УДК 517.929

©Мукминов В.А., Войнов Ю.В., Баландин А.В.
Mukminov W., Voinov Y., Balandin A.

О НОВЫХ МЕТОДАХ И АЛГОРИТМАХ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ABOUT NEW METHOD AND ALGORITHM OF THE TESTING OF SOFTWARE

Аннотация. *В статье рассматриваются новые методы и алгоритмы тестирования программного обеспечения на наличие уязвимостей в условиях моделирования компьютерных атак. Представлена полная классификация методов верификации программного обеспечения и описание алгоритмов тестирования.*

Annotation. *New methods and algorithms of the testing of software are considered In article on presence of the criticality in condition of modeling of the computer attacks. Will Presented full categorization of the methods to verification of software and description algorithm testing.*

Ключевые слова. *Информационные системы, защищенность, уязвимости, тестирование, моделирование, компьютерные атаки.*

Key words. *The information systems, protection, criticality, testing, modeling, computer attacks.*

Последние десять лет в области защиты информации активно развивается направление, связанное с анализом уязвимостей программного обеспечения. Для задачи поиска и анализа уязвимостей программного обеспечения используют различные методики, методы и алгоритмы тестирования. В настоящее время методы тестирования программного обеспечения в условиях моделирования (имитации) компьютерных атак не были увязаны в общей классификации методов верификации программного обеспечения (ПО).

Обзор методов верификации ПО [1-3] позволил определить группы методов, нацеленные на оценку технических составляющих качества программного обеспечения. Классификация методов верификации ПО и краткое их описание представлены в таблице.

Обзор методов верификации ПО (см. таблицу) и нормативно-технических документов в этой области показал, что в настоящее время окончательно не проработан вопрос тестирования ПО на наличие уязвимостей в условиях моделирования компьютерных атак.

В общем случае под термином «уязвимость» понимается нарушение политики безопасности, вызванное неправильно заданным набором правил или ошибкой в ПО. Стоит отметить, что теоретически все информационные системы имеют уязвимости. Но то, насколько велик потенциальный ущерб от компьютерной атаки, использующей уязвимость, позволяет подразделять уязвимости на активно используемые и не используемые вовсе [4].

Однако именно через уязвимости ПО компьютерные атаки получают неавторизованный доступ к системе и к информации, обрабатываемой в ней. Существующие уязвимости являются точками, требующими особенно внимательной проверки при настройке системы безопасности против неавторизованного вторжения.

Существует два основных метода, при помощи которых проводятся проверки на наличие уязвимостей - сканирование и зондирование. Данные методы тестирования ПО в условиях компьютерных атак можно выделить в отдельную группу методов верификации ПО под условным названием «Функциональный анализ».

Мукминов Владислав Аликович – кандидат технических наук, начальник отдела 4 ЦНИИ Минобороны России, тел. (495) 544-26-24;
Войнов Юрий Витальевич – начальник отдела войсковой части 31659;
Баландин Алексей Владимирович – аспирант МГТУ им. Н.Э.Баумана.

Mukminov Wladislav Alikovich – Ph.D. , the chief of department 4 Central Scientific Research Institute Ministry of Defence of Russia, tel. (495) 544-26-24;
Voinov Yuriy – the chief of department troop part 31659;
Balandin Aleksey – the graduate student MGTU im. N.E.Baumana.

Классификация методов верификации программного обеспечения

Наименование группы методов верификации ПО	Краткое описание группы методов верификации ПО
Экспертиза	В качестве видов экспертиз традиционно выделяют организационные экспертизы, технические экспертизы, сквозной контроль, инспекции и аудиты. Экспертиза применима к любым свойствам ПО и на любом этапе жизненного цикла ПО, хотя для разных целей могут использоваться разные ее виды. Она позволяет выявлять практически любые виды ошибок, тем самым минимизируя время их существования. В то же время экспертиза не может быть автоматизирована и требует активного участия экспертов. Эмпирические наблюдения показывают, что эффективность экспертиз в терминах отношения количества обнаруживаемых ошибок к затрачиваемым на это ресурсам несколько выше, чем для других методов верификации.
Статический анализ	Статический анализ используется для поиска часто встречающихся ошибок по некоторым шаблонам. Такой анализ хорошо автоматизируется и может быть практически полностью возложен на инструментальные средства. Одним из известных недостатков статического анализа является использование строгих методов анализа, не допускающих пропуска ошибок, но приводящих к большому количеству сообщений о возможных ошибках, которые таковыми не являются.
Формальные методы верификации	Формальные методы верификации используют для анализа свойств ПО. Анализ формальных моделей выполняется с помощью специфических техник, таких как дедуктивный анализ, проверка моделей или абстрактная интерпретация. Формальные методы применимы только к тем свойствам, которые выражены формально в рамках некоторой математической модели. Соответственно, для использования таких методов необходимо затратить значительные усилия на построение формальных моделей.
Динамические методы верификации	Динамические методы верификации, в рамках которых анализ и оценка свойств программной системы делаются по результатам ее реальной работы или работы некоторых ее моделей. Основными методами являются обычное тестирование или имитационное тестирование, мониторинг, профилирование. Для применения динамических методов необходимо иметь работающую систему или хотя бы некоторые ее компоненты, или же их прототипы, поэтому нельзя использовать их на первых стадиях разработки ПО. Тем не менее с их помощью можно контролировать характеристики работы системы в ее реальном окружении, которые иногда невозможно проанализировать с помощью других подходов.
Синтетические методы верификации	В последние время появилось множество инструментальных средств, в которых применяются элементы нескольких перечисленных выше видов верификации. Так, в отдельные области выделились динамические методы, использующие элементы формальных, - тестирование на основе моделей и мониторинг формальных свойств. Ряд инструментальных средств построения тестов существенно использует как формализацию некоторых свойств ПО, так и статический анализ кода. Общая идея таких методов заключается в комплексировании преимуществ основных подходов к верификации.

Полная классификация методов верификации программного обеспечения представлена на рис. 1.

Сканирование – метод пассивного анализа наличия уязвимости без фактического подтверждения ее наличия – по косвенным признакам. Этот метод является наиболее простым для реализации. В терминах компании ISS данный метод получил название «логический вывод», в терминах компании Cisco этот процесс идентифицирует открытые порты, найденные на каждом сетевом устройстве, и собирает связанные с портами заголовки и служебную информацию. Каждый полученный заголовок сравнивается с таблицей правил определения сетевых устройств, операционных систем и потенциальных уязвимостей. На основе проведенного сравнения делается вывод о наличии или отсутствии уязвимости.

Сканирование используется с целью:

• обнаружения неизвестных устройств в сети;

• инвентаризации ресурсов сети (узлов, ОС, служб) без влияния на производительность;

• сбора информации о сети (топология, протоколы, средства защиты и т.п.).

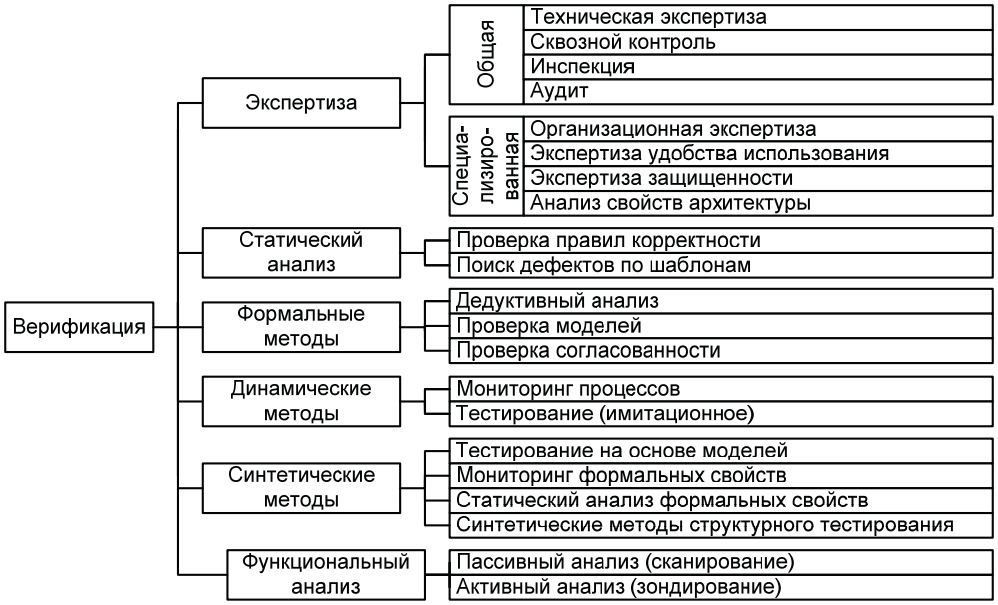

Рис. 1. Классификация методов верификации программного обеспечения

Зондирование – метод активного анализа, который выполняется путем имитации компьютерной атаки, использующей проверяемую уязвимость. Этот метод более длительный, чем «сканирование», но почти всегда гораздо более точный. В терминах компании ISS данный метод получил название «подтверждение», в терминах компании Cisco этот процесс использует информацию, полученную в процессе сканирования, для детального анализа каждого сетевого устройства. Этот процесс также использует известные методы реализации компьютерных атак для того, чтобы полностью подтвердить предполагаемые уязвимости и обнаружить другие уязвимости, которые не могут быть обнаружены пассивными методами.

Зондирование рекомендуется проводить для особо критичных систем, причем тестирование лучше всего осуществлять снаружи и изнутри (с определенным уровнем привилегий пользователя и оценкой возможностей повышения этих привилегий).

На практике указанные методы реализуются следующими алгоритмами, представленными на рис. 2.

Алгоритм проверки заголовков пакетов. Указанный алгоритм представляет собой ряд проверок типа “сканирование” и позволяет делать вывод об уязвимости, опираясь на информацию в заголовке ответа на запрос. Типичный пример такой проверки - анализ заголовков программы Sendmail или FTP-сервера, позволяющий узнать их версию, и на основе этой информации сделать вывод о наличии в них уязвимости. Является наи-

Рис.2. Алгоритмы тестирования программного обеспечения на наличие уязвимостей

более простым для реализации алгоритмом проверки наличия на сканируемом узле уязвимости.

Алгоритм статического анализа исходного кода программы. Статический анализ кода - это анализ программного обеспечения, который производится над исходным кодом программ и реализуется без реального исполнения программы. Глубина анализа может варьироваться от определения поведения отдельных операторов до анализа, включающего весь имеющийся исходный код.

Используя статические анализаторы кода, можно выявлять следующие виды уязвимостей ПО:

• переполнение буфера (возникает из-за отсутствия контроля за выходом за пределы массива в памяти во время выполнения программы);

• уязвимости испорченного ввода (могут возникать в случаях, когда вводимые пользователем данные без достаточного контроля передаются интерпретатору некоторого внешнего языка);

• ошибки форматных строк (возникают из-за недостаточного контроля параметров при использовании функций форматного ввода-вывода printf, fprintf, scanf и т. д. стандартной библиотеки языка Си);

• уязвимости ошибок синхронизации (связаны с многозадачностью и приводят к ситуациям, когда программа, не рассчитанная на выполнение в многозадачной среде, может считать, что используемые ею при работе файлы не может изменить другая программа).

Алгоритм подготовки некорректных данных на вход программ. Выполнение алгоритма подготовки некорректных данных на вход программ заключается в передаче на вход тестируемой программы блока случайных или специально сформированных данных (в большинстве своем это некорректно составленные данные) с целью анализа последующей реакции программы, в ходе которого делается вывод о наличии ошибок и уязвимостей в тестируемом ПО.

В случае, если программа зависает или аварийно завершает работу, это считается нахождением ошибки в программе, которая может привести к обнаружению определенной уязвимости.

Основными преимуществами алгоритма являются:

• легкость автоматизации;

• большой объем тестирования с множеством вариаций;

• выявление большого количества ошибок и уязвимостей.

Алгоритм имитации компьютерных атак. Алгоритм имитации компьютерных атак заключается в моделировании компьютерных атак. Однако существуют случаи, когда имитация компьютерных атак не всегда может быть реализована. Такие случаи можно разделить на две категории: ситуации, в которых тест приводит к “отказу в обслуживании” анализируемого узла, и ситуации, при которых уязвимость в принципе не предназначена для реализации компьютерной атаки.

В некоторых случаях нежелательно использовать имитацию компьютерных атак (например, для анализа защищенности критически важных серверов), так как это может привести к большим затратам (материальным и временным) на восстановление работоспособности выведенных из строя элементов информационных систем.

Вывод. Рассмотрены новые методы и алгоритмы тестирования программного обеспечения на наличие уязвимостей в условиях моделирования компьютерных атак. Представленные в статье алгоритмы могут быть реализованы в виде инструментальных комплексов, применение которых позволит повысить эффективность работ по оценке реального уровня защищенности объектов информатизации.

Литература

1. Кулямин В.В. Методы верификации программного обеспечения. – М.: Горячая линия – Телеком. 2006.
2. Липаев В.В. Программная инженерия. Методологические основы. – М.: Изд-во «Теис». 2006.
3. Синицын С.В., Налютин Н.Ю. Верификация программного обеспечения. Курс лекций. – М.: МИФИ. 2006.
4. Черноскутов А.И., Зорин Э.Ф., Рыжов Б.С. Оценка уязвимости и защищенности автоматизированной системы военного назначения на основе метода Саати и его модификаций. Двойные технологии, №2 (51).

Материал поступил в редакцию 22. 02. 2011 г.