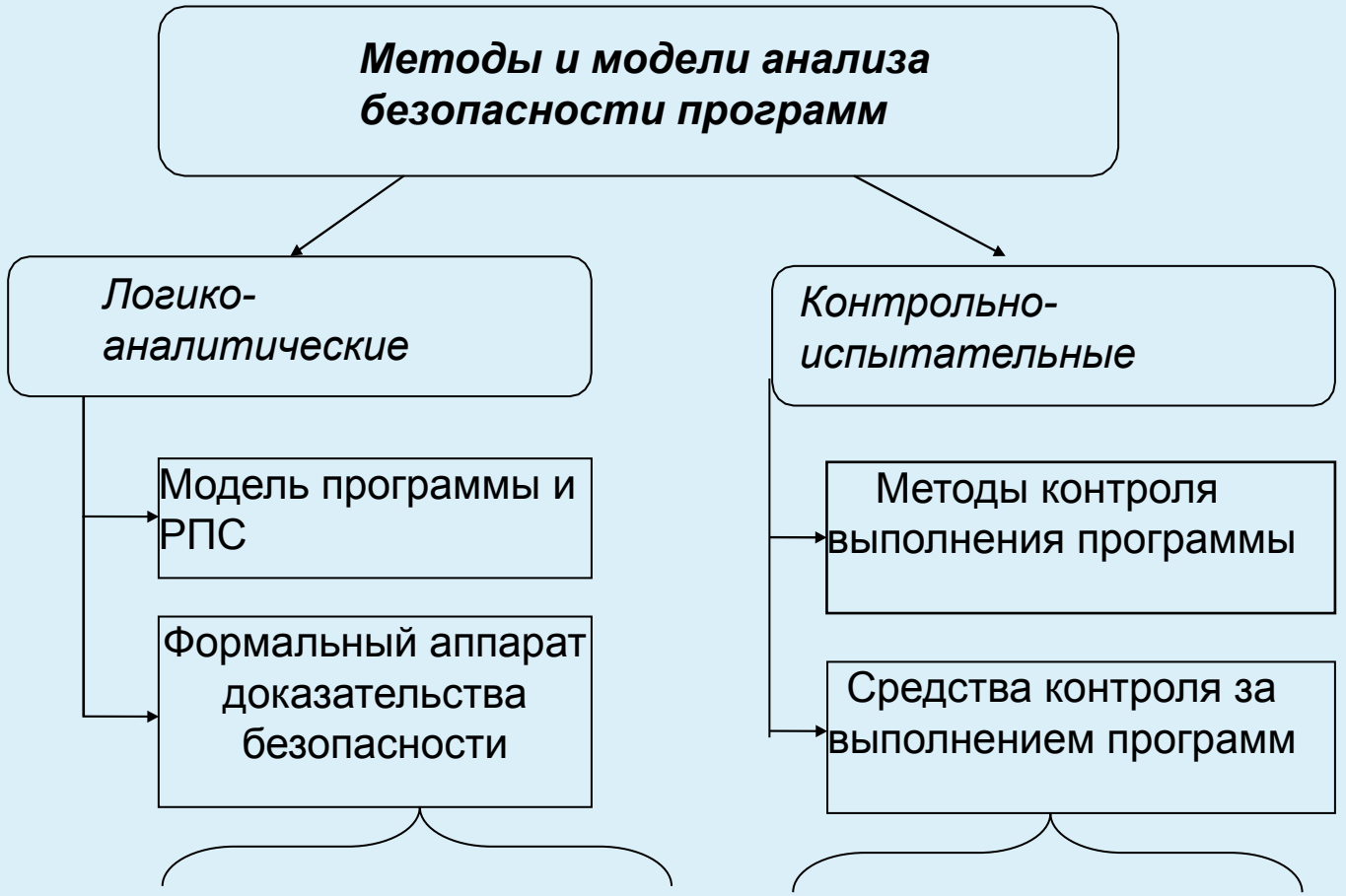
МЕТОДЫ И СРЕДСТВА АНАЛИЗА БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

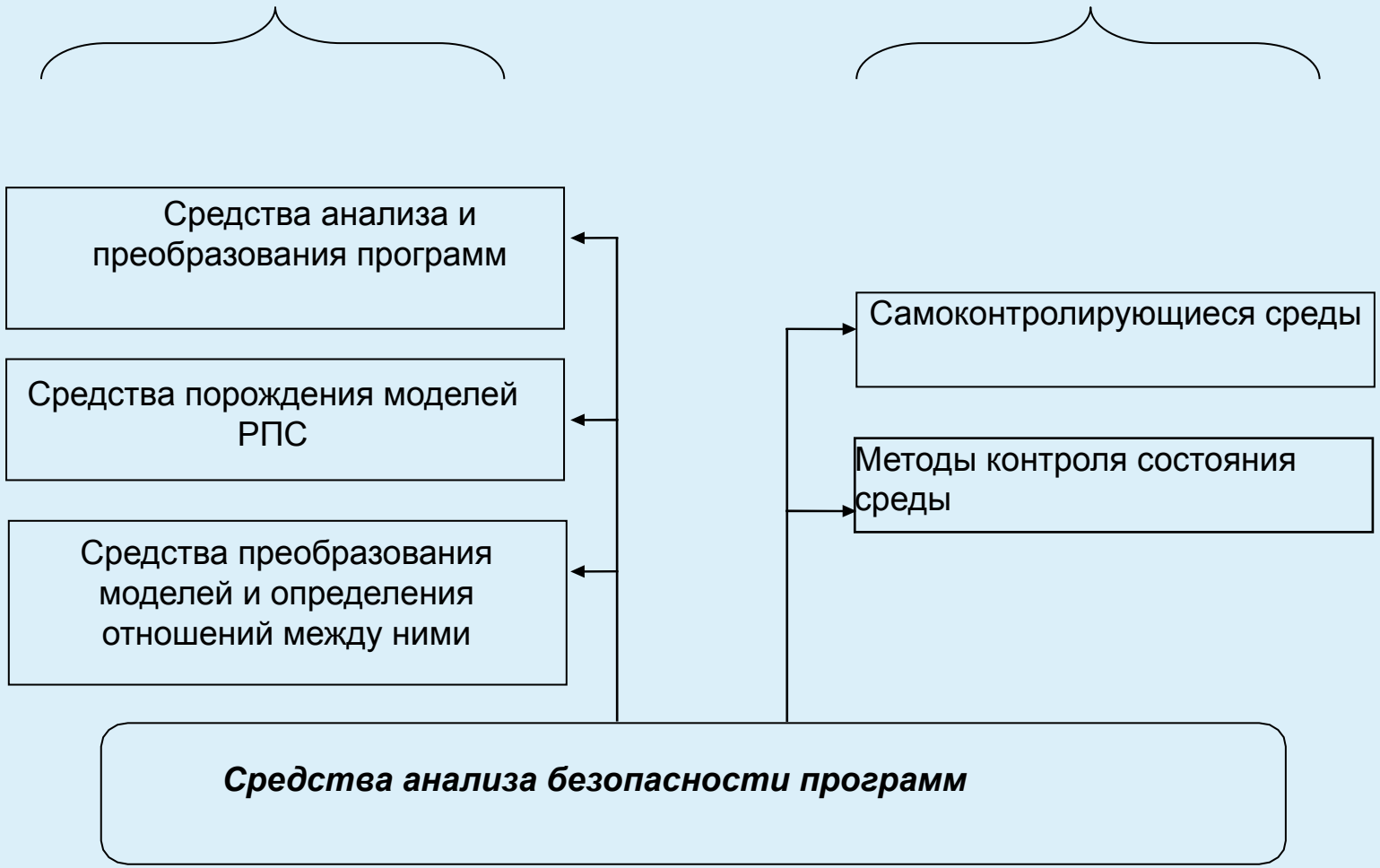
На основе различных программных средств обнаружения элементов РПС выработался набор методов, которыми осуществляется анализ безопасности ПО.
Методы, используемые для анализа и оценки безопасности ПО, делятся на две категории:

- контрольно- испытательные
- логико-аналитические

В основу данного разделения положены принципиальные различия в точке зрения на исследуемый объект (программу). Контрольно-испытательные методы анализа рассматривают РПС как факт нарушения безопасного состояния системы, а логико-аналитические – как доказательство наличия отношения эквивалентности между моделью исследуемой программы и моделью РПС.
В такой классификации тип используемых для анализа средств не принимается во внимание.

Комплексная система исследования безопасности ПО должна включать как контрольно-испытательные, так и логико-аналитические методы анализа, используя преимущества каждого их них. С методической точки зрения логико-аналитические методы выглядят более предпочтительными, так как позволяют оценить надежность полученных результатов и проследить последовательность (путем обратных рассуждений) их получения. Однако эти методы пока еще мало развиты.


Методы и модели анализа безопасности программ
Логико-аналитические
Контрольно-испытательные
Модель программы и РПС
Формальный аппарат доказательства безопасности
Методы контроля выполнения программы
Средства контроля за выполнением программ


Средства анализа и преобразования программ
Средства порождения моделей РПС
Средства преобразования моделей и определения отношений между ними
Самоконтролирующиеся среды
Методы контроля состояния среды
Средства анализа безопасности программ

КОНТРОЛЬНО-ИСПЫТАТЕЛЬНЫЕ МЕТОДЫ АНАЛИЗА БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Контрольно-испытательные методы - это методы, в которых критерием безопасности программы служит факт регистрации в ходе тестирования программы нарушения требований по безопасности, предъявляемых в системе предполагаемого применения исследуемой программы.

Контрольно-испытательные методы делятся на те, в которых контролируется процесс выполнения программы и те, в которых отслеживаются изменения в операционной среде, к которым приводит запуск программы. Эти методы наиболее распространены, так как они не требуют формального анализа и позволяют использовать имеющиеся технические и программные средства и быстро ведут к созданию готовых методик.

Контрольно-испытательные методы анализа безопасности начинаются с определения набора контролируемых параметров среды или программы. Необходимо отметить, что этот набор параметров будет зависеть от используемого аппаратного и программного обеспечения (от операционной системы) и исследуемой программы. Затем необходимо составить программу испытаний, осуществить их и проверить требования к безопасности, предъявляемые к данной программе в предполагаемой среде эксплуатации.

Наибольшую трудность здесь представляет определение набора критичных с точки зрения безопасности параметров программы и операционной среды. Они очень сильно зависят от специфики операционной системы и определяются путем экспертных оценок.

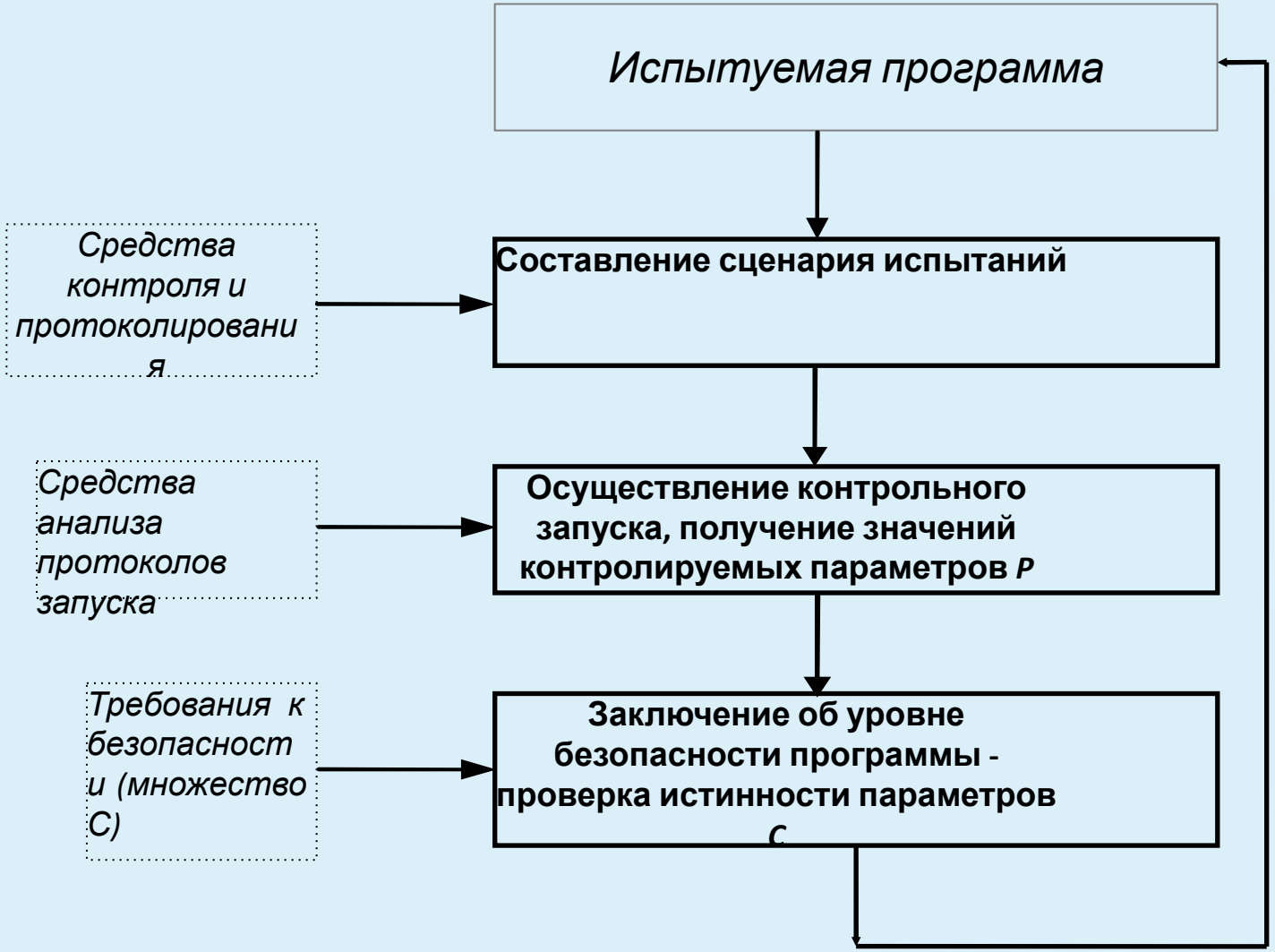


Схема анализа безопасности ПО с помощью контрольно- испытательных методов

ЛОГИКО-АНАЛИТИЧЕСКИЕ МЕТОДЫ КОНТРОЛЯ БЕЗОПАСНОСТИ ПРОГРАММ

При проведении анализа безопасности с помощью логико-аналитических методов строится модель программы и формально доказывается эквивалентность модели исследуемой программы и модели РПС. Например в качестве модели программы может выступать ее битовый образ, в качестве моделей вирусов множество их сигнатур, а доказательство эквивалентности состоит в поиске сигнатур вирусов в программе. Более сложные методы используют формальные модели, основанные на совокупности признаков, свойственных той или иной группе РПС.

Для проведения логико-аналитического анализа безопасности программы необходимо, во-первых, выбрать способ представления и получения моделей программы и РПС. После этого необходимо построить модель исследуемой программы и попытаться доказать ее принадлежность к отношению эквивалентности, задающему множество РПС.

В целом полный процесс анализа ПО включает в себя три вида анализа:

- лексический верификационный анализ;
- синтаксический верификационный анализ;
- семантический анализ программ.

Каждый из видов анализа представляет собой законченное исследование программ согласно своей специализации. Результаты исследования могут иметь как самостоятельное значение, так и коррелироваться с результатами полного процесса анализа.

Лексический верификационный анализ предполагает поиск распознавания и классификацию различных лексем (сигнатур) объекта исследования (программы), представленного в исполняемых кодах. В данном случае осуществляется поиск сигнатур следующих классов:

- сигнатуры вирусов;
- сигнатуры элементов РПС;
- сигнатуры «подозрительных функций»;
- сигнатуры штатных процедур использования системных ресурсов и внешних устройств.

Синтаксический верификационный анализ предполагает поиск, распознавание и классификацию синтаксических структур РПС, а также построение структурно-алгоритмической модели самой программы.

Решение задач поиска и распознавания синтаксических структур РПС имеет самостоятельное значение для верификационного анализа программ, поскольку позволяет осуществлять поиск элементов РПС, не имеющих сигнатуры. Структурно-алгоритмическая модель программы необходима для реализации следующего вида анализа - семантического

Семантический анализ предполагает исследование программы изучения смысла составляющих ее функций (процедур) в аспекте операционной среды компьютерной системы. В отличие от предыдущих видов анализа, основанных на статическом исследовании, семантический анализ нацелен на изучение динамики программы - ее взаимодействия с окружающей средой. Процесс исследования осуществляется в виртуальной операционной среде с полным контролем действий программы и отслеживанием алгоритма ее работы по структурно-алгоритмической модели.

Семантический анализ является наиболее эффективным видом анализа, но и самым трудоемким. По этой причине целесообразно сочетать в себе три перечисленных выше вида анализа. Выработанные критерии позволяют разумно сочетать различные виды анализа, существенно сокращая время исследования, не снижая его качества.

СРАВНЕНИЕ ЛОГИКО-АНАЛИТИЧЕСКИХ И КОНТРОЛЬНО- ИСПЫТАТЕЛЬНЫХ МЕТОДОВ АНАЛИЗА БЕЗОПАСНОСТИ ПРОГРАММ

Для сравнения методов предлагаются следующие признаки:

- представления предметной области,
- методы решения проблем неразрешимости легитимности и неперечислимости рабочего пространства,
- надежность получаемых результатов .

Надежность методов анализа определяется вероятностью ошибок первого и второго рода. Под ошибкой первого рода понимается принятие за РПС безопасной программы, а под ошибкой второго рода - объявление программы безопасной, когда на самом деле она содержит РПС.

С методической точки зрения логико-аналитические методы выглядят более предпочтительными, так как основываются на формальном подходе и приближают перспективное решение проблемы связанное с доказательством разрешимости множества РПС. Кроме того, они позволяют создать легкоприменяемые средства анализа, независящие от анализируемых программ. Однако на данное время любой из этих методов имеет существенный недостаток - исследование безопасности проводится лишь относительно некоторого подмножества РПС.
С практической точки зрения, - с точки зрения обеспечения безопасности контрольно-испытательные методы обладают рядом преимуществ, связанных с их привязкой к конкретной КС и программе, а также с их надежностью в отношении ошибок второго рода. Однако затраты, необходимые для организации процесса тестирования, являются преградой для их применения, за исключением критических компьютерных систем.

ни один из методов не имеет решающего преимущества перед другим. Использование методов той и другой группы должно опираться только на их соответствие решаемой задаче, необходимо применять те методы, которые в данной ситуации наиболее эффективны и оправданы.

Для полного решения проблемы анализа безопасности программ необходимо осуществить следующие действия

1. Создать теоретические основы анализа безопасности ПО, создать словарь предметной области и осуществить в рамках этого словаря формальную постановку задачи анализа безопасности ПО;
2. Создать методы анализа безопасности ПО, используя выбранные формальные определения, доказать их эффективность и реализуемость;
3. Создать конкретные программные средства, реализующие методы анализа безопасности программ в конкретных аппаратно- программных средах;
4. Создать методики применения этих средств и оценить их эффективность.

Методы	***Контрольно-испытательные***	***Логико-аналитические***
Способ представления предметной области	Пространство отношений программы с объектами КС.	Пространство программ.
Принцип поиска РПС	Фиксация установления программой нелегитимности отношения доступа к объектам КС.	Доказательство принадлежности программы к множеству РПС.
Поиск проблемы неразрешимости легитимности отношений	С помощью аппроксимации пространства легитимных отношений для данной программы и КС.	С помощью сведения к проблеме разрешимости множества РПС и анализ безопасности относительно разрешимого подмножества РПС.
Решение проблемы перечислимости рабочего пространства	Статистические и экстраполяционные методы теории верификации и функционального тестирования.	Не требуется.

Методы	***Контрольно-испытательные***	***Логико-аналитические***
Ошибки первого рода	Весьма вероятны. Чем строже требования, предъявляемые в заданной КС, тем больше вероятность ошибки.	При строгом доказательстве разрешимости подмножества РПС и корректно определенной характеристической функции исключены.
Ошибки второго рода	Маловероятны. Чем строже требования по безопасности, тем меньше вероятность ошибки.	Неизбежны. Определяются мощностью выбранного разрешимого подмножества РПС.
Преимущества	Не требует теоретической подготовки. Допускает использование имеющихся стандартных программных средств. Устойчивость к ошибкам второго рода. Метод отражает требования конкретных КС.	Опирается на формальные методы. Не требует значительных затрат на этапе применения. Высокая надежность полученных результатов относительно выбранного подмножества РПС. Инвариантность метода по отношению к различным классам программ. Позволяет создавать автоматические простые и доступные средства проверки безопасности.
Недостатки	Проведение испытаний требует существенных затрат времени и других ресурсов. Процесс тестирования требует выделения испытательной КС и должен проводится специалистами.	Подтверждены ошибками второго рода – проверяется лишь часть множества РПС.