## Методы и средства анализа безопасности программного обеспечения.

Широко известны различные средства программного обеспечения обнаружения элементов РПС - от простейших антивирусных программ-сканеров до сложных отладчиков и дизассемблеров - анализаторов и именно на базе этих средств и выработался набор методов, которыми осуществляется анализ безопасности ПО.

Авторы работ предлагают разделить методы, используемые для анализа и оценки безопасности ПО, на две категории: контрольно-испытательные и логико-аналитические (см. рис.2.3). В основу данного разделения положены принципиальные различия в точке зрения на исследуемый объект (программу). Контрольно-испытательные методы анализа рассматривают РПС через призму фиксации факта нарушения безопасного состояния системы, а логико-аналитические - через призму доказательства наличия отношения эквивалентности между моделью исследуемой программы и моделью РПС.

В такой классификации тип используемых для анализа средств не принимается во внимание - в этом ее преимущество по сравнению, например, с разделением на статический и динамический анализ.

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, т.к. позволяют оценить надежность полученных результатов и проследить последовательность (путем обратных рассуждений) их получения. Однако эти методы пока еще мало развиты и, несомненно, более трудоемки, чем контрольно-испытательные.

###### 2.2.1. 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. Тестирование может проводиться с помощью тестовых запусков, исполнения в виртуальной программной среде, с помощью символического выполнения программы, ее интерпретации и другими методами.

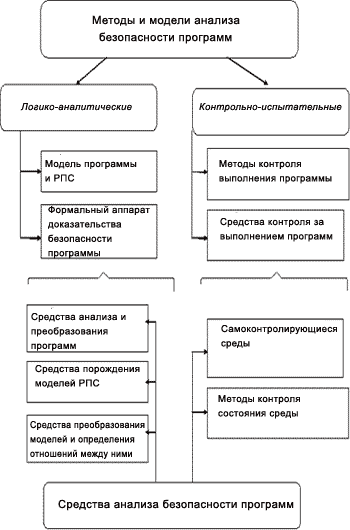


Рис. 2.3. Методы и средства анализа безопасности ПО

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, позволяют использовать имеющиеся технические и программные средства и быстро ведут к созданию готовых методик. В качестве примера - можно привести методику пробного запуска в специальной среде с фиксацией попыток нарушения систем защиты и разграничения доступа. Рассмотрим формальную постановку задачи анализа безопасности ПО для решения ее с помощью контрольно-испытательных методов.

Пусть задано множество ограничений на функционирование программы, определяющих ее соответствие требованиям по безопасности в системе предполагаемой эксплуатации. Эти ограничения задаются в виде множества предикатов С={ci(a1,a2,...am)|i=1,...,N} зависящих от множества аргументов A={ai|i=1,...,M}.

Это множество состоит из двух подмножеств:

* подмножества ограничений на использование ресурсов аппаратуры и операционной системы, например оперативной памяти, процессорного времени, ресурсов ОС, возможностей интерфейса и других ресурсов;
* подмножества ограничений, регламентирующих доступ к объектам, содержащим данные (информацию), то есть областям памяти, файлам и т.д.

Для доказательства того, что исследуемая программа удовлетворяет требованиям по безопасности, предъявляемым на предполагаемом объекте эксплуатации, необходимо доказать, что программа не нарушает ни одного из условий, входящих в С. Для этого необходимо определить множество параметров P={pi|i=1,...,K}, контролируемых при тестовых запусках программы. Параметры, входящие в это множество определяются используемыми системами тестирования. Множество контролируемых параметров должно быть выбрано таким образом, что по множеству измеренных значений параметров Р можно было получить множество значений аргументов А.

После проведения Тиспытаний по вектору полученных значений параметров Pi,i=1,...,T можно построить вектор значений аргументов Ai, i=1,...,T.

Тогда задача анализа безопасности формализуется следующим образом.

Программа не содержит РПС, если для любого ее испытания i=1,...,T множество предикатов C={cj(a1i,a2i...aMi)|j=1,...,N} истинно.

Очевидно, что результат выполнения программы зависит от входных данных, окружения и т.д., поэтому при ограничении ресурсов, необходимых для проведения испытаний, контрольно-испытательные методы не ограничиваются тестовыми запусками и применяют механизмы экстраполяции результатов испытаний, включают в себя методы символического тестирования и другие методы, заимствованные из достаточно проработанной теории верификации (тестирования правильности) программы.

Рассмотрим схему анализа безопасности программы контрольно-испытательным методом (рис.2.4).

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, на запротоколированных действиях программы и изменениях в операционной среде, а также используя методы экстраполяции результатов и стохастические методы.

Очевидно, что 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. Кроме того, в условиях ограниченных объемов испытаний, заключение о выполнении или невыполнении требований безопасности как правило будет носить вероятностный характер.

###### 2.2.2. Логико-аналитические методы контроля безопасности программ

При проведении анализа безопасности с помощью логико-аналитических методов (см. рис.2.5) строится модель программы и формально доказывается эквивалентность модели исследуемой программы и модели РПС. В простейшем случае 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.

Формальная постановка задачи анализа безопасности логико-аналитическими методами может быть сформулирована следующим образом.

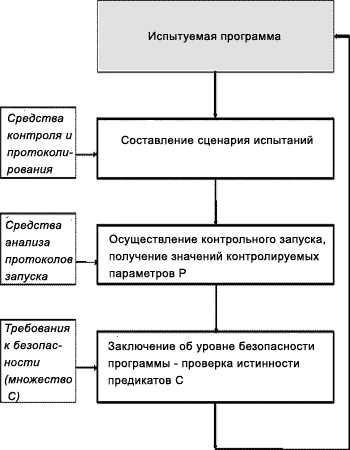


Рис.2.4. Схема анализа безопасности ПО с помощью контрольно-испытательных методов

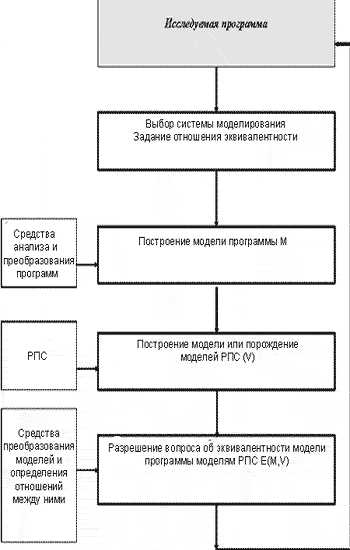


Рис. 2.5. Схема анализа безопасности ПО с помощью логико-аналитических методов

Выбирается некоторая система моделирования программ, представленная множеством моделей всех программ - Z. В выбранной системе исследуемая программа представляется своей моделью М, принадлежащей множеству Z. Должно быть задано множество моделей РПС V={vi|i=1,...,N}, полученное либо путем построения моделей всех известных РПС, либо путем порождения множества моделей всех возможных (в рамках данной модели) РПС. Множество V является подмножеством множества Z. Кроме того, должно быть задано отношение эквивалентности определяющее наличие РПС в модели программы, обозначим его Е(x,y). Это отношение выражает тождественность программы x и РПС y, где x - модель программы, y - модель РПС, и y принадлежит множеству V.

Тогда задача анализа безопасности сводится к доказательству того, что модель исследуемой программы М принадлежит отношению E(M,v), где v принадлежит множеству V.

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.

На основании полученных результатов можно сделать заключение о степени безопасности программы. Ключевыми понятиями здесь являются "способ представления" и "модель программы". Дело в том, что на компьютерную программу можно смотреть с очень многих точек зрения - это и алгоритм, который она реализует, и последовательность команд процессора, и файл, содержащий последовательность байтов и т.д. Все эти понятия образуют иерархию моделей компьютерных программ. Можно выбрать модель любого уровня модели и способ ее представления, необходимо только чтобы модель РПС и программы были заданы одним и тем же способом, с использованием понятий одного уровня. Другой серьезной проблемой является создание формальных моделей программ, или хотя бы определенных классов РПС. Механизм задания отношения между программой и РПС определяется способом представления модели. Наиболее перспективным здесь представляется использование семантических графов и объектно-ориентированных моделей.

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 объекта исследования (программа), представленного в исполняемых кодах. При этом лексемами являются сигнатуры. В данном случае осуществляется поиск сигнатур следующих классов:

* сигнатуры вирусов;
* сигнатуры элементов РПС;
* сигнатуры (лексемы) "подозрительных функций";
* сигнатуры штатных процедур использования системных ресурсов и внешних устройств.

Поиск лексем (сигнатур) реализуется с помощью специальных программ-сканеров.

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 методика сочетает в себе три перечисленных выше анализа. Выработанные критерии позволяют разумно сочетать различные виды анализа, существенно сокращая время исследования, не снижая его качества.