ИСП РАН

Вопросы разработки отечественного безопасного ПО

ПАДАРЯН В.А.
Ведущий научный сотрудник ИСП РАН
vartan@ispras.ru

23 марта 2022

Уязвимости – причина компьютерных атак

Принципиальное наличие **уязвимостей*** в ПО и аппаратуре

- функциональные
- архитектурные
- программного кода/микрокода

***Границы между ошибками программиста, закладками, НДВ размыты**

Утечка информации о уязвимостях

Спецслужбы

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

Преступность

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

Хакеры «для ущерба»

- Вандализм
- Ограничены ресурсы

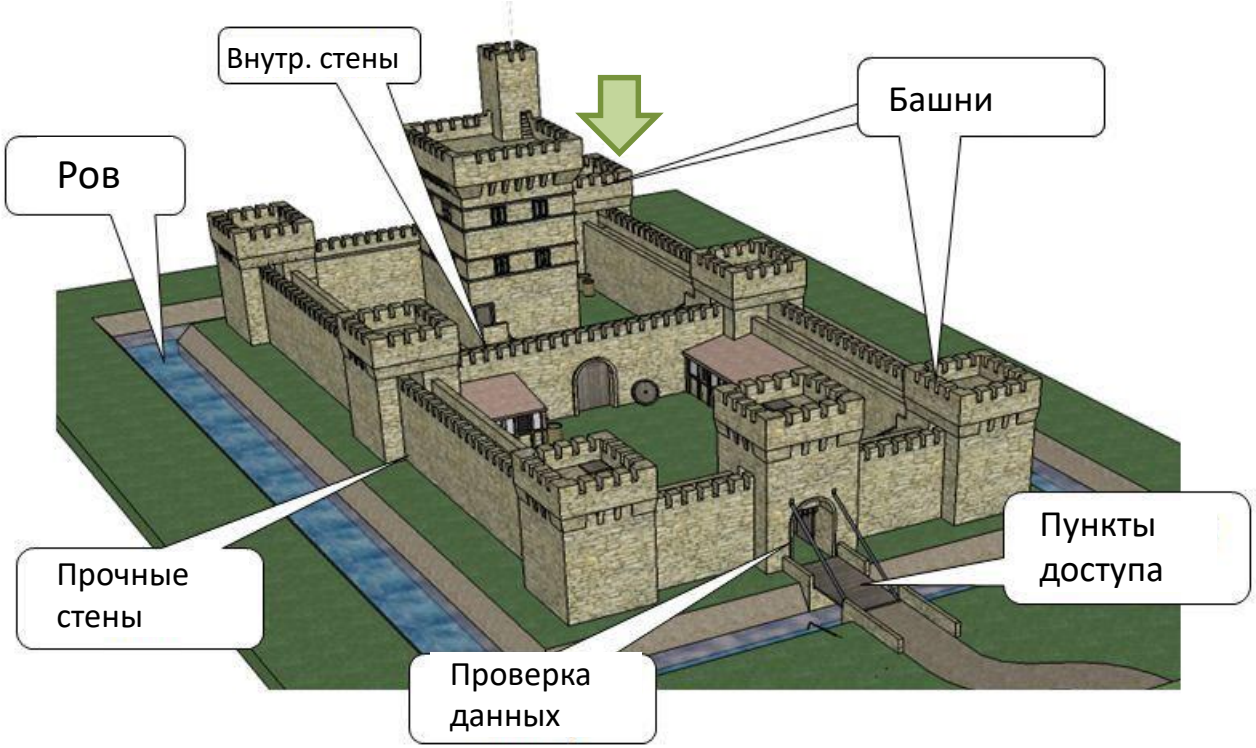
Хакеры «для интереса»

- Взломы ради известности
- Ограничены ресурсы
- Только известные эксплойты

Рост ресурсов и сложности атак

Классические методы защиты не являются основными

- Защита по периметру
- Организационные мероприятия (проверка доступа)
- Антивирусы и др.



Вызовы

- Эскалация размеров. ПО – большие данные
  (размер Astra Linux – 150 млн. строк кода,
  размер Debian – более миллиарда строк)
- Сложность среды разработки и сборки
  (например, компилятор может добавить уязвимость,
  которой нет в исходном коде)
- Облачно-мобильные платформы, ИИ, «Интернет вещей»
  (отсутствие локально изолированных систем)…
- Сложность цепочек поставок
  (большая часть кода программных систем – привлекаемое и
  заимствованное ПО)
- ***Доступность глобальной IT-инфаструктуры?***

Техдирский Клуб @ctorecordschat | Случай малвари, шифровальщиков и прочего

Опасно | Пропаганда | Досилки и прочие атаки | Одумались

	A	B	C	D	E
1	Отметка времени	Тип проблемы	Название продукта	Ссылка на пуш в репе	Комментарий
2	**https://t.me/ctorecordschat | https://fb.com/ctorecords/** **СПИСОК МАЛВАРИ, ШИФРОВАЛЬЩИКОВ и прочих военных действий в OpenSource продуктах** Коллеги! В связи с появившимися случаями встраивания политических лозунгов, а теперь и малвари и шифровальщиков в опен-сорс продукты, который теперь реально несут риски гибели в используемых продуктах, начинаем вести список опасных для использования продуктов. Чтобы пополнить список если вам что-то известно, оставляйте сообщение в специальной анкете https://forms.gle/3fPraenYPQ7skrPCA				
3	19.03.2022 21:26:10	Малварь, шифровальщик или иные фатальные угрозы	https://github.com/brzuchal/stop-the-war	https://github.com/evermade/support-ukraine-banner/commit/38d2bd5cd898452576de53055f48aa82d235b63e	PHP_CodeSniffer rule adding non-collapsable comment calling out to Stop The War
4	19.03.2022 8:04:27	Блокировка доступа по IP	Qt	https://download.qt.io/	Закрыли доступ ко всем внутренним ресурсам с 7.03.2022 Не работают ни чат, ни сервер загрузок, ни багтреккер. Фины из Qt Company ничего официально не заявляли - и, будем надеяться, что не будут ничего пихать в сорцы.
5	18.03.2022 18:57:37	Малварь, шифровальщик или иные фатальные угрозы	es5-ext	https://github.com/medikoo/es5-ext/commit/3beace4b3d00f02da61b72dd328f90cf069d46de	Малварь
6	18.03.2022 18:23:29	Малварь, шифровальщик или иные фатальные угрозы	Kinoko	https://f-droid.org/en/packages/com.ero.kinoko/	После последнего обновления напрочь убил телефон
7	17.03.2022 15:11:51	Малварь, шифровальщик или иные фатальные угрозы	Tasmota	https://github.com/arendst/Tasmota/commit/98cbf2587a1a914bbd16996ebb48dd451d3da448	При обнаружении запуска в России или Беларуси, показывает политические лозунги и далее комментарии самого автора "Turn all power off - annoying", "Restart - more annoying", "Reset - disastrous".
8	17.03.2022 16:27:20	Отзыв лицензии	Wowza	https://www.wowza.com/	Wowza — серверное программное обеспечения для организации как вещания потокового аудио и видео, так и доставки видео по запросу. Отозвало лицензии, в связи с чем вещание прекратилось. Ищите замену
9	17.03.2022 17:21:42	Малварь, шифровальщик или иные фатальные угрозы	uashield	https://github.com/opengs/uashield	Читайте описание
10	17.03.2022 21:48:25	Малварь, шифровальщик или иные фатальные угрозы	К существующему пункту про node-ipc	https://gist.github.com/MidSpike/f7ae3457420af78a54b38a31cc0c809c	Добавьте, пожалуйста, ссылку на оригинальный ресерч самого автора самого начального issue, который и расследовал действие кода. Сейчас его issue потёр автор node-ipc, поэтому исследователь переехал в gist: https://gist.github.com/MidSpike/f7ae3457420af78a54b38a31cc0c809c По ссылке есть таймлайн события, расследования и меры минимизации ущерба.
11	17.03.2022 21:56:52	Малварь, шифровальщик или иные фатальные угрозы	https://github.com/Sateviss	https://github.com/Sateviss/sateviss.github.io/blob/master/ddos.html	Гитхаб игнорит абузы с ру айпи
12	18.03.2022 10:46:13	Малварь, шифровальщик или иные фатальные угрозы	https://github.com/Yaffle/EventSource	https://github.com/Yaffle/EventSource/commit/de137927e13d8afac153d2485152ccec48948a7a	https://github.com/Yaffle/EventSource/issues/199
13	18.03.2022 11:52:48	Бан аккаунтов	Oracle cloud infrastructure	https://www.oracle.com/corporate/conflict-in-ukraine/russia/	Не дает зайти в аккаунт, виртуальные машины пока еще работают

Зарубежный опыт: разработка безопасного ПО

- Госстандарты: Common Criteria (развитие с 1999), документы МО США (Program Protection Plan для всех этапов разработки, с 2000х)
- Технологии, инструменты анализа выбираются сертификационными лабораториями → разработчики ПО вынуждены пользоваться инструментами требуемого уровня
- Заложено непрерывное обновление стандартов по мере развития технологий (текущая версия CC – 3.1)
- Принятие нормативных документов привело к взрывному росту рынка технологий анализа
  https://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html

Технологии и процессы разработки безопасного ПО

Без внедрения ГОСТ 56939-2016, затруднительно пройти сертификацию по действующей редакции Методики



Bug Bounty Program: жизнь после SDL

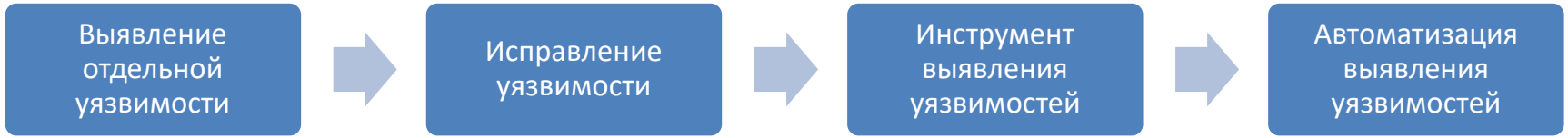
- Bug Bounty Program (BBP) – программа поощрения сторонних специалистов за найденные ошибки в ПО, в первую очередь – эксплуатируемых уязвимостей

Тип	Примеры	Особенности
Системообразующие BBP	Microsoft, Facebook, Google, US DoD	Проводятся непосредственно крупнейшими разработчиками ПО, которые устанавливают порядок выкупа информации об ошибках
Платформы BBP	BugCrowd, HackerOne, TippingPoint	Легальная общая площадка для BBP различных организаций Размер вознаграждения определяется самой площадкой
Частные посреднические BBP	Zerodium, Sonosoft	Выкупает уязвимости с целью перепродажи Размер вознаграждения больше, чем в легальных программах

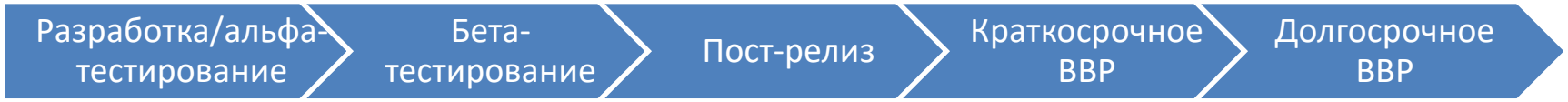
- Размер вознаграждения сильно варьируется
  - Скидочный купон на 12,50$ от Yahoo!
  - До 2 500 000$ за полную цепочку реализации уязвимости в Android смартфоне (с сохранением присутствия) от Zerodium
- BBP – mainstream. Почему?
  - Кратный ROI при правильной организации BBP, снижение репутационных издержек

Bug Bounty Program
Легко сказать, сложно сделать

- Как определить область действия BBP?
  - Отдельный программный продукт / текущая версия продукта
  - Вся линейка версий программного продукта
  - Сторонние библиотеки, ПО поглощенных компаний
- Как организовать долгосрочное использование результатов BBP, т.е. «обогатить» SDL новыми технологиями и практиками?



- На каком этапе жизненного цикла подключать продукт к BBP?

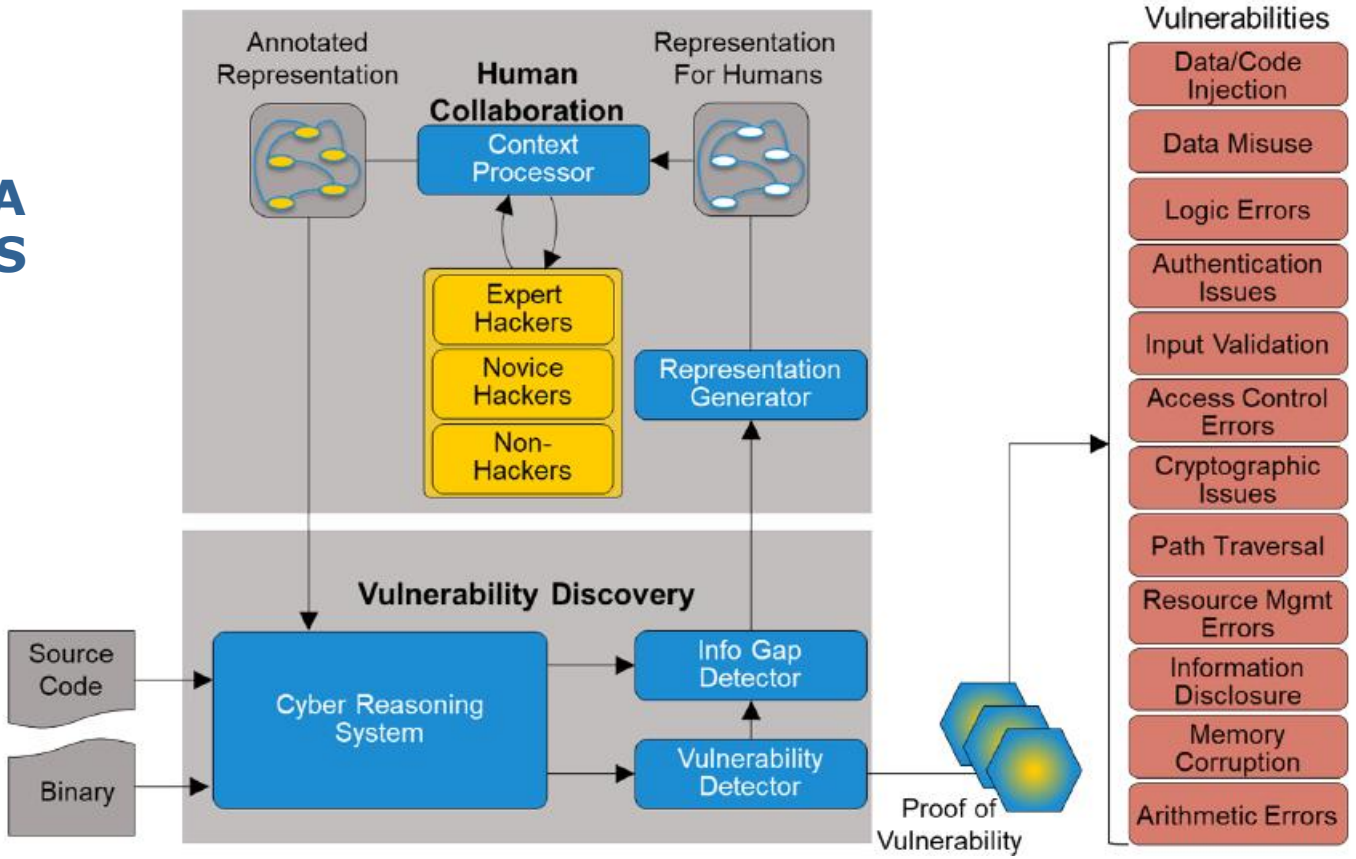


Какую информацию о продукте (документация, система сборки, исходные коды, система тестирования, тесты, результаты тестирования, …) допустимо раскрыть сторонним (или приглашенным) специалистам?
Больше информации $\Rightarrow$ лучше результат + больше рисков утечки информации

- Анализ потока обращений: отсеивание False Positive, соотнесение сбоя с кодом, нахождение уникальных ошибок, оценка критичности ошибки, …

Зарубежный опыт: автоматизированный поиск критических дефектов

DARPA CHESS



- **DARPA Cyber Grand Challenge** (2016)
  Соревнование по автономному поиску и исправлению критических дефектов в бинарном коде, победитель: Mayhem – Carnegie-Mellon-University

- Проект **DARPA CHESS** (2018, объявлен конкурс)
  Масштабируемый поиск критических дефектов за счет синергии инструментов анализа и экспертных знаний

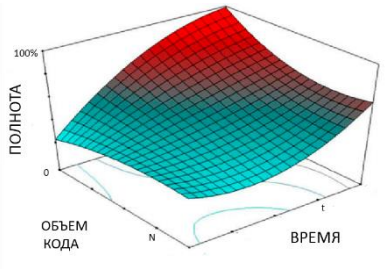
Необходимый этап SDL: непрерывный поиск дефектов в ПО после сдачи в эксплуатацию

- **Нет ограничения по времени**
- **Требует больших ресурсов: вычислительных мощностей и экспертизы специалистов**



Подтверждение доверия ПО





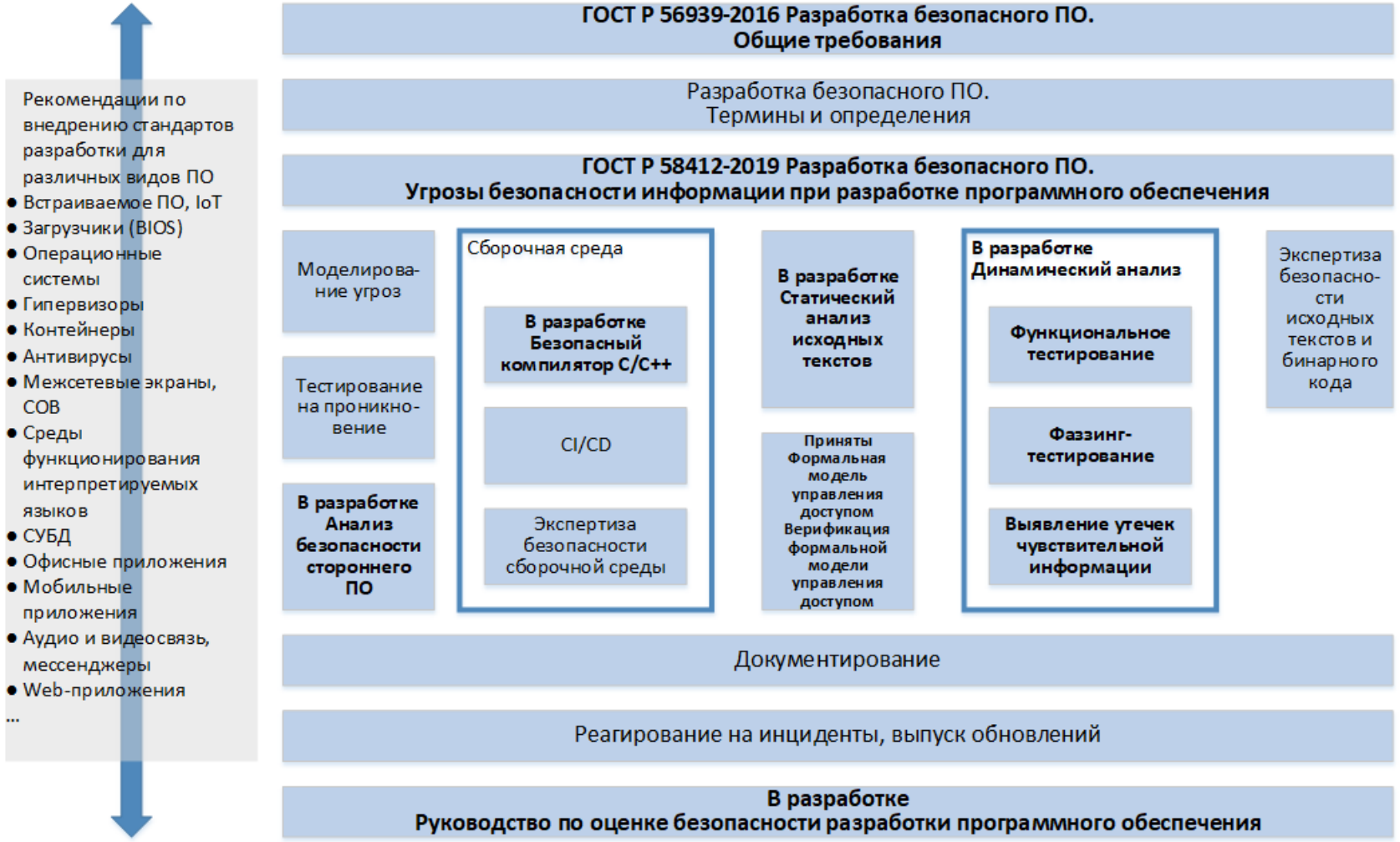
- Непрерывный анализ безопасности ПО, на выделенных вычислительных мощностях
- Совместное участие разработчиков, регуляторов, науки
- Возможные пилотные проекты: ядро Linux, СУБД, runtime и VM Java, …

Развитие нормативной базы в 2022 г.

- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Руководство по проведению статического анализа. Общие требования**»
- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Руководство по проведению динамического анализа. Общие требования**»
- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Доверенный компилятор языков Си/Си++. Общие требования**»
- ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Управление безопасностью программного обеспечения при использовании заимствованных и привлекаемых компонентов**»

- ГОСТ Р «Информационная технология. Методология разработки доверенных систем. **Конструктивная информационная безопасность. Общие положения**»;
- ГОСТ Р «Информационная безопасность. Методология разработки доверенных систем. **Конструктивная информационная безопасность. Шаблоны проектирования**»;
- ГОСТ Р «Информационная технология. Методология разработки доверенных систем. **Конструктивная информационная безопасность. Методология разработки**».

Карта стандартизации



Технологии ИСП РАН жизненного цикла разработки безопасного ПО

Разра-ботка

- Svace и Binside: статический анализ исходного и бинарного кода
- Поиск клонов кода: ошибки, нарушения лицензии
- Дедуктивный анализ моделей безопасности

Тестиро-вание

- Контролируемое выполнение на базе эмулятора QEMU
- Natch: автоматизированное построение поверхности атаки
- Crusher (ISP Fuzzer и SyDr): фаззинг и динамическое символьное исполнение, расширенное тестирование
- Блесна: выявление утечек чувствительных данных

Выпуск

- Доверенный компилятор
  - Диверсификация кода
  - ИСП–Обфускатор: защита кода от анализа
  - Доработка системных защитных механизмов (ASLR …)

Реагиро вание

- Непрерывный поиск дефектов в выпущенном ПО
- Анализ аварийных завершений для оценки критичности программных дефектов

Деятельность ИСП РАН

- **Исследования**
  - Ведущая в РФ научная школа в области системного программирования
- **Разработка**
  - НИОКР в области кибербезопасности, в том числе в рамках ГОЗ
  - Инструменты для разработки безопасного ПО
  - Большие данные и анализ социальных сетей
  - …
- **Обучение**
  - Кафедры в ведущих университетах
  - Обучение для сотрудников Испытательных Лабораторий (ФСТЭК России)
- **Сотрудничество**
  - Соглашение о сотрудничестве между ФСТЭК России и ИСП РАН
  - ИСП РАН возглавляет Подкомитет 4 «Разработка безопасного программного обеспечения» в ТК 362 «Защита информации»
  - telegram-площадка https://t.me/sdl_community
  - Группы по интересам: статический анализ, фаззинг, доверенный репозиторий и не только …
  - Создание центров компетенции

Схема
организации работ по исследованиям безопасности операционных систем на базе ядра Linux

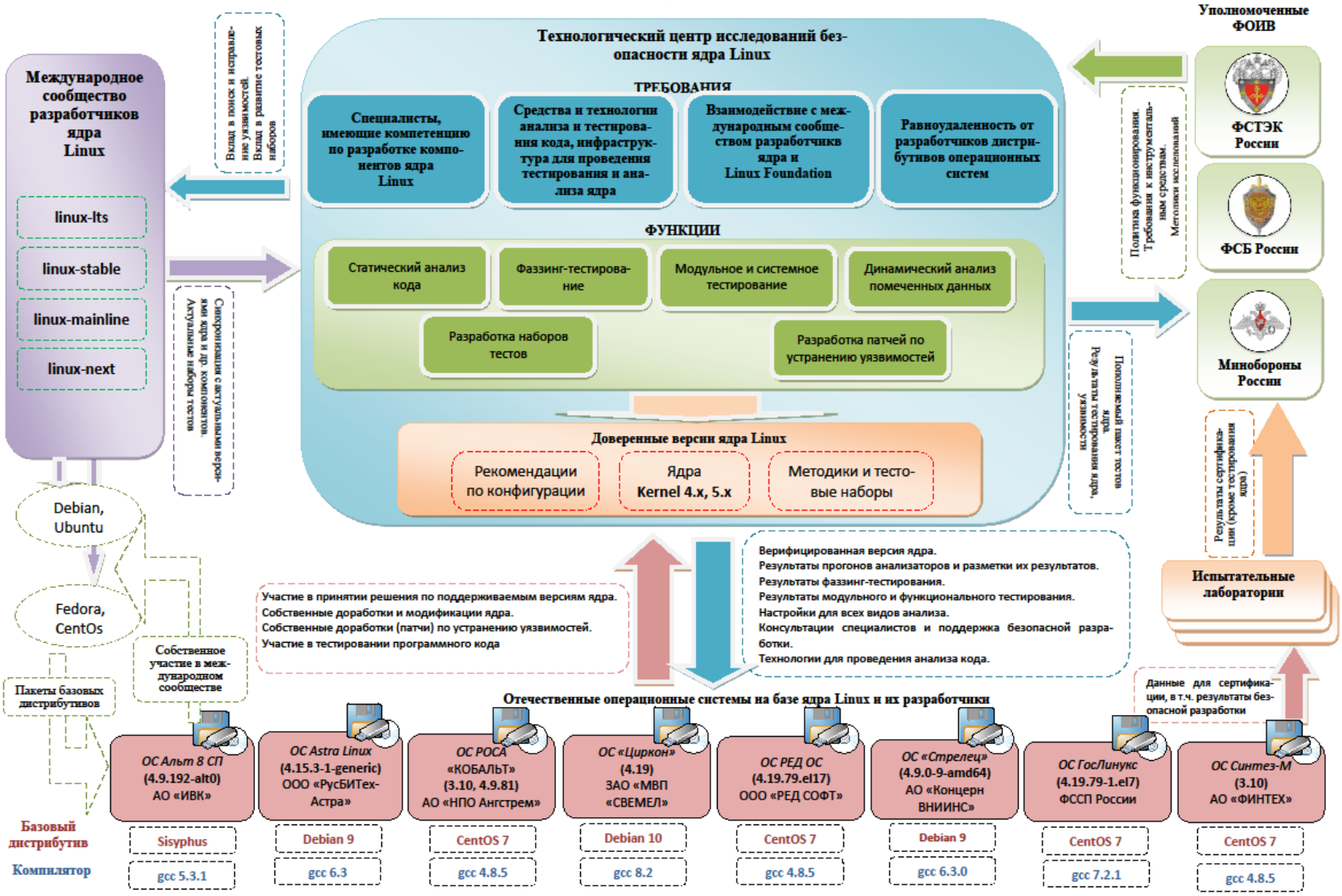
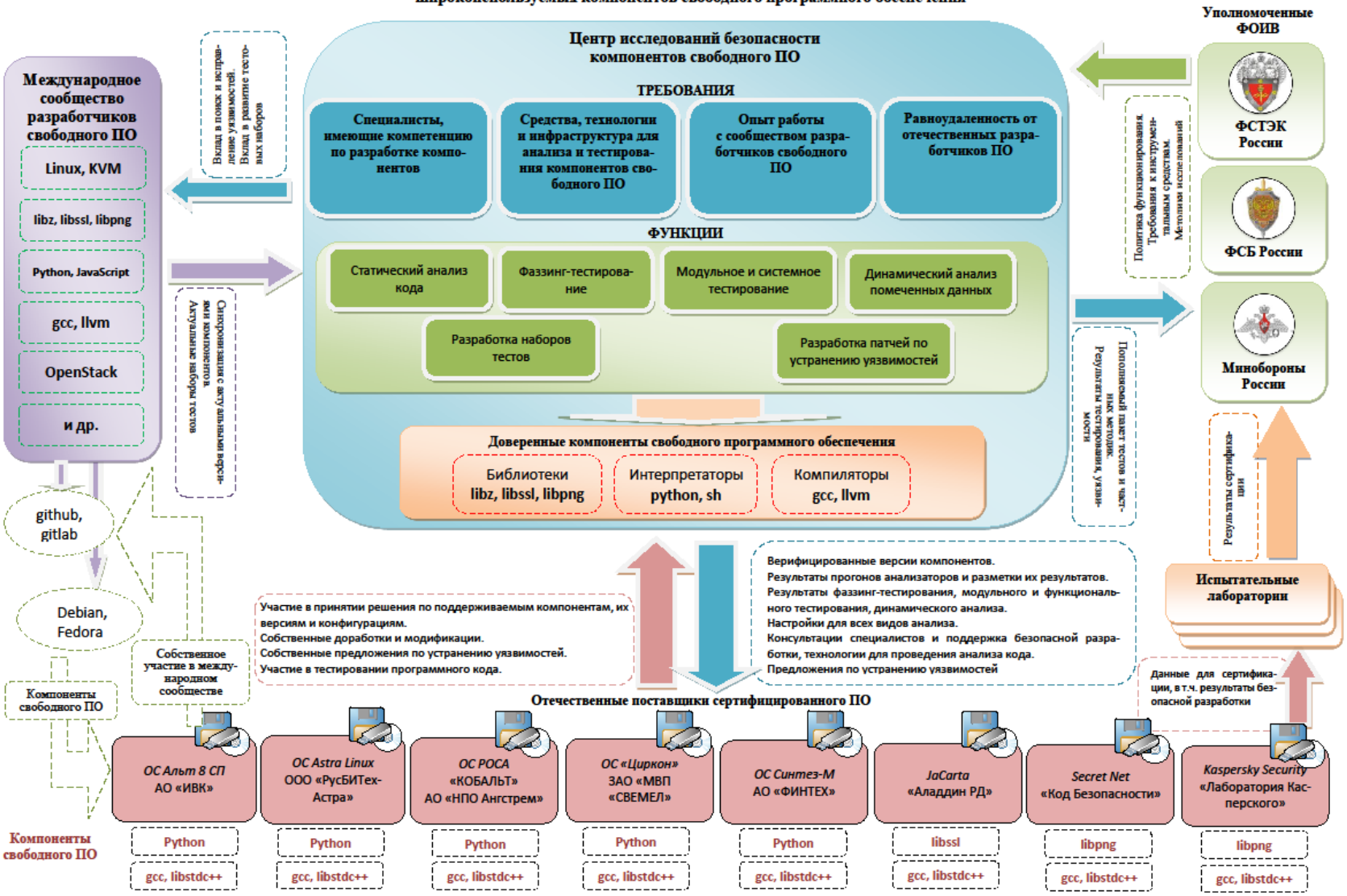


Схема
организации работ по исследованиям безопасности
широкоиспользуемых компонентов свободного программного обеспечения



Что дальше?

- Непрерывный анализ критических компонент
  Специалисты + Инструменты анализа + Методики + ЦОД = Доверие
  - Распространенные библиотеки
  - Средства виртуализации и контейнеризации
  - Популярные языки (с динамической типизацией)
    - Node.js, Python, Java, …
  - Совместная работа заинтересованных IT-компаний
- Унифицированная среда разработки – инфраструктура для отечественных разработчиков безопасного ПО
  - Система интегрированной разработки, CI/CD
  - Безопасный компилятор C/C++
  - Кросс-платформенная разработка (x86-64/ARM)
  - Разработка и выполнение модульных тестов
  - Статический анализ: C/C++, Java + один интерпретируемый язык
  - Фаззинг-тестирование
  - Выявление утечек информации по памяти на основе полносистемного динамического анализа помеченных данных.
  - Исследование и портирование инструментов на архитектуры RISC-V, Байкал, Эльбрус
- Создание ведомственных центров анализа программного обеспечения

ИСП РАН

Спасибо за внимание
Вопросы

Вопросы разработки отечественного безопасного ПО

ПАДАРЯН В.А.
vartan@ispras.ru

Разработка безопасного ПО в РФ

- Самостоятельное внедрение практик SDL ведущими компаниями-разработчиками
- В декабре 2018 в составе ТК 362 Защита Информации создан подкомитет «Разработка безопасного программного обеспечения»
  - Процессы: организация жизненного цикла ПО
  - Технологии: уровень инструментов анализа кода
- Соглашение о сотрудничестве между ФСТЭК России и ИСП РАН, центр компетенций
  - telegram-площадка https://t.me/sdl_community
  - Группы по интересам: статический анализ, фаззинг, доверенный репозиторий и не только

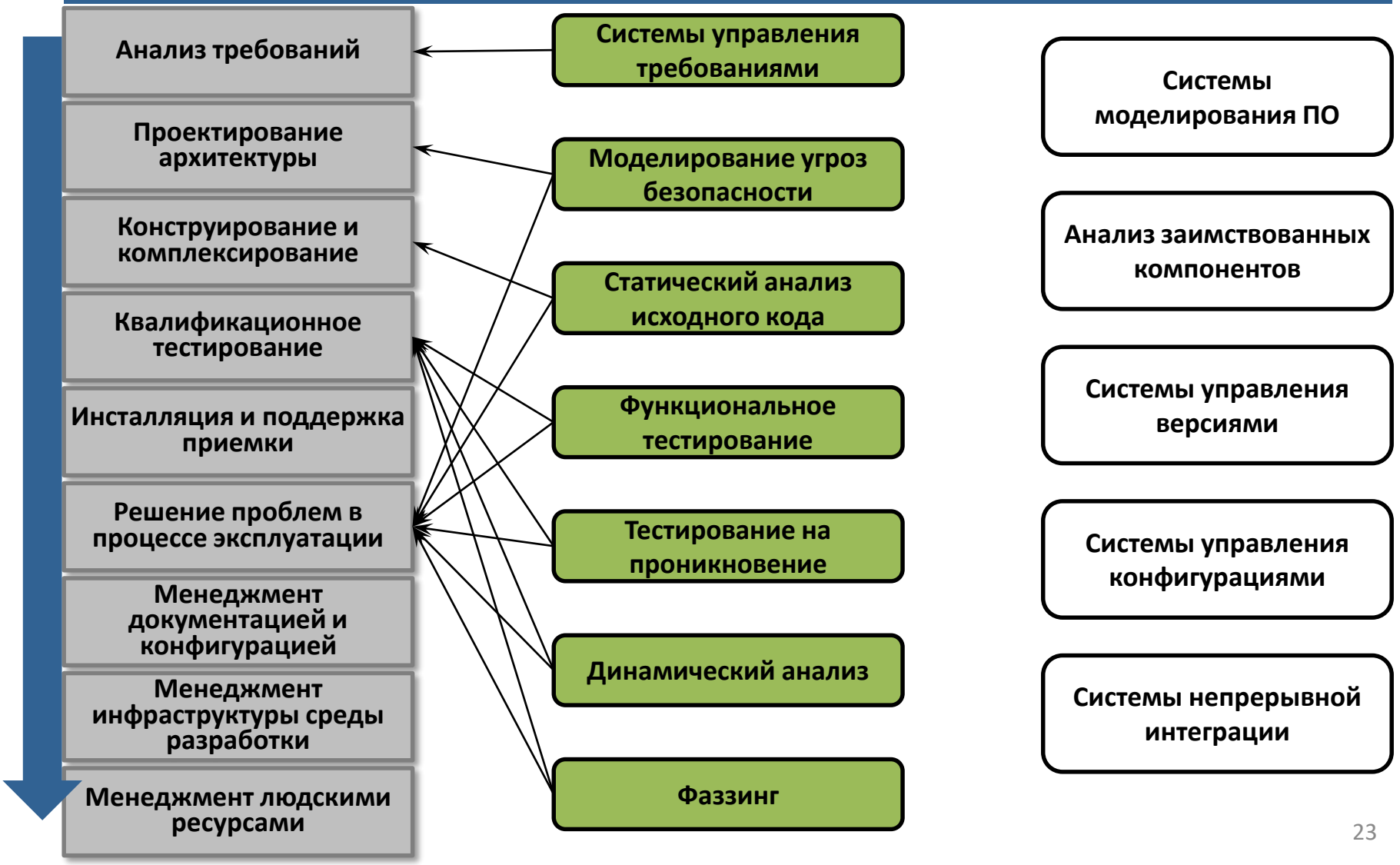
Документы ФСТЭК России

- **ГОСТ Р 56939-2016** «Защита информации. Разработка безопасного программного обеспечения. Общие требования» Утверждён и введен в действие **01.06.2016**

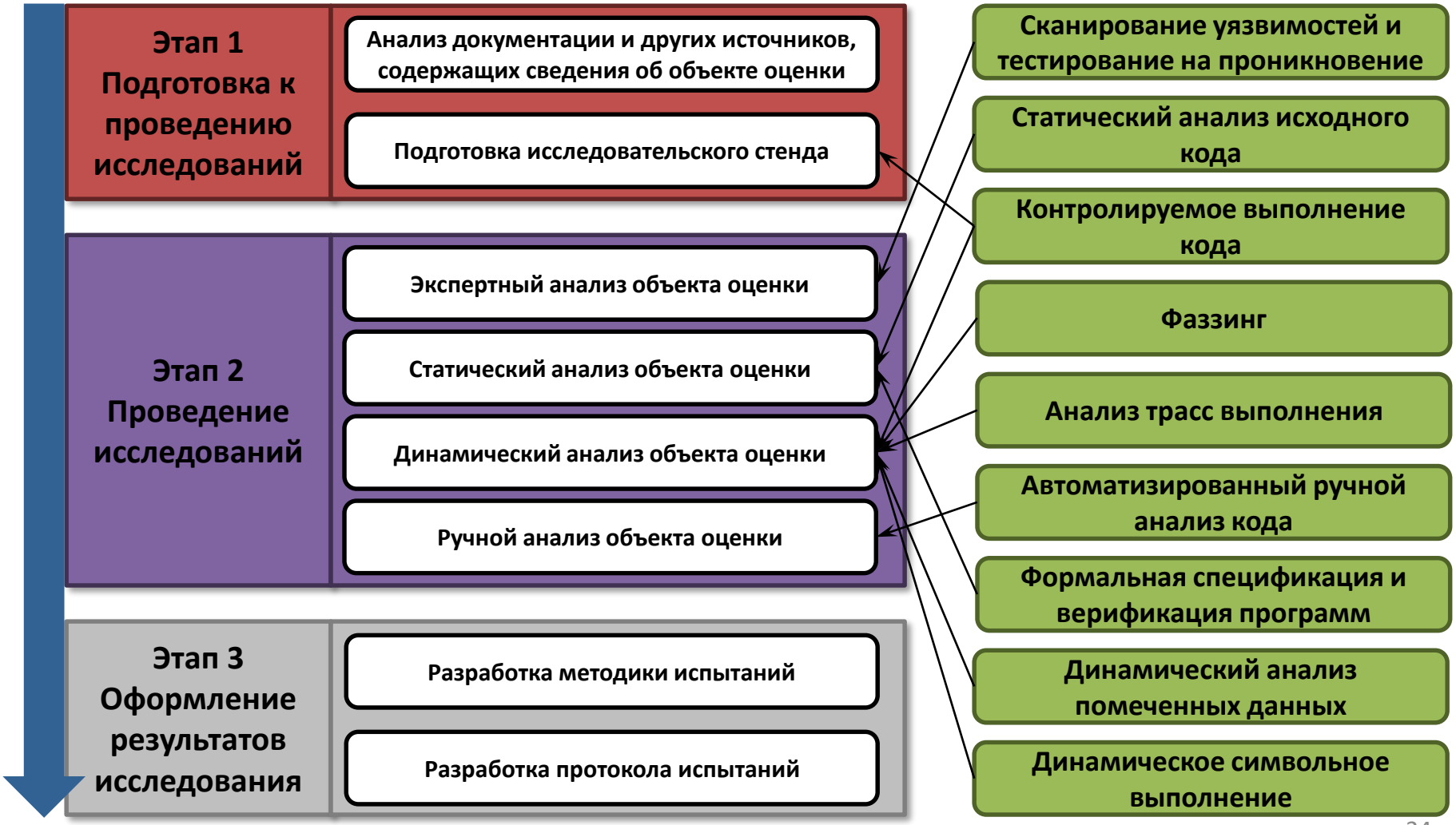
- «**Методика выявления уязвимостей и недекларированных возможностей в программном обеспечении**» утверждена ФСТЭК России 11 февраля 2019 г.
  - Применяется при проведении сертификационных испытаний **с 1 июня 2019 г.**

- ***Задано направление на применение в системе сертификации автоматизированных и автоматических технологий анализа безопасности программного кода***

ГОСТ Р 56939-2016



Методика выявления уязвимостей и НДВ★



★Первая редакция Методики, 2019 г.

Доверенная компиляция



- Уязвимости в программе могут появляться не только из-за ошибок в ее коде, но и в результате оптимизаций, выполняемых компилятором
- В набор средств для обеспечения безопасности ПО должны быть также включены средства разработки (компилятор)

Доверенный компилятор

- Требования доверия
  - Безопасные оптимизации: нельзя опираться на предположения об априорной «корректности» программы
    - Например, значения указателей различающихся типов могут совпадать, могут происходить целочисленные переполнения, …
  - Принудительная инициализация всех переменных
  - Предупреждение (ошибка компиляции) при обнаружении в коде неопределенных ситуаций
  - Встраивание механизмов защиты, санитайзеров и т.п.
- Необходимо сохранять приемлемую производительность
- Пилотный компилятор – на базе gcc

Статический анализ исходного кода

Технологии:

1. Доверенный компилятор
2. Инструменты статического анализа программ

- Доступные на рынке инструменты:
- Svace (ИСП РАН, Россия)
- Klocwork (RogueWave, США)
- Coverity (Synopsys, США)
- Java FindBugs (Maryland University, США)
- Pyhton: PyLint, PyType
- JavaScript: JSHint, JSLint, Google Closure Linter
- …

Фаззинг и динамический анализ

Технологии:

1. Фаззинг + Sanitizers
2. Динамическое символьное исполнение
3. Комбинированные инструменты

- Доступные на рынке инструменты:
- Anxiety (ИСП РАН, Россия)
- ИСП Фаззер (ИСП РАН, Россия)
- Peach Fuzzer (США, Peach Tech)
- Synopsys Defensics (Synopsys, США)
- angr (Open source)
- Americal Fuzzy Lop (Open source)
- Driller (Open source)
- …

Динамический анализ помеченных данных

Технологии:

1. Контролируемое выполнение в виртуальной машине
2. Автоматизированная обратная инженерия бинарного кода
3. Анализ потоков данных и управления на уровне бинарного кода

Доступные в РФ инструменты:

- Среда анализа бинарного кода ТРАЛ (ИСП РАН, Россия)

Известные инструменты:

- MAYHEM (ForAllSecure, США)
- APAC (2013-2015), VET (2013), CGC (2016), CHESS (2019) (Проекты DARPA)