УДК 004.492.2

ПОВЫШЕНИЕ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ЭТАПЕ ЕГО РАЗРАБОТКИ ПУТЕМ ПРОТИВОДЕЙСТВИЯ РУТКИТАМ РЕЖИМА ПОЛЬЗОВАТЕЛЯ В ОПЕРАЦИОННЫХ СИСТЕМАХ WINDOWS NT

Р. С. Цвентарный, Ю. В. Зилинский, А. В. Котовский
Кременчугский национальный университет имени Михаила Остроградского
ул. Первомайская, 20, 39600, г. Кременчуг, Украина. E-mail: ktp@kdu.edu.ua

На основе анализа механизмов, используемых руткитами режима пользователя, предложены методы и их программная реализация в системе защиты программного обеспечения с прозрачной стыковкой с приложением на этапе его разработки. Разработанная программная реализация предложенных методов в виде библиотеки динамической линковки операционной системы Windows NT может быть использована разработчиками программного обеспечения на стадии кодирования для противодействия многим известным на сегодня руткит-технологиям режима пользователя.

Ключевые слова: руткит, перехват функций, режим пользователя, Win API.

SOFTWARE RELIABILITY IMPROVING ON STAGE OF ITS DEVELOPMENT BY COUNTERACTION TO USER MODE ROOT-KITS IN THE WINDOWS NT OPERATING SYSTEMS

R. S. Tsventarniy, Yu. V. Zilinskiy, A. V. Kotovskiy
Kremenchuk Mykhailo Ostrohradskyi National University
vul. Pershotravneva, 20, 39600, Kremenchug, Ukraine. E-mail: ktp@kdu.edu.ua

Based on analysis of the mechanisms used by user-mode root-kits, methods and their software implementation in a system of software protection with clear connections with the application during development are suggested. The developed programmatic realization of the offered methods as the library of the dynamic linking operating system of Windows NT can be used by the developers of software on the stage of code for counteraction to many root-kits-technologies of the mode of user known for today.

Key words: root-kit, function calls intercept, user mode, Win API.

ПІДВИЩЕННЯ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ЕТАПІ ЙОГО РОЗРОБКИ ШЛЯХОМ ПРОТИДІЇ РУТКІТАМ РЕЖИМУ КОРИСТУВАЧА В ОПЕРАЦІЙНИХ СИСТЕМАХ WINDOWS NT

Р. С. Цвєнтарний, Ю. В. Зілінський, А. В. Котовський
Кременчуцький національний університет імені Михайла Остроградського
вул. Першотравнева, 20, 39600, м. Кременчук, Україна. E-mail: ktp@kdu.edu.ua

На основі аналізу механізмів, які використовують руткіти режиму користувача, запропоновані методи та їх програмна реалізація в системі захисту програмного забезпечення з прозорою стиковкою із додатком на етапі його розробки. Розроблена програмна реалізація пропонованих методів у вигляді бібліотеки динамічної лінковки операційної системи Windows NT може бути використана розробниками програмного забезпечення на стадії кодування для протидії багатьом відомим на сьогодні руткіт-технологіям режиму користувача.

Ключові слова: руткіт, перехоплення функцій, режим користувача, Win API.

АКТУАЛЬНОСТЬ РАБОТЫ. Стремительное в последнее время развитие информационных технологий, к сожалению, неразрывно связано с параллельным развитием сопутствующих неблагоприятных факторов. Одним из них является расширение круга угроз в области информационной безопасности, связанное, в частности, с распространением вредоносного программного обеспечения (malware).

Одной из основных причин появления подобных программ для конкретной операционной системы (ОС) является ее широкое распространение и наличие разнообразной и достаточно полной технической документации по системе.

По статистическим данным Интернет-ресурса StatCounter (http://gs.statcounter.com/#os-ww-monthly-200908-201009) в период с августа по сентябрь 2010 года более 90% обращений к примерно трем миллионам сайтов, размещенных на серверах по всему миру, производилось пользователями с компьютеров под управлением различных версий ОС семейства Windows NT. Учитывая этот факт, становится очевидным, что подавляющее большинство вредоносных программ разрабатывается на сегодняшний день для операционных систем именно этого семейства.

Аналитики информационной безопасности непрестанно фиксируют все новые действия злоумышленников, направленные на нарушение безопасности конфиденциальных данных пользователей, и наблюдают новые тенденции развития вредоносных программ. В частности, в последнее время отмечается возрастающая тенденция использования разработчиками вредоносного программного обеспечения для ОС семейства Windows NT разнообразных руткит-технологий режима пользователя (user-mode root-kit) для сокрытия от пользователя факта своего присутствия в системе и препятствия работе программных средств защиты.

Анализ известных авторам настоящей статьи литературных источников показывает, что хотя методы безопасного кодирования [1] и повышают надежность программного обеспечения на стадии его эксплуатации, но оказываются бессильными против используемых руткит-технологий [2]. В этом аспек-

те разработчику программного обеспечения отводится пассивная роль, и защищенность его программ зависит от наличия и возможностей используемых в системе средств защиты.

В связи с этим целью работы является разработка принципов построения и программная реализация системы защиты для противодействия руткитам режима пользователя операционной системы семейства Windows NT и повышения степени защищенности программного обеспечения на этапе его разработки путем прозрачной для разработчика стыковки его приложения с системой защиты.

МАТЕРИАЛ И РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ. Как свидетельствует опыт авторов, практическая реализация предложенного в [3] метода организации прямого взаимодействия программы с ядром ОС для противодействия руткит-технологиям режима пользователя сопровождается определенными трудностями.

Прежде всего, для реализации указанного метода к разработчику программного обеспечения предъявляются требования достаточно глубокого знания архитектуры операционной системы, навыков системного программирования, знания низкоуровневого языка программирования (ассемблера), умения работы со средствами отладки.

В связи с этим, авторами данной работы предлагается разработка и программная реализация методов прозрачной для разработчика программного обеспечения стыковки программного модуля с системой защиты, позволяющей повысить надежность программного обеспечения на этапе его разработки.

Предложенный метод базируется на использовании в системе защиты технологий, аналогичных вредоносным руткит-технологиям, однако направленных на противодействие им.

Система защиты построена на использовании нескольких методов перехвата функций программного интерфейса приложений (Win API) [2].

Первый метод перехвата заключается в модификации таблицы импорта (IAT) защищаемого приложения путем поиска и подмены адресов API-функций, использующихся приложением, которые наиболее часто подвергаются атакам со стороны руткитов. Адреса таких функций заменяются адресами адресного пространства программного модуля системы защиты. Алгоритм метода защиты:

1. Поиск точки входа нужной функции в таблице импорта.
2. Замена найденной точки входа адресом библиотеки системы защиты.
3. Подготовка и исполнение чистого кода функции.
4. Возврат управления в вызвавшее приложение.

Схема защиты путем модификации таблицы импорта изображена на рисунке 1.

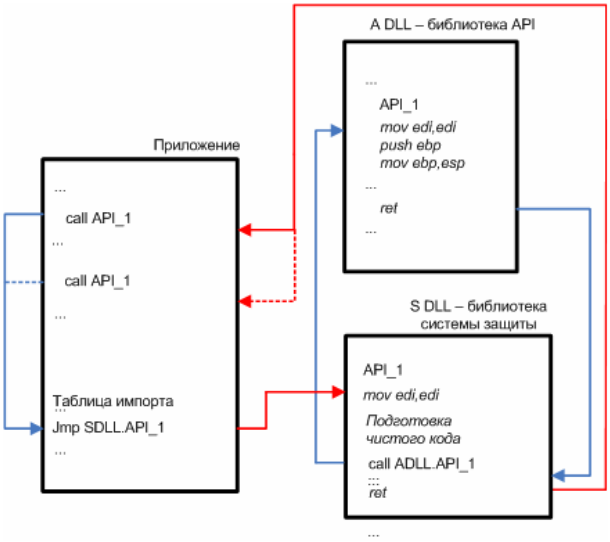

Рисунок 1 – Схема защиты путем модификации таблицы импорта

В адресном пространстве системы защиты предложено несколько способов безопасного, защищенного исполнения кода перехваченных функций. Наиболее простой из них – исполнение заведомо реализованной «чистой» копии кода перехваченной функции. Исследования показали, что такой способ защищенного исполнения кода имеет ряд недостатков: изменение исходного кода перехватываемой функции вследствие обновления компонентов ОС; наличие в коде функции вызовов других функций.

Первый недостаток не существенен в том случае, если компоненты ОС не претерпевают изменений.

Учет второго недостатка произведен при реализации второго метода перехвата, о котором речь пойдет ниже, а также другого способа защищенного исполнения кода перехваченной функции в рамках рассматриваемого метода перехвата функций. Этот способ использует метод организации прямого взаимодействия модуля защиты с ядром ОС, который описан в [3], и исключает возможность перехвата функции руткитом режима пользователя на этом этапе работы системы защиты.

Другой способ защищенного исполнения кода перехваченных функций базируется на обязательном перехвате и защите функций библиотеки kernel32.dll LoadLibrary (динамическая загрузка библиотеки) и GetProcAddress (получение адреса функции), а также перехвате используемых этими функциями в самой kernel32.dll функций LdrLoadDll и LdrGetProcedureAddress, импортируемых kernel32.dll из ntdll.dll. Этот перехват осуществляется путем внедрения кода команды перехода в начале указанных функций (сплайсинг) [2]. Способ позволяет динамически загрузить «чистую» библиотеку и получить в ней адрес функции, вызов которой предпринимает защищаемая программа. Рассматриваемый способ имеет существенное преимущество: он позволяет заблокировать динамическую загрузку посторонних библиотек (позднее связывание). В таком случае становится невозможным один из наиболее популярных методов внедрения руткита, исполненного в виде динамической библиотеки (DLL), с помощью запуска удаленного потока. Этот метод

внедрения известен также под названием инжект (inject). Однако при этом необходимо учитывать, что защищаемая программа сама может использовать механизм позднего связывания.

Второй метод перехвата учитывает то, что руткит может осуществлять более глубокий перехват в таблицах импорта системных библиотек, таких как kernel32.dll. Способы защищенного исполнения кода перехваченных функций при этом аналогичны описанным при рассмотрении первого метода перехвата. Алгоритм метода защиты:

1. Поиск всех загруженных модулей.
2. Для каждого найденного модуля применение алгоритм первого метода защиты.

Третий метод перехвата использует технологию внедрения машинного кода в перехватываемые функции (сплайсинг). Алгоритм метода защиты:

1. Подготовка буфера кода для команд push и ret.
2. Копирование первых 6 байтов кода оригинальной функции.
3. Замещение первых 6 байтов кода оригинальной функции кодом подготовленного в п. 1 буфера.

Схема защиты путем внедрения машинного кода изображена на рисунке 2.

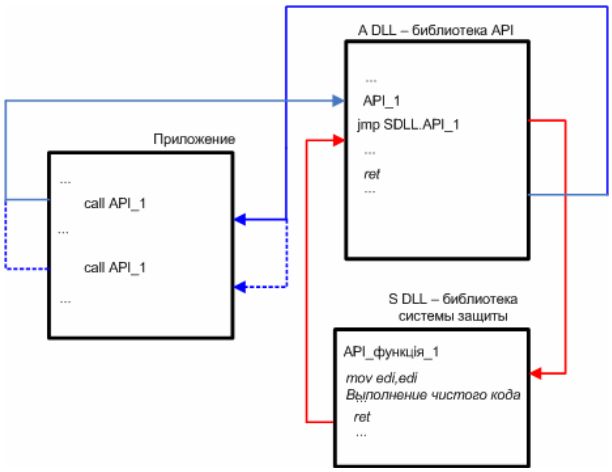

Рисунок 2 – Схема защиты путем внедрения машинного кода

Следует отметить, что описанные методы защиты от руткитов режима пользователя не лишены недостатка. Он заключается в том, что на момент запуска защищаемого приложения руткит уже может присутствовать в системе и, например, контролировать создание всех процессов или вообще не создавать отдельных потоков для своей работы. При этом не исключается возможность успешного противодействия описанной системе защиты со стороны такого руткита. Однако авторами ведется исследование методов самоконтроля приложений и обнаружения ими руткитов режима пользователя в процессе эксплуатации программного обеспечения с применением методов, описание которых выходит за рамки настоящей публикации, но будет дано в последующих работах.

Несмотря на это, авторами предлагается несколько иное решение указанной уязвимости системы защиты при условии, что приложение использует интерфейс Win API. Вместо стандартных названий функций системных библиотек можно использовать их псевдонимы или добавлять суффиксы или префиксы в стандартные названия функций. Преимуществом такого решения является то, что псевдонимы неизвестны разработчикам руткитов, ориентированных на определенный набор важных функций, и, соответственно, перехват функций приложения в таком ракурсе становится невозможным. Реализация же всех стандартных функций возлагается на систему защиты, которая может их исполнить, используя метод организации прямого взаимодействия с ядром ОС [3].

Описанная система защиты реализована в виде динамической библиотеки (DLL), с которой необходимо выполнить статическое (раннее) связывание. Остальную функциональность обеспечит код самой библиотеки.

ВЫВОДЫ. На основе анализа механизмов вызова функций программного интерфейса операционной системы Windows NT предложены методы самозащиты программ от вредоносного программного обеспечения, использующего руткит-технологии режима пользователя.

Предложенные методы применяются на этапе разработки программного обеспечения путем прозрачной для разработчика стыковки с системой защиты и не требуют от разработчика специальных знаний в области системного программирования.

ЛИТЕРАТУРА

1. Ховард М., Лебланк Д. Защищенный код / Пер. с англ. – М.: Издательско-торговый дом "Русская Редакция", 2004. − 704 с.

2. Зайцев О.В. Rootkits, Spyware/Adware, Keyloggers&Backdoors: обнаружение и защита. – СПб.: БХВ-Петербург, 2006. – 304 с.

3. Зілінський Ю.В., Бельська В.Ю., Юдіна А.Л. Захист і оптимізація програмного забезпечення шляхом прямих викликів сервісів ядра операційних систем Windows NT // Вісник Кременчуцького державного політехнічного університету імені Михайла Остроградського. – Кременчук: КДПУ, 2009. – Вип. 5/2009 (58), ч. 1. – С. 49–53.

REFERENCE

1. Howard M., LeBlanc D. Writing secure code. – Microsoft Corporation, 2002. – 800 p. [in Russian].

2. Zaycev O.V. Rootkits, Spyware/Adware, Keyloggers&Backdoors: detection and protection. – SPb.: BHV-Petersburg, 2006. – 304 p. [in Russian].

3. Zilinskiy U.V., Belska V.U., Udina A.L. Software protection and optimization by direct system services calls in Windows NT operating system // Transactions of the Kremenchuk Mykhailo Ostrohradskiy state polytechnic university. – Kremenchuk: KSPU, 2009. – Ed. 5/2009 (58), V. 1. – P. 49–53 [in Ukrainian].

Стаття надійшла 20.01.2011.
Рекомендована до друку
д.т.н., проф. Гученком М.І.