопоставления синтаксических конструкций на уровне дерева AST программы (Clang-Tidy)
- механизм символьной интерпретации Clang Static Analyzer (CSA)

Совокупное покрытие детекторами стандартов безопасного кодирования:

- MISRA C 2012 (C99) — **95%**
- MISRA C++ 2008 (C++98) — **более 50%**
- AUTOSAR C++14 — **более 50%**

Решение поставляется в виде отдельного исполняемого файла для ОС на базе Linux **(в перспективе может использоваться как компонент статического анализатора SVACE).**

MISRA: сравнение с открытыми анализаторами

ИСП РАН

Проект	Срабатываний			Точность	Время анализа
	Всего	Рассмотренных	Ложных (из рассмотренных)		
zlib	8388	1034	0	100%	33 сек.
openjpeg	20371	1433	9	99.4%	5 мин.
OpenSSL	275528	2229	7	99.7%	49 мин. 29 сек.
coreJSON*	7	7	0	100%	10 сек.

Оценка реализованных детекторов для правил MISRA C 2012 на проектах с открытым исходным кодом:

Сравнение с открытым анализатором **Cppcheck** на проекте **zlib** (для отличающихся правил)

Анализатор	Clang-Tidy/CSA			Cppcheck		
	TP	FP	FN	TP	FP	FN
7.2	1172	0	0	61	0	1111
10.3	98	0	0	20	0	78
15.7	15	0	0	11	0	4
16.3	38	0	0	13	0	25
12.3	26	0	0	6	24	20

MISRA: тестирование на coreJSON

coreJSON — открытая библиотека для работы с данными в формате JSON, ***соответствующая*** *стандарту кодирования MISRA C 2012.*

Соответствие стандарту MISRA разработчики coreJSON проверяют **коммерческим статическим анализатором Coverity.**

- С помощью разработанных детекторов для проверки правил MISRA C 2012 были найдены ***несоответствия*** кода правилам **10.3, 13.5 и 20.12**

- О найденных несоответствиях кода стандарту MISRA C 2012 было сообщено разработчикам ***coreJSON*** в результате чего были внесены соответствующие исправления в код*

* https://github.com/FreeRTOS/coreJSON/pull/148
https://github.com/FreeRTOS/coreJSON/pull/157

Анализатор Svace в применении к АСУТП

ИСП РАН

- Широкая поддержка процессорных архитектур
  - Intel x86/x86-64, ARM/ARM64, MIPS/MIPS64, Power PC/Power PC 64
  - RISC-V 32/64, SPARC/SPARC64, Hexagon
  - AEON, TriCore, HIDSP, OpenRISC, **Эльбрус**
- Широкая поддержка C-компиляторов, в том числе для встраиваемых платформ
  - GCC/Clang/MSVC/Intel
  - RealView/ARM (ARMCC), WindRiver Diab, Keil CA51
  - NEC/Renesas CA850, Renesas M16C/R8C, CC78K0(R), IAR C/C++
  - Panasonic MN10300, Toshiba TLCS-870/T900
  - Texas Instruments TMS320C6*, Digital Mars, Green Hills
  - TriCore, CEVA, Samsung CalmSHINE16, Cadence Tensilica Xtensa
  - CodeWarrior for StarCore, Freescale CodeWarrior, компилятор Эльбруса (lcc)

Анализатор Svace в применении к АСУТП – II

ИСП РАН

- Быстрая доработка под специфические компиляторы и диалекты C (1-3 месяца)
- Критические ошибки C/C++ по ГОСТ 71207-2024
  - разыменование нуля
  - переполнение буфера
  - утечки памяти и ресурсов
  - неинициализированные переменные, недостижимый код
  - анализ указателей, вызовы по указателю (девиртуализация)
- Пользовательские спецификации библиотечных функций,
  в том числе источников и стоков для анализа чувствительных данных (taint)
  - позволяет работать с нестандартными / обрезанными библиотеками
  - позволяет отразить специализированные потоки чувствительных данных

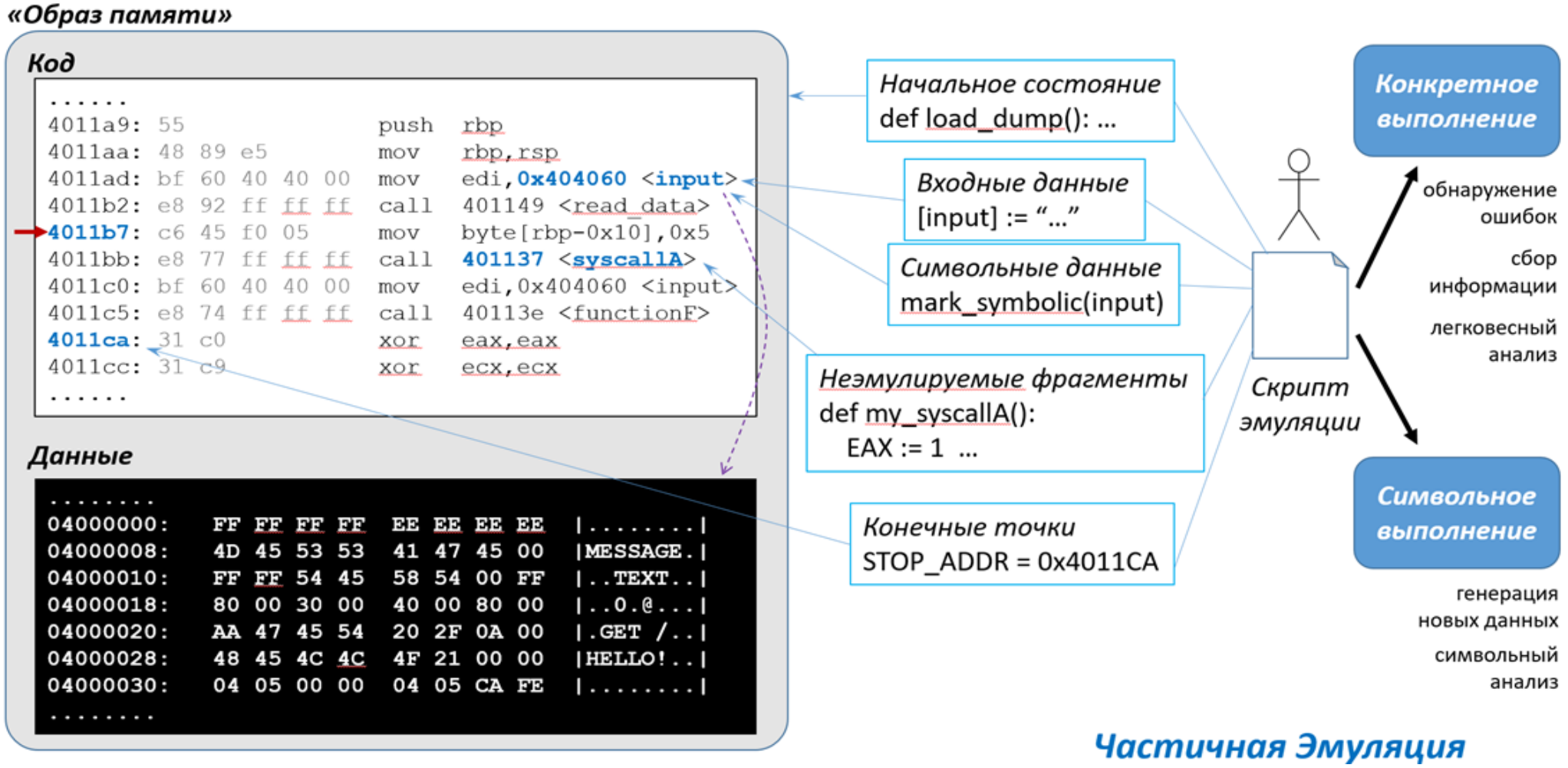
Комплексная технология фаззинга Crusher

ИСП РАН

- Гибкость – тонкая настройка под конкретные задачи и требования
- Масштабируемость — вплоть до нескольких тысяч ядер
- Расширяемость — пользовательские мутации, плагины, интеграция открытых инструментов и т.д.
- Интеграция с инструментом Svace и другими технологиями ИСП РАН
- Фаззинг GUI приложений
- Гибридный фаззинг с поддержкой SymCC, Angr, Sydr
- Фаззинг ядра и драйверов ОС
- Фаззинг произвольного кода при помощи частичной эмуляции
- Поддержка языков C/C++, Java, Python, C#
- Широкая поддержка процессорных архитектур – x86, ARM, MIPS
- Поддерживает долгосрочный, распределенный фаззинг

Анализ произвольного кода комплексом Crusher

ИСП РАН



Унифицированная среда разработки ПО

ИСП РАН

Регулятор

Разработчики СЗИ

Квалификационные тесты для оценки ключевых характеристик инструментов

Инструменты и методические рекомендации по внедрению / ГОСТы

Технологии / инструменты анализа кода ПО

IT-сервисы, CI/CD: Jenkins, Gitea, IDE Eclipse Theia

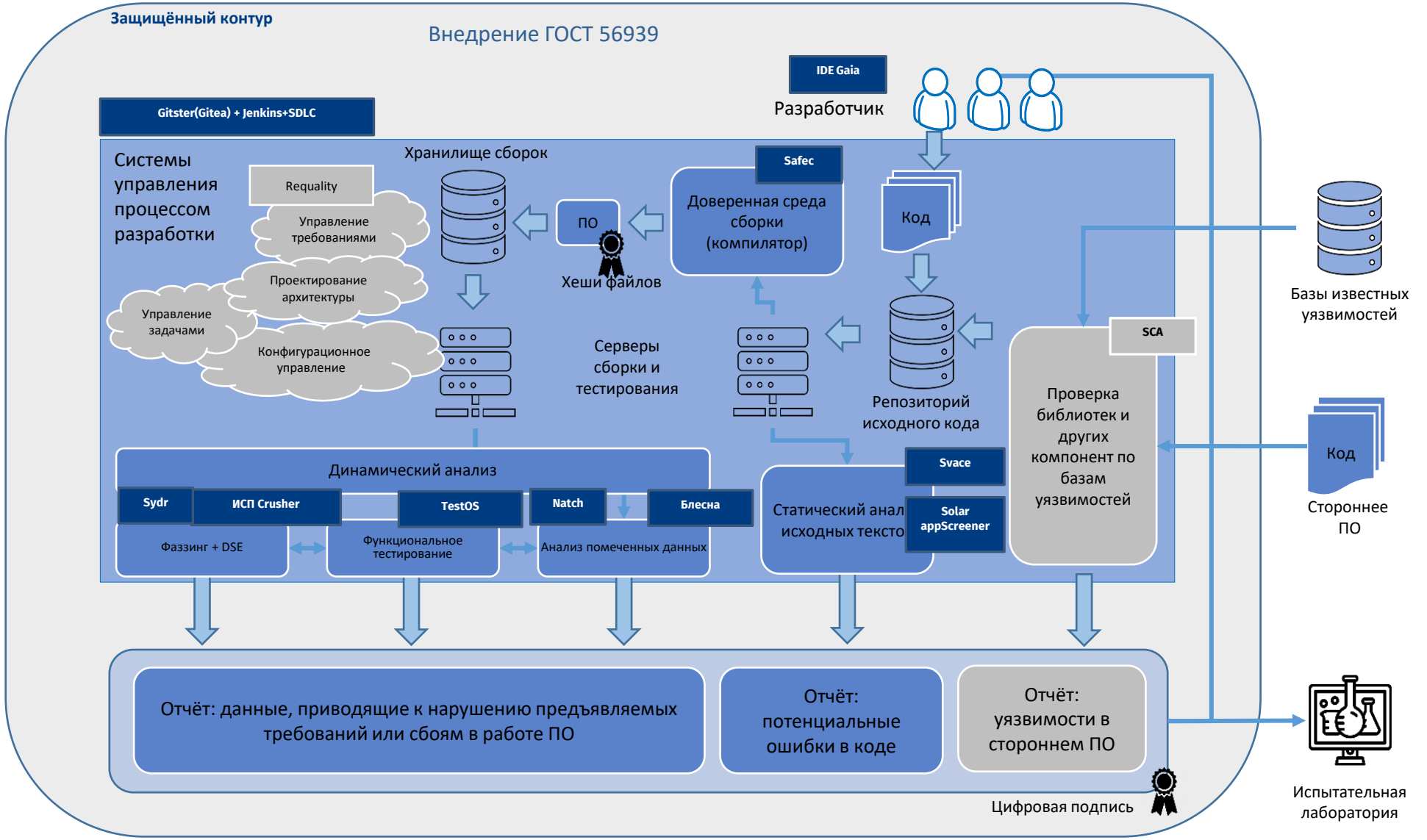
Инфраструктура, аттестованная АС: серверы и СЗИ

Унифицированная среда: результаты работы

ИСП РАН

- Комплекс инструментов, развернутых в виртуальных машинах
  - IDE Gaia, Gitea, Jenkins, ISP SDLC, Svace, appScreener, Crusher (SyDr), Natch, Блесна, SAFEC, TestOS, Qemu (ARM, RISC-V)
- Поддержка инструментами отечественных процессоров Байкал и RISC-V
- Тесты: стат. анализ, фаззинг, утечки данных
- Инфраструктура для развертывания и апробации инструментов
  - Основные сценарии применения: «чистый» репозиторий, подготовка материалов для сертификации ПО
- Учебными научным заведениям, гос. ведомствам и корпорациям предлагается предоставлять УНС бесплатно
  - Лицензии включенного в ее состав проприетарного ПО оплачиваются отдельно
  - При поставке лицензий может быть специальный тариф за комплект

Унифицированная среда: технологии



Развитие унифицированной среды и АСУТП

ИСП РАН

- Поддержка композиционного анализа и защиты цепочек поставки
- Поддержка стандартов по безопасному кодированию MISRA и AUTOSAR набором инструментов УС
- Разработка частных методик анализа кода:
  - Программных модулей на языках C/C++, Java, Go, Python, C#, JavaScript, Kotlin
  - Типовых доверенных ПАК, применяемых на объектах критической инфраструктуры
- Создание облачной версии УС
- Создание в составе УС сервиса оповещения пользователей о выявленных уязвимостях ПО по базам данных уязвимостей
- Расширение возможностей инструментов анализа на базе технологий ИИ

Спасибо!

ИСП РАН