УДК 004.056.53

В.М. РУВИНСКАЯ, А.А. ЛОТОЦКИЙ

Одесский национальный политехнический университет, Украина

СЦЕНАРИИ ДЛЯ ОПИСАНИЯ РАЗЛИЧНЫХ ТИПОВ ВРЕДОНОСНЫХ ПРОГРАММ И ИХ ТЕСТИРОВАНИЕ

В статье рассматриваются сценарии для описания действий программы и способы выявления вредоносных программ на базе сценариев. Приведены обобщенные иерархии сценариев для основных типов вредоносных программ: червей, вирусов, троянских программ, а также описаны поведения некоторых наиболее распространенных представителей этих классов с помощью регулярных выражений. Также предлагается информационная технология для автоматизированного тестирования анализаторов вредоносных программ, построенных на основе сценариев, позволяющая уменьшить трудозатраты за счет использования сценариев, лежащих в основе самих анализаторов. Тестирование предполагает два этапа: тестирование составляющих сценариев нижнего уровня, то есть правильности перехватов отдельных «подозрительных» действий в системе, и полное тестирование, в качестве исходных данных для которого используются вредоносные программы.

Ключевые слова: *безопасность, вредоносная программа, анализатор вредоносных программ, сценарии, тестирование.*

Постановка задачи

Работа посвящена защите компьютерных систем от вредоносных программ и является продолжением опубликованной ранее статьи [1], в которой нами было предложено в качестве аналитического компонента для проактивной защиты, а именно, в поведенческих и эвристических анализаторах использовать экспертные системы на основе сценариев. При этом, в отличие от традиционных решений, анализируются не только отдельные «подозрительные» действия или их наборы, а и сценарии таких действий с иерархической структурой и правилами взаимосвязи, описывающие «подозрительные» поведения. Эвристический анализатор и/или поведенческий блокиратор в составе антивируса отслеживает каждое из этих действий и передает их в модуль анализа на базе сценариев поведения. Далее модуль анализа делает выводы, является ли данное поведение подозрительным, и в случае необходимости информирует пользователя. В [1] приведены примеры сценариев и предложен аппарат для их описания на основе регулярных выражений.

Первой целью дальнейшего исследования является создание базы знаний сценариев для основных типов вредоносных программ. Для этого решаются следующие задачи:

– провести обобщенный анализ поведения для каждого типа, в частности, для самых в настоящее время распространенных, червей, троянских программ и вирусов;

– далее проанализировать их подтипы и так далее по иерархии до самого нижнего уровня базовых подцелей, которые должен перехватывать технический компонент антивируса;

– записать эти поведения на языке сценариев.

Так как при разработке любого программного продукта возникает необходимость оценки его качества, второй целью нашего исследования является уменьшение трудозатрат при тестировании анализаторов вредоносных программ за счет использования сценариев, лежащих в основе самих анализаторов.

1. Сценарии для основных типов вредоносных программ

Рассматривают нижеследующие основные категории вредоносных программ и их поведение (за основу была взята классификация Лаборатории Касперского [2]).

1) Червь (Worm – сетевой червь) - это вредоносная программа, распространяющаяся по сетевым каналам и способная к самостоятельному преодолению систем защиты компьютерных сетей, а также к созданию и дальнейшему распространению своих копий, не обязательно совпадающих с оригиналом. Черви классифицируются по типу проникновения в систему, или по типу распространения.

Обобщенное поведение сетевых червей, описанное в виде иерархии сценариев верхнего уровня, показано на рис. 1.

Жирной чертой обведены прямоугольники, представляющие этапы, по которым классифицируются вредоносные программы, пунктиром – необязательные действия. Линии показывают связи между сценариями по иерархии.

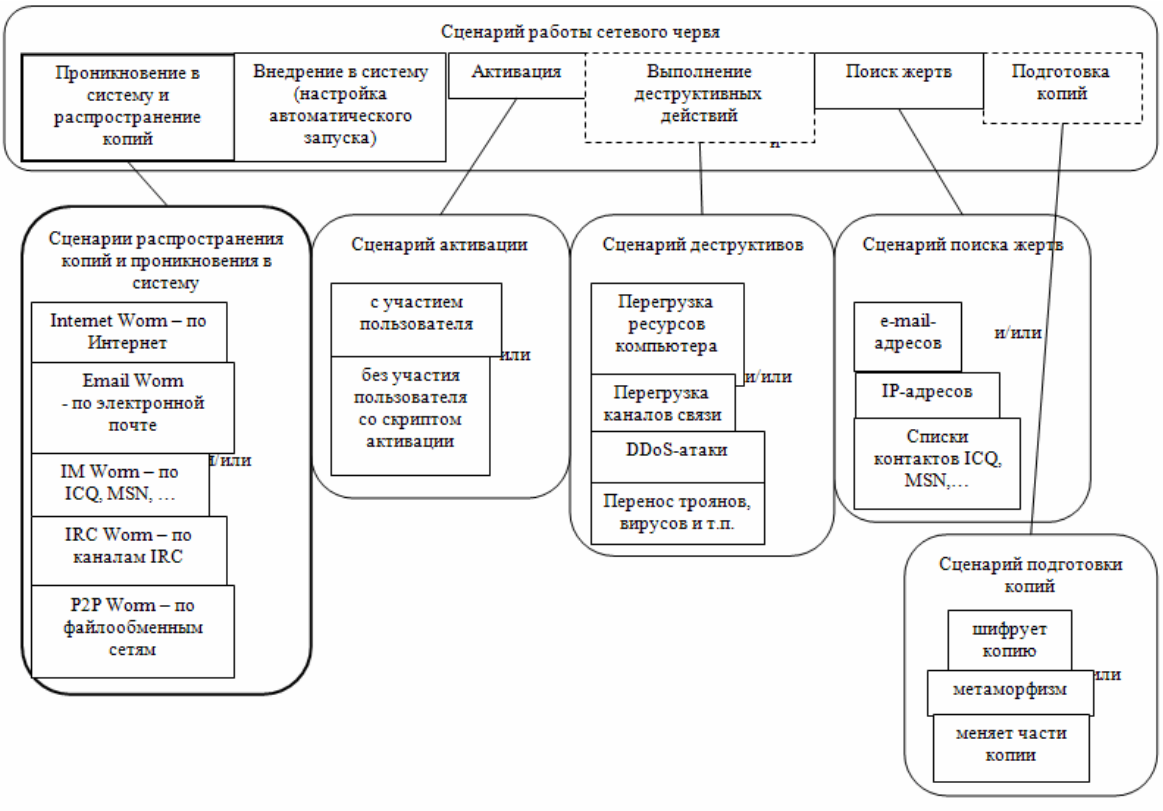

Рис. 1. Обобщенная иерархия сценариев для описания поведения сетевых червей

На рис. 2 показан сценарий, описывающий поведение червей и троянских программ при внедрении своих копий в систему, т.е. возможные способы их автозагрузки.

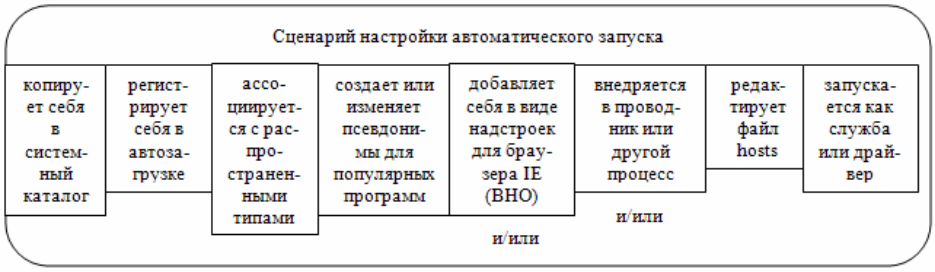

Рис. 2. Различные способы настройки автозапуска для вредоносных программ

Видно, что такая иерархия описывает достаточно большое множество «подозрительных» поведений. Например, простой почтовый червь (не выполняющий явных деструктивных действий) активируется при чтении письма пользователем, для поиска жертвы сканирует адресную книгу и/или файлы на компьютере для нахождения адресов электронной почты, а затем посылает свои копии по этим адресам. На рис. 3 представлено такое поведение.

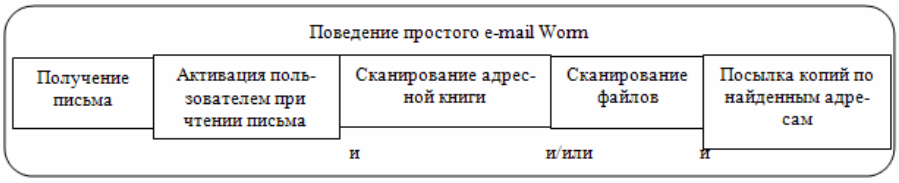

Рис. 3. Поведение простого почтового червя

Ниже приведена запись поведения на языке регулярных выражений. «–» обозначено последовательное выполнение, «|» – ИЛИ.

Получение письма – Активация пользователем при чтении письма – (Сканирование адресной книги | Сканирование файлов) – Посылка своей копии по найденным адресам.

2) Троянская программа – программа, основной целью которой является вредоносное воздействие по отношению к компьютерной системе. Чаще всего он сам не распространяется, а проникает вместе с червем или вирусом, либо каким-то образом маскируется, и его загружает пользователь, не догадываясь об этом, либо загружается автоматически. Трояны классифицируются по типу деструктивных действий. Обобщенное поведение троянских программ, описанное в виде иерархии сценариев верхнего уровня, показано на рис. 4.

Ниже приведена запись поведения некоторых наиболее распространенных троянских программ в виде двухуровневой иерархии с помощью регулярных выражений.

Проникнуть вместе с червем из Интернет – Скопировать_себя_в_системный_каталог – Зарегистрироваться_в_автозапуске – Получить_команды_через_интернет – Выполнить_команду

Выполнить_команду = Загрузить_файл | Установить_файл | Просканировать_компьютер | Установить_себя_на_компьютер | Мониторинг_пакетов | Сканирование_сети| DoS | Socks_сервер | HTTP_сервер | Отсылка_данных

3) Вирус – это программа, способная создавать свои дубликаты (не обязательно совпадающие с оригиналом) и внедрять их в файлы, системные области компьютера и прочие объекты. При этом дубликаты сохраняют способность к дальнейшему распространению. Исторически вирусы были первыми вредоносными программами. Вирусы классифицируются по среде активации. Обобщенное поведение вирусов, описанное в виде иерархии сценариев верхнего уровня, показано на рис. 5.

Ниже приведена запись с помощью регулярных выражений поведения файлового вируса. «+» означает итерацию, «?» – необязательная подцель.

Проникает с носителя | Проникает из Интернет – Активация из зараженного файла – (Поиск_исполняемого_файла – Открытие_файла – Запись_в_секцию_кода – Модификация_заголовка?)+

2. Тестирование анализаторов вредоносных программ

Одним из наиболее трудозатратных, но, в тоже время, обязательным, является функциональное тестирование, т.е. проверка соответствия реализованных функций требованиям технического задания [3].

Обычно для функционального тестирования строятся так называемые варианты тестирования (Test Cases), содержащие исходные данные, действия с ними в программе и ожидаемый результат. Например, Test Case, соответствующий рассмотренному ранее сценарию внедрения вредоносного кода в систему, приведен ниже.

Действия:

1) Создать файл.

2) Добавить его в автозагрузку.

Ожидаемый результат:

Поведенческий блокиратор перехватил действия и определил поведение как подозрительное.

В общем случае тестирование требует большого количества рутинных действий для «ручного» создания вариантов тестирования. А поведенческий анализатор содержит в себе большое количество динамически изменяющихся сценариев, каждый из которых требует верификации. Таким образом, необходимо их автоматизированное тестирование.

Предлагается Test Case-ы для тестирования поведения вредоносных программ не формировать вручную, а создавать автоматически на базе сценариев, которые лежат в основе самих анализаторов. При этом автоматизируется процесс функционального тестирования: программа-тестировщик строит на основе базы знаний сценариев варианты тестирования, затем их выполняет и собирает результаты о корректности работы анализатора.

Нами предлагается два способа автоматизации тестирования:

1) Частичное тестирование отдельных подцелей сценариев. Test Case-ы автоматически строятся на основе отдельных действий, которые являются составляющими сценариев. Затем программа тестирования автоматически запускает эти действия на выполнение. При этом тестируемый анализатор должен среагировать правильно на эти действия. Этот способ позволяет проверить не полностью сценарии, а лишь их элементы.

2) Полное тестирование сценариев. Test Case-ы автоматически строятся на основе сценариев поведения вредоносных программ. Затем программа тестирования запускает вредоносные программы на выполнение (в контролируемом окружении). При этом анализатор должен в большинстве случаев правильно диагностировать каждую вредоносную программу, а для не вредоносных программ в большинстве случаев не выдавать ложных срабатываний.

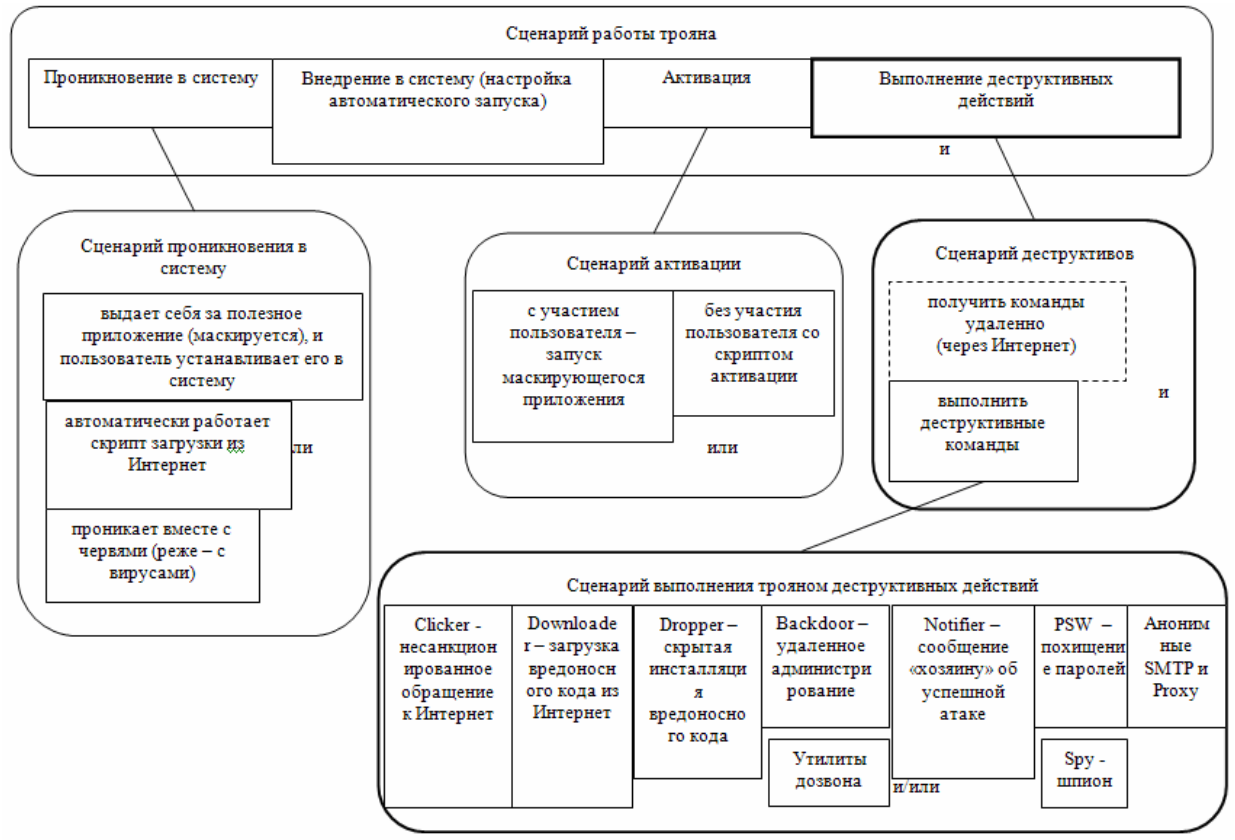

Рис. 4. Обобщенная иерархия сценариев для описания поведения троянов

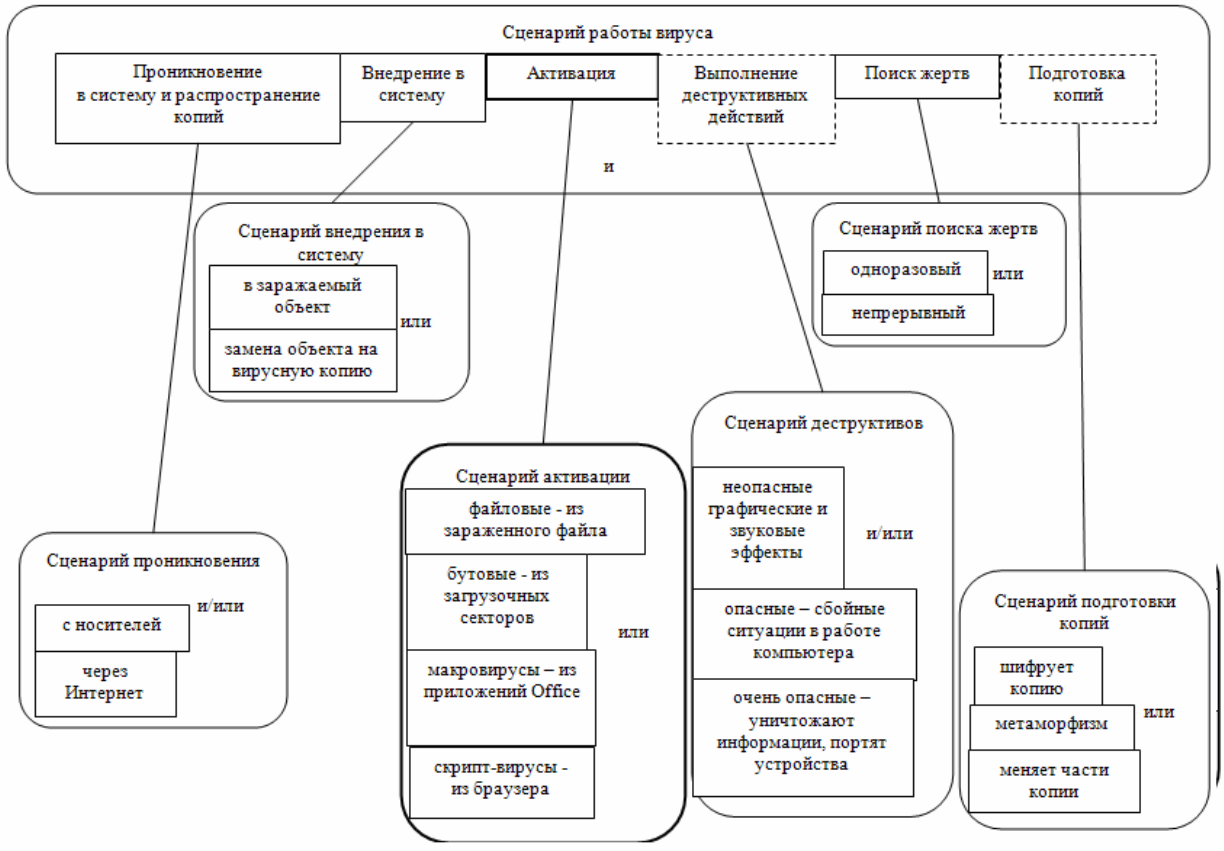

Рис. 5. Обобщенная иерархия сценариев для описания поведения вирусов

Программа-тестировщик автоматически генерирует Test Case-ы, запускает их и показывает результаты их срабатывания. Участие человека-тестировщика состоит в сопоставлении каждому сценарию вредоносных программ, на которые ему следует реагировать, а также анализе и интерпретации результатов тестирования.

На рис.6. представлена UML-диаграмма прецедентов, показывающая, какие функции выполняет программа-тестировщик, а также как происходит взаимодействие с ней человека-тестировщика, антивируса на основе сценариев, самих сценариев, а также вычислительной системы.

Рассмотрим далее информационные технологии для каждого способа автоматизации тестирования, т.е. этапы тестирования поведенческих блокираторов, разработанных на основе сценариев.

Информационная технология для частичного тестирования отдельных подцелей сценариев.

1) Тестировщик на базе элементов сценариев создает Test Case-ы (варианты тестирования).

2) Для каждого Test Case выполняет описанные в нем действия.

3) Для каждого TestCase проверяется реакция поведенческого анализатора. Если поведенческий анализатор верно определил действие, то тест считается пройденным, иначе – проваленным.

Информационная технология для полного тестирования сценариев

Предварительное условие для данного тестирования: каждому сценарию поведения должна быть сопоставлена одна или несколько вредоносных программ.

1) Тестировщик на базе сценариев создает Test Case-ы (варианты тестирования).

2) Для каждого Test Case-а тестировщик запускает на выполнение соответствующую вредоносную программу (в контролируемом окружении).

3) Для каждого Test Case-а проверяется реакция поведенческого анализатора.

4) Если реакцией поведенческого анализатора был сценарий, описывающий вредоносную программу, то тест считается пройденным, иначе – проваленным.

Для всех Test Case-ов запускаются обычные программы и проверяется реакция поведенческого анализатора. Если ни один сценарий не сопоставляется Test Case-у, то тест считается пройденным, иначе – проваленным.

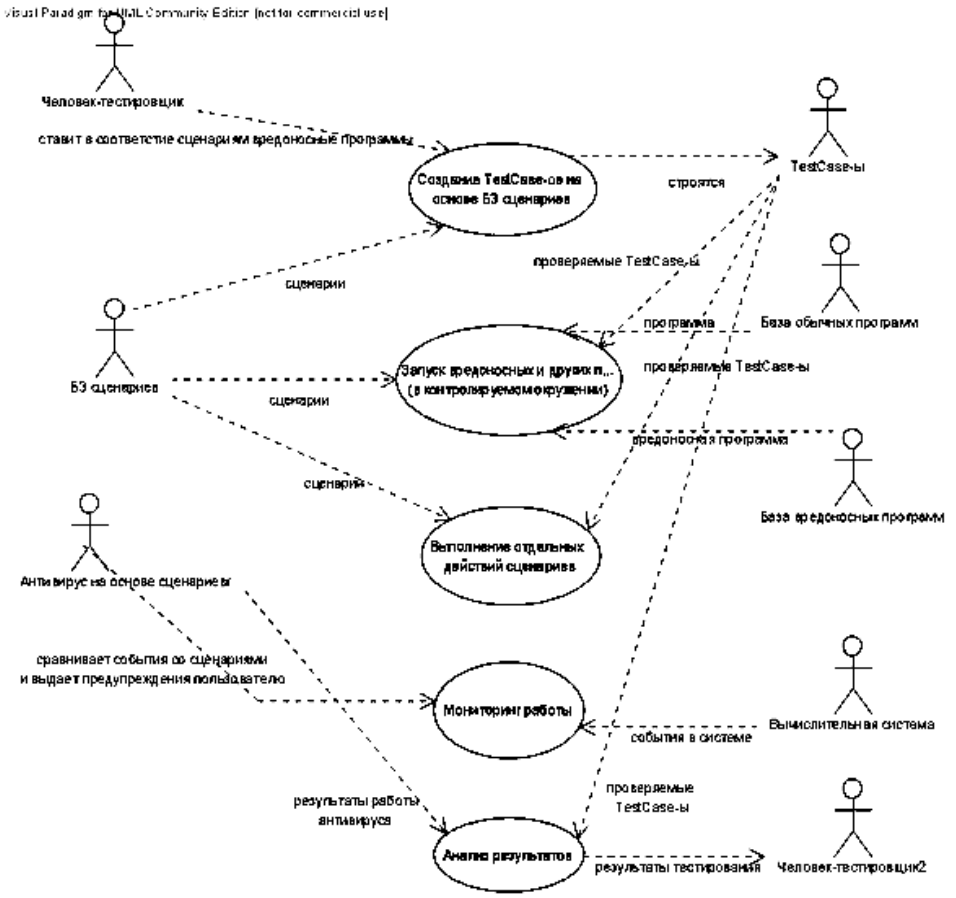

Рис. 6. Диаграмма вариантов использования для автоматизированного тестирования

Таблица 1

Сравнение автоматизированного и неавтоматизированного подхода

Действия, которые необходимо выполнить пользователю	Неавтоматизированный подход	Автоматизированный подход
Сопоставить сценарию одну или несколько вредоносных программ	+	+
Создать Test Case	+	–
Выполнить Test Case	+	–
Проверить результат	+	–
Оценить результат	+	+

В результате тестирования мы получим следующую информацию:

1) Срабатывает ли поведенческий блокиратор на все действия в системе, которые предполагается анализировать (прописанные в сценариях).

2) Сопоставляет ли модуль анализа всем прописанным событиям в системе подходящий сценарий вредоносных программ.

3) Процент вредоносных программ, детектируемый поведенческим блокиратором.

4) Процент ложных срабатываний.

Выводы

Таким образом, сценарный подход при анализе вредоносных программ позволяет:

– улучшить характеристики работы анализаторов, а именно, уменьшить количество ложных срабатываний за счет анализа не отдельных событий в системе, а их последовательностей;

– дает возможность автоматизировать процесс тестирования анализаторов, что снижает трудозатраты, поскольку тесты создаются, выполняются и проверяются автоматически (см. табл. 1).

Литература

1. Рувинская В.М. Эвристические методы детектирования вредоносных программ / В.М. Рувинская, Е.Л. Беркович, А.А. Лотоцкий // Искусственный интеллект. – 2008. – №8. – С. 197-207.

2. Лаборатория Касперского. Вирусы и средства борьбы с ними, учебный курс. [Электронный ресурс] // Режим доступа к статье: http://www.kaspersky.ru/view.html?id=32.

3. *Котляров В.П. Основы тестирования программного обеспечения. Курс лекций. Учебное пособие. / В.П. Котляров. – Интернет университет информационных технологий ИНТУИТ.ру, 2006. – 360 с.*

Поступила в редакцию 21.01.2010

Рецензент: д-р техн. наук, проф., проф. кафедри комп'ютерних інтелектуальних систем та мереж О.В. Дрозд, Одеський національний політехнічний університет, Одеса, Україна.

СЦЕНАРІЇ ДЛЯ ОПИСУ РІЗНИХ ТИПІВ ШКІДЛИВИХ ПРОГРАМ ТА ЇХ ТЕСТУВАННЯ

В.М. Рувінська, О.А. Лотоцький

У статті розглядаються сценарії для опису дій програми та способи виявлення шкідливих програм на базі сценаріїв. Наведено узагальнені ієрархії сценаріїв для основних типів шкідливих програм: червів, вірусів, троянських програм, а також описані поведінки деяких найбільш поширених представників цих класів за допомогою регулярних виразів. Також пропонується інформаційна технологія для автоматизованого тестування аналізаторів шкідливих програм, побудованих на основі сценаріїв, що дозволяє зменшити трудовитрати за рахунок використання сценаріїв, що лежать в основі самих аналізаторів.

Ключові слова: безпека, шкідлива програма, аналізатор шкідливих програм, сценарії, тестування.

SCENARIOS FOR DIFFERENT TYPES OF MALICIOUS PROGRAMS AND THEIR TESTING

V.M. Ruvinskaya, A.A. Lototsky

In the article are considered the scenarios for the description of the program's actions and the detection of malicious programs on the base of scenarios. We propose generalized scenarios hierarchies for the basic types of the malicious programs: worms, viruses, trojan programs, and are also described the behavior of some most common representatives of these classes with the use of regular expressions. Also it is proposed information technology for such malware analyzer's automated testing. The scenarios make it possible to decrease labor expenses due to the use of scenarios, which lie at the basis of analyzers themselves.

Keywords: security, malware, malware analyzer, scenarios, testing.

Рувинская Виктория Михайловна – канд. техн. наук, доц., Одесский национальный политехнический университет, Одесса, Украина, e-mail:iolnlen@te.net.ua.

Лотоцкий Александр Анатолиевич – аспирант, Одесский национальный политехнический университет, Одесса, Украина, e-mail: for_lotos@mail.ru.