УДК 004.492
DOI: 10.12737/article_5a02fa05dbd4a6.89264719

И.С. Кабак, Н.В. Суханова, С.А. Шептунов

АППАРАТНО-ПРОГРАММНЫЙ СПОСОБ БОРЬБЫ С КОМПЬЮТЕРНЫМИ ВИРУСАМИ

С целью защиты информации и программного обеспечения от деструктивного воздействия компьютерных вирусов предусмотрено дополнение аппаратуры компьютера специальной микросхемой антивирусного сопроцессора и внесение изменений в сервисное стандартное программное обеспечение - компиляторы, редакторы связей и загрузчики. Разработан новый способ защиты программы от модификации в процессе функционирования, что является характерной особенностью вирусных программ.

Ключевые слова: компьютерный вирус, антивирус-инспектор, автоматическое восстановление программы, антивирусный сопроцессор.

I.S. Kabak, N.V. Sukhanova, S.A. Sheptunov

HARDWARE-SOFTWARE METHOD OF FIGHT AGAINST COMPUTER VIRUS

A work purpose – protection of information and software against destructive effects of computer viruses. There is provided a computer hardware supplement with a special chip of antivirus co-processor and an introduction of changes into the service standard software – compiler routines, editors of ties and loaders.

As a result of investigation a new method of software protection against modification in the course of functioning which is a character of virus programs is developed.

When a virus is defined the work of an infected program is stopped for the time being essential for its working capacity recovery.

Key words: computer virus, antivirus-inspector, automatic program recovery, antivirus co-processor.

Введение

Одним из ключевых направлений информационной безопасности является борьба с компьютерными вирусами, потери от которых чрезмерно велики. Так, один только компьютерный вирус CodeRed в 2001 году вызвал потерю 2,62 млрд долларов США, а суммарные потери в 2001 году составили 13,2 млрд долларов США [5; 8]. Более 40 % потерь данных в компьютерных системах являются следствием вирусов. Работы в области компьютерной вирусологии не только актуальны, но и жизненно необходимы.

Состояние вопроса

Чаще всего обнаружение компьютерных вирусов осуществляется на базе сигнатурного детектирования, когда происходит поиск фиксированных последовательностей, характерных для тех или иных вирусов. Этот подход требует постоянной модификации, постоянного обновления базы сигнатур при появлении новых компьютерных вирусов.

Известен подход, когда определение вируса основано на его характерных действиях. Например, обнаруживают попытки изменения запрещенных к изменению областей памяти. Работа некоторых служебных программ требует действий, которые могут классифицироваться как деятельность вируса, что приводит к ложному детектированию.

Для защиты от вирусов использовались программы-ревизоры. Они запоминали исходное состояние программного кода, областей данных и системных областей при загрузке компьютера, проводилось сравнение текущего и исходного состояний информации и определялись различия. Этот метод требовал существенных трудозатрат, поскольку в процессе работы данные постоянно меняются. Примерами та-

ких программ являлись антивирус-ревизор диска ADINF и AVP Inspector [5].

Работа антивирусных программ существенно сказывается на функционировании программы на компьютере, что недопустимо в некоторых случаях, например при работе систем реального времени. Сам процесс поиска вирусов весьма трудоемок и занимает несколько часов.

Описание способа антивирусной защиты

Основная идея предлагаемого способа защиты состоит в разделении двух процессов - выполнения программы (на одном или нескольких процессорах) и антивирусного контроля. Для распараллеливания этих процессов на системную плату установлен дополнительный антивирусный сопроцессор, как это показано на рис. 1. Этот сопроцессор имеет выход на системную шину.

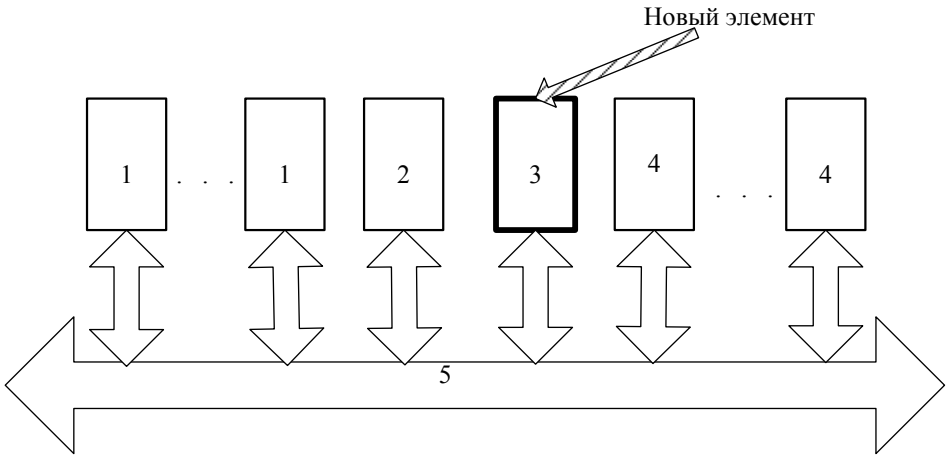

Рис.1. Структура антивирусной защиты для компьютерной системы:
1 - центральные процессоры; 2 - оперативная память; 3 - антивирусный сопроцессор (процессор);
4 - внешние устройства (НЖМД, оптические дисководы и т.п.); 5 - шина

Как и основные (центральные) процессоры, антивирусный сопроцессор имеет возможность прямого обращения через шину (п. 5 рис. 1) к оперативной памяти (п. 2 рис. 1), а также может обмениваться информацией с центральными процессорами.

Антивирусный сопроцессор получает от центральных процессоров данные о том, какая программа - процесс им активируется и какие блоки памяти команд этот процесс использует. На основании этих данных антивирусный сопроцессор формирует свою контрольную таблицу. Основная функция антивирусного сопроцессора - постоянно (или периодически) контролировать содержимое блоков команд оперативной памяти каждого активного в данный момент процесса на компьютере. Контроль осуществляется подсчетом хеш-функции блока команд оперативной памяти. На рис. 2 приведена структурная схема антивирусного сопроцессора.

Антивирусный сопроцессор состоит из двух основных блоков [1-4; 7]:

- Специализированного сопроцессора, аппаратно реализующего ряд алгоритмов.
- Оперативной памяти, хранящей контрольную таблицу.

Рис. 2. Структурная схема антивирусного сопроцессора:
1 - связь с шиной компьютера (п. 5 рис. 1); 2 - специализированный процессор;
3 - внутренняя шина антивирусного сопроцессора; 4 - оперативная память
антивирусного сопроцессора

Специализированный процессор аппаратно реализует следующие функции:

1. При загрузке программы получает от операционной системы адреса начала и конца блоков команд программы в оперативной памяти компьютера (п. 2 рис. 1).

2. Подсчитывает значение хеш-функции для каждого блока команд программы и помещает в свою локальную оперативную память (п. 4 рис. 2) адреса начала и конца для каждого блока и значение хеш-функции.

3. Последовательно подсчитывает значения хеш-функций для всех блоков оперативной памяти компьютера (п. 2 рис. 1), адреса которых занесены в его локальную оперативную память (п. 4 рис. 2).

4. Сравнивает подсчитанные значения хеш-функций для блоков оперативной памяти компьютера (п. 2 рис. 1) с соответствующими значениями, хранящимися в его локальной оперативной памяти (п. 4 рис. 2).

5. При несовпадении вышеупомянутых значений хеш-функций передает операционной системе заявку на восстановление исходного блока команд в оперативной памяти компьютера. Процесс восстановления исходного блока команд описан ниже.

6. Продолжает контролировать оперативную память компьютера (пп. 3-5).

Для подсчета хеш-функции могут использоваться различные алгоритмы хеширования, от простейшего контрольного суммирования (суммы всех четырех- или восьмибайтовых слов с добавлением разряда переполнения к младшему биту) до применения более сложных и трудоемких известных алгоритмов, например распространенных SHA-1 и SHA. Специализированный процессор аппаратно реализует алгоритмы хеширования. Быстродействие специализированного процессора зависит от сложности используемого алгоритма. В экспериментальном варианте был использован алгоритм контрольного суммирования.

Программное обеспечение работы антивируса заключается в изменении ряда сервисных программ, в частности транслятора, редактора связей, стандартных библиотек и операционной системы. Транслятор кроме объектного кода программы должен дополнительно создать теги начала и конца каждого блока команд. Эти теги будут обработаны редактором связей для всех программных модулей и представлены загрузчику операционной системы для передачи их антивирусному сопроцессору. Загрузчик операционной системы обладает способностью загружать не только программу в целом, но и ее отдельные фрагменты (последовательности команд, соответствующие паре тегов начала и конца фрагмента). Программная часть способа заключается в следующей последовательности действий:

1. При компиляции программы компилятор устанавливает теги, соответству-

ющие началу и концу последовательности исполняемых операторов (блоку программного кода). В остальном компиляция программ ничем не отличается от стандартной процедуры. В классах объектно-ориентированных программ теги устанавливаются для всех методов класса.

2. При загрузке объектного кода программы в оперативную память компьютера антивирусный сопроцессор вычисляет быстрые хеш-функции для каждого блока программы в оперативной памяти. В момент загрузки известны фактические адреса оперативной памяти, ограничивающие блоки команд. Антивирусный сопроцессор сохраняет хеш-функцию для всех блоков адресов программы в оперативной памяти.

3. Параллельно с работой центрального процессора антивирусный сопроцессор вычисляет хеш-функции всех программных блоков. При несовпадении с ранее подсчитанной хеш-функцией блок считается испорченным. После этого программа приостанавливается и проводится повторная загрузка блока.

Экспериментальная проверка антивирусной защиты

Для проверки работоспособности предложенного способа антивирусной защиты был проведен эксперимент. Он включал разработку антивирусного сопроцессора на базе ПЛИС Altera 28 nmStratix® V.

В рамках учебного проекта студенты МГТУ «СТАНКИН» создали схему и обеспечили ее встраивание в материнскую плату персонального компьютера. В качестве основы для программного обеспечения были выбраны операционная система LINUX и соответствующие сервисные программы. Основным критерием выбора являлось наличие свободной лицензии на использование и открытого программного кода, что позволило провести модификацию с минимальными затратами. Программа-компилятор с языка С++ была дополнена блоком, который в текст объектного модуля вставлял теги начала и конца фрагмента скомпилированного кода, ограничивая исполняемый фрагмент кода тегами.

В операционной системе были сделаны доработки, в частности в загрузчике исполняемых кодов. Эти изменения определяли фактические адреса тегов начала и конца фрагмента исполняемого кода и фиксировали соответствующие этим тегам адреса оперативной памяти компьютера. При фиксации фрагментов адреса тегов начала и конца всех фрагментов задачи передавались в антивирусный сопроцессор, который составлял таблицу. При составлении таблицы все фрагменты задачи имели в первой колонке один и тот же номер. Порядковый номер задачи определялся антивирусным сопроцессором. Для каждого фрагмента кода вычисляется хеш-функция и помещается в соответствующее поле таблицы.

При завершении задачи операционная система передавала в антивирусный сопроцессор адрес начала любого своего фрагмента. По этому адресу антивирусный сопроцессор определяет номер задачи и удаляет из таблицы все строки с этим номером.

Эксперимент по проверке антивирусного средства включал разработку программы эксперимента. Блок-схема программы приведена на рис. 3.

Проверка работы антивирусного средства состоит из двух параллельно функционирующих программных компонентов. Первый компонент (изображен на рисунке слева) имитирует появление вируса в программном обеспечении. Имитация работы вируса состоит в том, что по случайному адресу, принадлежащему блоку адресов исполняемого кода программы, изменяется содержимое одного из адресов. В данной конкретной программе таким проявлением вируса является инкрементация содержимого этого адреса. Отметим, что важно не значение изменения, а само изменение, поскольку оно имитирует работу вируса. Внесенное изменение фиксируется в параллельно работающем на антивирусном сопроцессоре программном средстве. Инкрементация одного из контролируемых адресов приведет к отличию хеш-функции от эталона, что является признаком вируса.

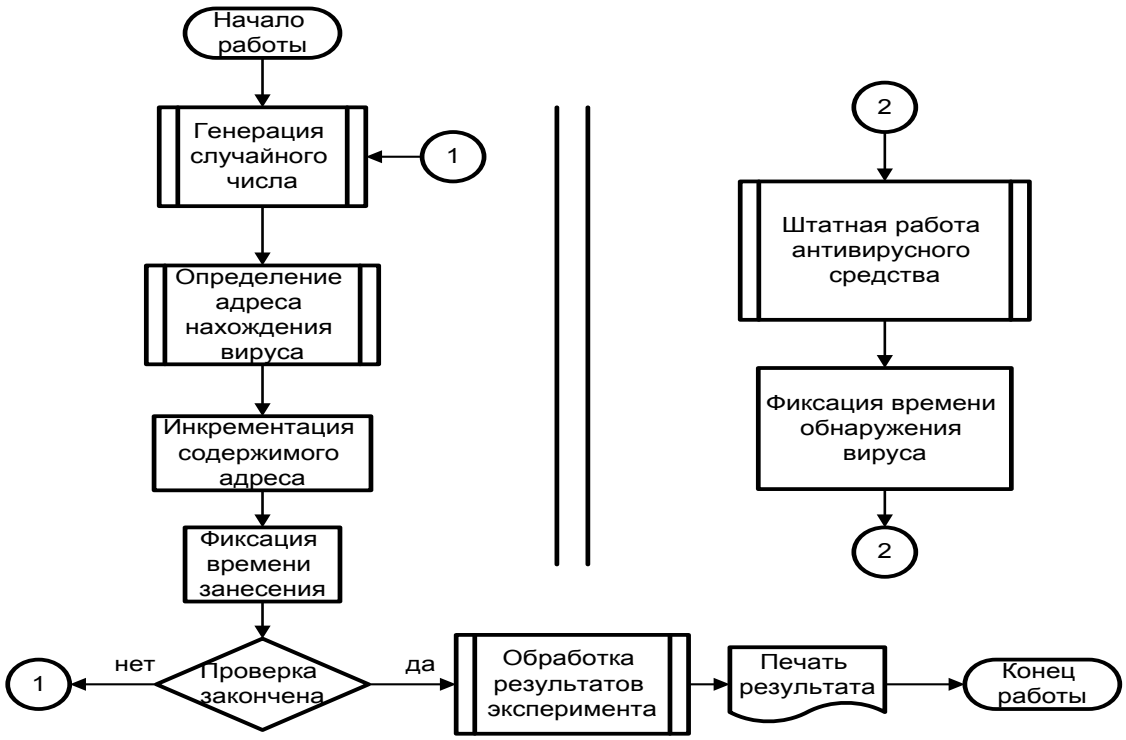

Рис. 3. Блок-схема проведения экспериментальной проверки антивирусного средства

Разница во времени между моментом изменения содержимого контролируемого адреса и временем фиксации вируса определяет реактивность антивирусного средства [6]. При увеличении количества задач или их имитаторов нагрузка на антивирусное средство будет возрастать, что вполне закономерно показал эксперимент [9-11]. Проведенная проверка показала высокую эффективность работы антивирусного средства.

Заключение

Разработана структура системы антивирусной защиты. Антивирусная защита включает аппаратную часть и программные средства. Аппаратная часть состоит из антивирусного сопроцессора, который подключен к системной шине.

Антивирусный сопроцессор определяет начало, конец блока исполняемого кода программы и рассчитывает хеш-функцию кодов программы.

Программные средства системы антивирусной защиты включают компиляторы, компоновщики, загрузчики и фрагменты операционной системы.

Разработанная система антивирусной защиты не использует базу сигнатур вирусов и может обнаруживать любые изменения в кодах программы.

Описана компьютерная модель системы антивирусной защиты.

СПИСОК ЛИТЕРАТУРЫ

1. Кабак, И.С. Аппаратная реализация ассоциативной памяти произвольного размера / И.С. Кабак, Н.В. Суханова // Вестник МГТУ «Станкин». - 2010. - № 1. - С. 135-139.
2. Кабак, И.С. Методика применения аппарата нейронных сетей для решения задач диагностики процесса резания / И.С. Кабак, Н.В. Суханова, А.М. Гаделев // Вестник МГТУ «Станкин». - 2012. - № 4 (23). - С. 130-133.
3. Кабак, И.С. Технология реализации автоматизированных систем управления на базе больших искусственных нейронных сетей МОДУС-НС / И.С. Кабак, Н.В. Суханова // Межотраслевая информационная служба. - 2012. - № 4. - С. 43-47.
4. Кабак, И.С. Модульное построение систем числового программного управления / И.С. Кабак, Н.В. Суханова, М.А. Григорьев // Системы проектирования, технологической под-

готовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM - 2010): тр. междунар. конф. - 2010. - С. 245-248.
5. «Лаборатория Касперского» выпускает новую версию AVP Inspector. - URL: http://www.kaspersky.ru/about/news/product/2000/Laboratoriya_Kasperskogo_vyipuskaet_novuyu_versiyu_AVP_Inspector.
6. Соломенцев, Ю.М. Повышение быстродействия суперкомпьютера за счет оптимизации информационного межпроцессорного трафика / Ю.М. Соломенцев, С.А. Шептунов, И.С. Кабак, Н.В. Суханова // Известия Кабардино-Балкарского государственного университета. - 2012. - Т. II. - № 4. - С. 71-73.
7. Шептунов, С.А. Технология МОДУС-НС в разработке высоконадежных и живучих систем управления техническими объектами / С.А. Шептунов, Ю.М. Соломенцев, И.С. Кабак, Н.В. Суханова // Вестник Брянского государственного технического университета. - 2014. - № 3. - С. 170-173.
8. CNNewsAnalitics. Методика подсчета потерь от компьютерных вирусов. - URL: http://www.cnews.ru/reviews/free/security/part1/eco_loss.shtml.
9. Sheptunov, S.A. Optimimization of the Complex Software Reliability of Control Systems / S.A. Sheptunov, M.V. Larionov, N.V. Sukhanova, I.S. Kabak, D.A. Alshinbaeva // IEEE Conference on Quality Management, Transport and Information Security, Information Technologies (IT&MQ&IS 2016) PROCEEDINGS. - 2016. - P. 225-228.
10. Sheptunov, S.A. Simulating of Reliability of Robotics System Software on Basis of Artificial Intelligence / S.A. Sheptunov, M.V. Larionov, N.V. Sukhanova, M.R. Salakhov, Y.M. Solomentsev, I.S. Kabak // IEEE Conference on Quality Management, Transport and Information Security, Information Technologies (IT&MQ&IS 2016) PROCEEDINGS. - 2016. - P. 220-224.
11. Solomentsev, Yu.M. Assessing the Reliability of CAD Software by Means of Neural Networks / Yu.M. Solomentsev, I.S. Kabak, N.V. Sukhanova // Russian Engineering Research. - 2015. - № 12.

1. Kabak, I.S. Hardware realization of associative memory of arbitrary dimension / I.S. Kabak, N.V. Sukhanova // *Bulletin of MSTU "Stankin"*. – 2010. – No. 1. – pp. 135-139.
2. Kabak, I.S. Procedure for application of neural networks apparatus to solve problems of cutting process diagnostics / I.S. Kabak, N.V. Sukhanova, A.M. Gadelev // *Bulletin of MSTU "Stankin"*. – 2012. – No.4 (23). – pp. 130-133.
3. Kabak, I.S. Technology of automated management systems realization based on large artificial neural networks MODUS-NS / I.S. Kabak, N.V. Sukhanova // *Inter-branch Information Service*. – 2012. – No.4. – pp. 43-47.
4. Kabak, I.S. Modular structure of numerical program control systems / I.S. Kabak, N.V. Sukhanova, M.A. Grigoriev // Systems of Design, Technological Pre-production and Management of Product Life Stages (CAD/CAM/PDM – 2010: *Proceedings of the Inter. Conf.* – 2010. – pp. 245-248.
5. "Kaspersky Laboratory" issues new version AVP Inspector. - URL: http://www.kaspersky.ru/about/news/product/2000/Laboratoriya_Kasperskogo_vyipuskaet_novuyu_versiyu_AVP_Inspector.
6. Solomentsev, Yu.M. Super-computer speed increase at expense of information inter-processor traffic / Yu.M. Solomentsev, S.A. Sheptunov, I.S. Kabak, N.V. Sukhanova // *Proceedings of Kabardino-Balkaria State University*. – 2012. – Vol.II. – No.4. – pp. 71-73.
7. Sheptunov, S.A. Technology MODUS-NS in development of trusted and enduring systems of technological objects management / S.A. Sheptunov, Yu.M. Solomentsev, I.S. Kabak, N.V. Sukhanova // *Bulletin of Bryansk State Technical University*. – 2014. – No.3. – pp. 170-173.
8. CNNewsAnalitics. *Procedure for Computation of Losses Caused by Computer Viruses*. - URL: http://www.cnews.ru/reviews/free/security/part1/eco_loss.shtml.
9. Sheptunov, S.A. Optimimization of the Complex Software Reliability of Control Systems / S.A. Sheptunov, M.V. Larionov, N.V. Sukhanova, I.S. Kabak, D.A. Alshinbaeva // IEEE Conference on Quality Management, Transport and Information Security, Information Technologies (IT&MQ&IS 2016) PROCEEDINGS. - 2016. - P. 225-228.
10. Sheptunov, S.A. Simulating of Reliability of Robotics System Software on Basis of Artificial Intelligence / S.A. Sheptunov, M.V. Larionov, N.V. Sukhanova, M.R. Salakhov, Y.M. Solomentsev, I.S. Kabak // IEEE Conference on Quality Management, Transport and Information Security, Information Technologies (IT&MQ&IS 2016) PROCEEDINGS. - 2016. - P. 220-224.
11. Solomentsev, Yu.M. Assessing the Reliability of CAD Software by Means of Neural Networks / Yu.M. Solomentsev, I.S. Kabak, N.V. Sukhanova // Russian Engineering Research. - 2015. - № 12.

Статья поступила в редколлегию 5.07.17.
Рецензент: д.т.н., профессор ИКТИ РАН
Куликов М.Ю.

Сведения об авторах:

Кабак Илья Самуилович, доктор технических наук, доцент, профессор кафедры «Компьютерные системы управления» ФГБО ВПО Московский государственный технологический университет «СТАНКИН» (Россия), e-mail: ikabak@mail.ru.
Суханова Наталия Вячеславовна, кандидат технических наук, доцент, доценткафедры «Компьютерные системы управления» ФГБО ВПО Московский государственный технологический университет «СТАНКИН» (Россия), e-mail: N_v_sukhanova@mail.ru.
Шептунов Сергей Александрович, доктор технических наук, профессор, директор ФГБУН Институт конструкторско-технологической информатики РАН (Россия), e-mail: ship.ikti@mail.ru.

Kabak Iliya Samuilovich, D. Eng., Assistant Prof., Prof. of the Dep. “Management Computer Systems”, Moscow State Technological University “STANKIN”, e-mail: ikabak@mail.ru.
Sukhanova Nataliya Vyacheslavovna, Can. Eng., Assistant Prof. of the Dep. “Management Computer System”, Moscow State Technological University “STANKIN”, e-mail: N_v_sukhanova@mail.ru.
Sheptunov Sergey Alexandrovich, D. Eng., Prof., Director of the Institute of Design-Technological Informatics of RAS, e-mail: ship.ikti@mail.ru.