РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ

Левыкин Михаил Владимирович

МОДЕЛИ И СРЕДСТВА ВЫЯВЛЕНИЯ УГРОЗ НАРУШЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ШТАТНЫХ МЕХАНИЗМОВ ОБНАРУЖЕНИЯ СКРЫТЫХ ИНФОРМАЦИОННЫХ ВОЗДЕЙСТВИЙ В ЯДРЕ ОС WINDOWS

Специальность: 05.13.19 – методы и системы защиты информации, информационная безопасность

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Москва – 2010

Работа выполнена на кафедре Компьютерной Безопасности Российского государственного гуманитарного университета

Научный руководитель:	д.ф.-м.н., профессор А.А. Грушо
Официальные оппоненты:	д.т.н., профессор С.Н. Смирнов, к.т.н. В.Б. Нетыкшо
Ведущая организация:	Институт проблем информационной безопасности, МГУ имени М.В. Ломоносова

Защита состоится 27 сентября 2010 г. в 14 часов на заседании диссертационного совета Д 212.198.13 при Российском государственном гуманитарном университете по адресу: г. Москва, Миусская пл., д. 6.

С диссертацией можно ознакомиться в библиотеке Российского государственного гуманитарного университета.

Автореферат разослан 05 августа 2010 г.

Ученый секретарь
диссертационного совета,
к.т.н. ст. науч. сотр. Д.Б. Халяпин

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ.

Актуальность проблемы:

В настоящее время человечество вплотную подошло к рубежу, за которым начинается новый этап его развития, получивший название «информационного общества». Активная информатизация различных сфер общественных отношений приводит к необходимости решения задач, направленных на обеспечение информационной безопасности его национальных информационно-телекоммуникационных инфраструктур. Основополагающим документом, регламентирующим политику России в области информационной безопасности, является Доктрина информационной безопасности Российской Федерации, утвержденная в сентябре 2000 года Президентом Российской Федерации[1]. Реализация мероприятий по обеспечению информационной безопасности Российской Федерации, отнесенных Доктриной к первоочередным, невозможна без глубокой комплексной научной проработки стоящих на этом направлении задач. В Доктрине даны общие формулировки проблем, на решение которых должны быть ориентированы задачи, изложенные в Перечне научно-технических проблем обеспечения информационной безопасности Российской Федерации (физико-математические, технические). Среди технических проблем этого Перечня[2], следующие положения включают разработку методов обнаружения скрытых от используемых средств обеспечения безопасности деструктивных информационных воздействий.

«Разработка моделей угроз безопасности систем и способов их реализации, определение критериев уязвимости и устойчивости систем к деструктивным воздействиям, разработка методов и средств мониторинга для выявления фактов применения несанкционированных информационных воздействий, разработка методологии и методического аппарата оценки ущерба от воздействия угроз информационной безопасности» (п. 46 Перечня)

«Разработка методов и средств обеспечения информационной безопасности информационных и телекоммуникационных систем, в том числе автоматизированных систем управления безопасностью,

---

[1] Доктрина информационной безопасности-Российской Федерации. «Российская газета» (утв. Президентом РФ от 9 сентября 2000 г. N Пр-1895) N 187, от 28.09.2000.

[2] Перечень основных направлений и приоритетных проблем научных исследований в области информационной безопасности Российской Федерации.

методов и средств распределения ключей и защиты информации и информационных ресурсов от несанкционированного доступа и разрушающего информационного воздействия, антивирусных технологий, методов и средств контроля состояния защищенности от НСД современных и перспективных технических средств и каналов связи, решение проблемы гарантированного уничтожения остаточной информации на магнитных носителях, исследование и развитие методов построения защищенных систем, использующих ненадежные (с точки зрения информационной безопасности) элементы, включая проблему их тестирования»[3] (п. 48 Перечня).

Под скрытым информационным воздействием в настоящей работе понимается скрытое от комплекса средств обеспечения информационной безопасности несанкционированное воздействие на защищаемую информацию и (или) другие ресурсы защищаемой информационной системы (ИС). Под методом обнаружения скрытого информационного воздействия в операционной системе (ОС) понимается алгоритмическое обеспечение и программные механизмы, с помощью которых осуществляется противодействие скрытию от комплекса средств обеспечения информационной безопасности несанкционированного воздействия на ресурсы ИС. Следует отметить, что методы скрытия и обнаружения информационного воздействия в большинстве случаев близки по технологии их реализации с точностью до направленности действия. Для обнаружения информационного воздействия чаще всего средство защиты использует те же принципы, что и средство скрытия. Этот факт в значительной степени связан с тем обстоятельством, что и те, и другие средства в качестве инструментария используют одни и те же особенности и механизмы ядра ОС. В последнее время как со стороны иностранных авторов: Батлер Дж.[4], Рутковская Ж.[5], Велер Р.[6], Эриксон Дж.[7], Бланден Б.[8], так

---

[3] Шерстюк В.П. МГУ: научные исследования в области информационной безопасности. www.ipib.msu.ru/documents/Sherstuk_VP.doc

[4] Холунг Г., Батлер Дж.. Руткиты: внедрение в ядро Windows. СПб.: Питер, 2007. – 285 с.: ил.

[5] Rutkowska Joanna. Rootkits Detection on Windows Systems - October 2004, http://invisiblethings.org/papers/ITUndergroundd2004_Win_rtks_detection.ppt

[6] Vieler Ric. Professional rootkits. USA. 2007, Wiley Publishing, Inc. Indianapolis, Indiana, Published simultaneously in Canada.

[7] Erickson Jon. Hacking: The art of exploitation, 2nd edition. USA. 2008, No starch press, Inc. 555 De Haro Street, Suite 250, San Francisco, CA 94107.

и со стороны российских – Зайцев О.[9], Колисниченко Д.[10], большое внимание уделяется скрытым информационным воздействиям на уровне ядра ОС Windows. Особую роль в вопросах, связанных с использованием и изучением скрытых информационных воздействий, играют Интернет - источники.

Согласно информации, представленной на сайте одной из ведущих российских компаний в области защиты информации[11], в последние 10-15 лет наблюдается значительный рост числа несанкционированных информационных воздействий деструктивного характера на уровне ядра ОС Windows одной из самых распространенных ОС в мире[12]. Это обстоятельство обусловлено многими практическими работами в этой области, их доступностью и относительной простотой в использовании.

Следует отметить тот факт, что значительная часть современных средств защиты информации (СЗИ), в частности антивирусные программные продукты, направлены на обнаружение скрытых информационных воздействий на уровне ядра ОС Windows. Более того, часто утверждается, что многие из них используют скрытые информационные воздействия для обеспечения самозащиты и для контроля подобных скрытых информационных воздействий[13].

Учитывая рост интереса к скрытым информационным воздействиям, разработки современных ОС семейства Windows предлагают для разработчиков СЗИ интегрированные в ОС механизмы выявления таких воздействий на уровне ядра. Такие механизмы представлены практически во всех подсистемах ядра ОС Windows:

---

[8] Blunden Bill. The rootkit arsenal. USA. 2009, Wordware Publishing, Inc. 1100 Summit Ave., Suite 102 Plano, Texas 75074.

[9] Зайцев О.В. ROOTKITS, SPYWARE/ADWARE, KEYLOGGER &BACKDOORS: обнаружения и защита. – СПб.: БХВ-Петербург, 2006. – 304 с.: ил.

[10] Колисниченко Д.Н. Rootkits под Windows. Теория и практика программирования «шапок-неведимок», позволяющих скрывать от системы данные, процессы, сетевые соединения. – СПб.: Наука и Техника, 2006. - 320 с.: ил.

[11] Securelist www.securelist.com

[12] Top Operating System Share Trend http://www.netmarketshare.com/os-market-share.aspx?qprid=9

[13] McMillan Robert. Symantec, Kaspersky Criticized for Cloaking SoftwareCompanies are accused of using rootkit-like techniques to hide information from users. Jan 13, 2006, http://www.pcworld.com/article/124365/symantec_kaspersky_criticized_for_cloaking_software.html

системном реестре[14]; файловой системе[15]; сетевой подсистеме; контроле процессов и потоков[16] и т.д. Операционные системы семейства Windows предлагают программный интерфейс уровня ядра[17] для реализации на его основе средств обнаружения скрытых информационных воздействий. С учетом широкого использования ОС семейства Windows, а также применения руткитов[18], растет популярность этого программного интерфейса.

Во многих зарубежных источниках представлен частичный анализ защищенности штатных механизмов обнаружения скрытых информационных воздействий на уровне ядра ОС Windows, а также уязвимостей в них. Однако проблема в настоящее время не разрешена. В этой области еще много нерешенных задач. Результаты исследования некоторых из них представлены в данной диссертационной работе.

Целью диссертационной работы является разработка методов и средств устранения скрытых информационных воздействий на уровне ядра ОС Windows на основе анализа штатных механизмов их обнаружения и определения уязвимостей.

В соответствии с поставленной целью в диссертационной работе **решаются следующие задачи**:

- анализ штатных механизмов обнаружения скрытых информационных воздействий в ОС Windows и разработка их логических моделей для поиска уязвимостей в этих механизмах;
- разработка инструментальных средств устранения скрытых информационных воздействий на уровне ядра ОС;

---

[14] CmRegisterCallback. Windows Driver Kit: Kernel-Mode Driver Architecture. Built on November 19, 2009 http://msdn.microsoft.com/en-us/library/aa906439.aspx

[15] FsRtlRegisterFileSystemFilterCallbacks. Windows Driver Kit: Installable File System Drivers. Built on November 23, 2009 http://msdn.microsoft.com/en-us/library/ms795401.aspx

[16] PsSetCreateProcessNotifyRoutine. Windows Driver Kit: Kernel-Mode Driver Architecture. Built on November 19, 2009. http://msdn.microsoft.com/en-us/library/ms802952.aspx

[17] Под программным интерфейсом уровня ядра понимается набор штатных механизмов выявления скрытых информационных воздействий на уровне ядра ОС Windows

[18] Холунг Г., Батлер Дж.. Руткиты: внедрение в ядро Windows. СПб.: Питер, 2007. – 285 с.: ил.

- экспериментальное обоснование существования уязвимостей в штатных механизмах обнаружения скрытых информационных воздействий на примере ОС Windows XP.

Объектом настоящего исследования являются штатные механизмы обнаружения скрытых информационных воздействий на уровне ядра ОС Windows. **Предметом исследования** являются уязвимости штатных механизмов обнаружения скрытых информационных воздействий на уровне ядра ОС Windows.

В ходе проведения исследований применялись следующие **методы**: теория графов; математическое моделирование средств защиты информации в компьютерных системах; системный анализ; анализ алгоритмов; дизассемблирование; отладка; реинжиниринг[19]; реверсинг[20].

Научная новизна работы характеризуется тем, что в результате ее выполнения разработан метод анализа штатных механизмов обнаружения скрытых информационных воздействий на уровне ядра ОС Windows, с использованием которого найдены новые уязвимости в рассмотренных механизмах.

На защиту выносятся следующие основные результаты:

1. Метод анализа штатных механизмов обнаружения скрытых информационных воздействий на уровне ядра ОС Windows, основанный на локализации закрытого программного кода, реализующего эти механизмы;
2. Решение задачи восстановления логической схемы штатных механизмов управления доступом к реестру, контроля создания и удаления процессов, фильтрации сетевого трафика, полученное на основе анализа этих механизмов предложенным автором методом;
3. Экспериментальное обоснование выводов и положений теоретических исследований по поиску

[19] Восстановление исходного алгоритма и его модели на основе анализа исполняемого кода.

[20] Восстановление исходного кода на основе анализа исполняемого кода.

уязвимостей в штатных механизмах защиты ядра ОС Windows.

Практическая значимость исследования определяется тем, что его результаты позволили показать отсутствие должного уровня защищенности самих штатных механизмов. Найдены новые уязвимости в рассматриваемых механизмах, существование которых обосновано экспериментально. Использование таких уязвимостей может повлечь за собою уязвимости в СЗИ, построенных на их основе. Вместе с тем, контролируя найденные уязвимости можно повысить надежность СЗИ, построенных на их основе. Как следствие, результаты работы могут быть использованы для повышения надежности СЗИ, опирающихся на ненадежные с точки зрения безопасности штатные механизмы защиты информации в ОС Windows.

Теоретическая значимость настоящего исследования состоит в разработке нового метода анализа штатных механизмов обнаружения скрытых информационных воздействиях на уровне ядра ОС Windows и выявления с его помощью новых уязвимостей этих механизмов.

Внедрение и апробация результатов исследований.

Результаты диссертации использованы:

1) в ОАО «Газпром промгаз» при создании испытательного стенда, предназначенного для оценки эффективности обнаружения программных закладок уровня ядра (скрытых информационных воздействий) антивирусным программным обеспечением, используемым в ОАО «Газпром промгаз»;
2) в учебном процессе РГГУ при проведении лабораторных работ по курсу Вычислительные сети ИИНиТБ ФЗИ для ознакомления студентов
   - с устройством сетевой архитектуры ОС Windows XP и интерфейсом сетевого программирования на уровне ядра ОС Windows XP;
   - с механизмами использования скрытых каналов по памяти в стеке протоколов TCP/IP для передачи информации в обход штатных средств фильтрации сетевого трафика ОС Windows XP.

Результаты внедрения подтверждены двумя актами:

1) ОАО «Газпром промгаз»
2) РГГУ ИИНиТБ ФЗИ КБ.

Результаты диссертации докладывались на семинарах кафедры Компьютерной Безопасности ИИНиТБ РГГУ.

Результаты диссертационного исследования прошли апробацию на следующих международных конференциях:

1) XXXIV Международная конференция и дискуссионный научный клуб «Информационные технологии в науке, образовании, телекоммуникациях и бизнесе (весенняя сессия)», Украина, Крым, Ялта-Гурзуф, 2007 г.
2) XXXV Международная конференция и дискуссионный научный клуб «Информационные технологии в науке, образовании, телекоммуникациях и бизнесе (весенняя сессия)», Украина, Крым, Ялта-Гурзуф, 2008 г.
3) XXXVII Международная конференция и дискуссионный научный клуб «Информационные технологии в науке, образовании, телекоммуникациях и бизнесе (осенняя сессия)», Украина, Крым, Ялта-Гурзуф, 2009 г.

Публикации: Основные положения диссертационной работы опубликованы в 5-ти научных статьях, в том числе две из них опубликованы в журнале, включенном ВАК РФ в перечень ведущих рецензируемых научных журналов и изданий.

Структура и объем работы.

Диссертация состоит из введения, трех глав, заключения, списка литературы из 61 наименований и приложения. Основная часть работы изложена на 120 страницах с вычислительными примерами, таблицами, рисунками, листингами и исходными текстами программ.

СОДЕРЖАНИЕ РАБОТЫ

В первой главе обозначен тип рассматриваемых в дальнейшей работе скрытых информационных воздействий, представлены его определения и классификация. Описаны методы обнаружения скрытых информационных воздействий, к которым относятся методы перехвата, обратного вызова, фильтрации. Анализируются способы их реализации в ОС Windows. Представлен краткий обзор архитектуры ядра ОС Windows, а также штатных механизмов обнаружения скрытых информационных воздействий на уровне ядра. Дано обоснование актуальности и новизны задач, решенных в диссертационной работе.

Под скрытыми информационными воздействиями на уровне ядра ОС Windows понимаются руткиты этого уровня - вредоносные программы (набор программ), предназначенные для осуществления скрытого от СЗИ несанкционированного воздействия на ресурсы информационной системы, подлежащие защите, с целью скрытия от СЗИ деятельности других вредоносных программ и самой себя. Под методом обнаружения скрытого информационного воздействия в ОС понимается – логическое обеспечение и программные механизмы, с помощью которых осуществляется противодействие скрытию от комплекса средств обеспечения информационной безопасности несанкционированного воздействия на ресурсы защищаемой информационной системы. В работе представлено описание методов обнаружения скрытых информационных воздействий, основанных на перехвате системных вызовов, обратном вызове и фильтрации.

Под штатным механизмом обнаружения скрытых информационных воздействий понимается реализованный программными средствами ядра ОС Windows алгоритм обнаружения таких воздействий. Штатный механизм обнаружения скрытых информационных воздействий реализован как интерфейс программирования (определенный набор функции), используя который проектировщики и разработчики СЗИ могут создавать средства защиты от подобных воздействий.

В диссертационной работе рассмотрены следующие штатные механизмы.

1. Механизм обнаружения скрытых информационных воздействий, контролирующий доступ к системному реестру на уровне ядра ОС. Данный механизм, представлен следующими функциями ОС Windows:

CmRegisterCallback[21] /CmUnregisterCallback[22] – для ОС Windows XP; их расширениями CmRegisterCallbackEx[23] для ОС Windows Vista и более поздних ОС семейства NT. Этот механизм реализует метод обратного вызова, то есть регистрацию драйвера, который будет вызываться при обращениях к системному реестру на уровне исполнительной системы[24] ядра, то есть диспетчера конфигурации.

2. Механизм обнаружения скрытых информационных воздействий, контролирующий создание и удаления процессов, потоков, загрузку образов на уровне ядра ОС, который представлен следующими функциями ОС Windows: PsSetCreateProcessNotifyRoutine[25] - для ОС Windows XP, PsSetCreateProcessNotifyRoutineEx[26] – для ОС Windows Vista и более поздних ОС семейства NT. Так же как и механизм контроля реестра, данный механизм реализует метод обратного вызова регистрирует функции, которые будут вызываться при создании и удалении процессов.
3. Механизм обнаружения скрытых информационных воздействий, контролирующий доступ к ресурсам сети (фильтрацию сетевых пакетов) на уровне ядра ОС (штатный межсетевой экран). Данный механизм реализуется сервисом Брандмауэр Windows в виде интерфейса программирования, доступного как на уровне пользователя, так и на уровне ядра. Основная функция этого механизма фильтрации сетевого трафика[27].

---

[21] CmRegisterCallback http://msdn.microsoft.com/en-us/library/ff541918%28v=VS.85%29.aspx

[22] CmUnRegisterCallback http://msdn.microsoft.com/en-us/library/ff541928%28v=VS.85%29.aspx

[23] CmRegisterCallbackEx http://msdn.microsoft.com/en-us/library/ff541921%28v=VS.85%29.aspx

[24] Руссинович М., Соломон Д.. Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP, Windows 2000. Мастер-класс. / Пер. с англ. – 4-е изд. – М.: Издательство «Русская редакция»; СПб.: Питер, 2006. 992 стр.: ил.

[25] PsSetCreateProcessNotifyRoutine http://msdn.microsoft.com/en-us/library/ff559951%28VS.85%29.aspx

[26] PsSetCreateProcessNotifyRoutineEx http://msdn.microsoft.com/en-us/library/ff559953%28VS.85%29.aspx

[27] Windows Firewall Interfaces http://msdn.microsoft.com/en-us/library/aa366449%28v=VS.85%29.aspx

Во второй главе представлено описание разработанного автором метода анализа механизмов уровня ядра ОС Windows. На основе проведенного анализа каждого из штатных механизмов восстановлена его логическая схема, позволяющая дать теоретическое обоснование новых уязвимостей для рассматриваемого механизма. Результаты представленного в первой главе диссертации анализа штатных механизмов не позволяет обосновать защищенность самих механизмов. С использованием предложенного в работе метода анализа удалось восстановить логические схемы исследуемых механизмов. В свою очередь, это позволило выявить новые угрозы в рассмотренных механизмах.

На основании предложенного метода в условиях закрытого исходного кода анализируемой ОС удалось установить, что интересующие механизмы реализованы на уровне исполнительной системы и каждый из них контролирует доступ к одному из компонентов исполнительной системы. Следовательно, создав средство защиты, основанное на одном из штатных механизмов, можно с его помощью контролировать доступ к компоненту исполнительной системы, точнее, к защищаемому объекту в этом компоненте.

С помощью штатного механизма можно создать средство обеспечения безопасности объекта на уровне ядра, например ключа реестра. Используя интерфейс системных вызовов можно создать средство осуществляющее попытку доступа к защищаемому объекту ядра. Средство защиты, построенное на основе штатного механизма, обнаружит эту попытку и воспрепятствует ей. В момент обнаружения попытки доступа контекст исполнения данной задачи находится на уровне ядра ОС, а именно – в самом средстве защиты. В этот момент сегмент стека хранит в себе всю цепочку вызовов с уровня интерфейса системных вызовов (используемого при создании средства, осуществляющего доступ к защищаемому объекту) до штатного механизма (используемого при создании средства защиты рассматриваемого объекта). Таким образом, в момент попытки доступа к защищаемому объекту, инициированной испытателем, средство защиты, основанное на штатном механизме, имеет доступ ко всей цепочке вызовов функции начиная со средства нападения и заканчивая средством защиты. Взяв за вершины функции, вызываемые в рассматриваемой цепочке, а их дуги за вызовы, можно построить граф, описывающий работу данного штатного механизма. Этот граф позволяет определить и локализовать участок закрытого программного кода, отвечающий за работу штатного механизма, что, в свою очередь,

дает возможность восстановить логическую схему работы анализируемого механизма.

В диссертационной работе рассмотрен механизм обратного вызова реестра. Логическая схема данного механизма восстановлена с помощью предложенного в работе метода анализа. Согласно этой схеме существует массив CmpCallBackVector, в котором хранятся адреса обработчиков – функции обратного вызова работы с реестром. В переменной CmpCallBackCount хранится количество установленных обработчиков. В ходе выполнения системного вызова – функции работы с системным реестром уровня ядра, например NtDeleteKey, вызывается функция CmpCallCallBacks, которая обращается к этому массиву и поочередно вызывает все установленные обработчики. Таким образом, имея сведенья о массиве обработчиков и количестве элементов в нем, можно контролировать механизм обратного вызова реестра, который, в свою очередь, позволяет осуществлять контроль доступа к реестру на уровне диспетчера конфигурации.

Рассмотрим следующую ситуацию. Любой модуль уровня ядра имеет доступ во все адресное пространство на уровне ядра. Соответственно, он имеет доступ к массиву, содержащему в себе все адреса обработчиков. Находя в памяти этот массив, модуль ядра может обнулить его и переменную, в которой хранится количество установленных в системе обработчиков – функций обратного вызова реестра. Тогда функция CmpCallBackVector будет вызвана в процессе выполнения системного сервиса взаимодействия с реестром, однако прочитав значение количества установленных обработчиков, которое после изменения равно нулю, она вернет управление системному вызову. Как следствие, контролируя массив обработчиков и переменную счетчика обработчиков, можно контролировать сами обработчики, а также средства обнаружения скрытых информационных воздействий, основанные на данном штатном механизме уровня ядра ОС.

При построении средств контроля доступа к реестру, основанных только на рассматриваемом механизме, нельзя гарантировать контроль доступа, так как в логическую схему реализации данного механизма заложена уязвимость, которая связанна с отсутствием возможности контроля самого механизма. Однако, контролируя неэкспортируемые переменные ядра, необходимые для функционирования штатного механизма контроля доступа к реестру на уровне ядра ОС, а именно - CmpCallBackVector (массив обработчиков обратного вызова) и CmpCallBackCount (переменную, хранящую

количество установленных в системе обработчиков), данную уязвимость можно устранить.

Далее в работе рассмотрен штатный механизм контроля создания и удаления процессов. Логическая схема данного механизма, как и предыдущего, была восстановлена с помощью разработанного автором метода анализа. Она позволила убедиться в том, что при создании нового процесса подсистема Windows уведомляется о его создании. При этом поочередно вызываются несколько процедур. Адреса этих процедур, аналогично тому, как это реализуется механизмом контроля реестра, хранятся в массиве PspCreateProcessNotifyRoutine, а их количество в переменной PspCreateProcessNotifyRoutineCount. При этом сам массив PspCreateProcessNotifyRoutine скрыт от разработчика. Таким образом повлиять на работу остальных обработчиков или удалить их без самостоятельной предварительной установки нельзя.

Для того, чтобы установить свой собственный обработчик, необходимо использовать штатный механизм Windows – функция PsSetCreateProcessNotifyRoutine[28]. Первый параметр этой функции – адрес функции-обработчика. Второй - может принимать значения:

0 – для установки обработчика;

1 – для его удаления.

Таким образом, в случае необходимости создания монитора, отслеживающего порождение процессов, можно, используя данный механизм, установить обработчик (для отслеживания необходимых ему процессов) и самостоятельно его удалить.

Однако каждый модуль уровня ядра имеет доступ во все адресное пространства ядра. Как следствие, он имеет доступ к массиву, содержащему адреса обработчиков. Любой массив элементов - это адрес первого элемента массива (плюс смещение - тип каждого элемента). Определив адрес первого элемента (иначе говоря, самого массива PspCreateProcessNotifyRoutine), можно, переходя от элемента к элементу, найти все установленные в системе обработчики. А также можно обнулить элементы массива и переменную, хранящую в себе их количество. Таким образом можно удалить все обработчики, установленные средствами защиты. Результатом таких действий будет отключение контроля и аудита создания и удаления процессов.

При построении средств контроля создания и удаления процессов, основанных только на рассматриваемом механизме, нельзя

[28] PsSetCreateProcessNotifyRoutine http://msdn.microsoft.com/en-us/library/ms802952.aspx

гарантировать контроль за деревом процессов, так как в логическую схему реализации данного механизма заложена уязвимость, которая связанна с отсутствием возможности контроля самого механизма. Однако контролируя неэкспортируемые переменные ядра, необходимые для функционирования рассматриваемого механизма, а именно - PspCreateProcessNotifyRoutine (массив обработчиков обратного вызова) и PspCreateProcessNotifyRoutineCount (переменную, хранящую количество установленных в системе обработчиков), данную уязвимость можно устранить.

В диссертационной работе рассмотрен штатный механизм фильтрации сетевого трафика. Логическая схема механизма фильтрации сетевого трафика была восстановлена с помощью разработанного метода анализа. Она позволила показать, что дополнительные сервисы фильтрации пакетов и механизмы создания IP-ловушек, реализованные в штатном брандмауэре Windows (то есть механизмы защиты), находятся на уровне более высоком, чем NDIS-библиотека (интерфейс, реализующий сетевое взаимодействие). Анализ восстановленной схемы с использованием предложенного в работе метода показал, что создание NDIS-драйвера позволяет получать доступ ко всем сетевым пакетам до обработки этих пакетов брандмауэром Windows – штанным средством фильтрации сетевого трафика на уровне ядра ОС Windows. Таким образом, с помощью NDIS-библиотеки можно построить средство скрытой передачи сетевого трафика, так как пакеты, поступающие на данное средство, будут обрабатываться до того, как они будут обработаны средством фильтрации, построенным на основе рассматриваемого механизма.

Однако NDIS-библиотека также позволяет создавать средства фильтрации сетевого трафика с их первичной обработкой на уровне ядра ОС. Она предоставляет доступ ко всем сетевым фрагментам на канальном уровне (уровне доступа к среде передачи данных), что, в свою очередь, позволяет устранить уязвимость в этом штатном механизме.

В третьей главе описан разработанный автором экспериментальный стенд, позволивший: получить доступ к ядру ОС и его интерфейсам; экспериментально обосновать возможность использования разработанного автором метода анализа, основанного на локализации закрытого исходного кода уровня ядра ОС. Это позволило подтвердить справедливость выводов, изложенных во второй главе.

На разработанном стенде, для каждого из механизмов, рассматриваемых во второй главе, экспериментально обоснованны результаты их анализа. По каждому из рассмотренных механизмов проводились две серии экспериментов, итоговые результаты которых позволяют констатировать следующее:

- первая серия испытаний подтверждает возможность использования данного штатного механизма в целях обнаружения скрытых информационных воздействий, за счет создания на его основе средств защиты;
- вторая серия испытаний подтверждает существование уязвимости средств обнаружения скрытых информационных воздействий, построенных на базе данного штатного механизма.

Экспериментальное обоснование уязвимости в штатном механизме контроля реестра.

Первая серия испытаний заключалась в экспериментальном обосновании возможности использования механизма обратного вызова реестра, для контроля факта удаления ключей реестра. Для этого был создан драйвер, использующий рассматриваемый штатный механизм, который вел аудит удаления ключей реестра, а при попытке удалить защищаемый ключ - возвращал статус «отказано в доступе». В ходе данной серии экспериментов была осуществлена попытка удаления защищаемого ключа с помощью редактора реестра. Эксперименты с высокой степенью надежности показали, что возвращается статус «отказано в доступе».

Вторая серия испытаний заключалась в обнулении (отключении) массивов, отвечающих за работу штатного механизма контроля доступа к реестру, рассматриваемых во второй главе (CmpCallBackVector, CmpCallBackCount) и в повторении первой серии. В результате экспериментов редактор реестра удалил защищаемый ключ, что не протоколировалось с помощью используемого драйвера (средства защиты построенного на базе рассматриваемого штатного механизма). Данный факт подтверждает уязвимость средств защиты

контроля доступа к реестру, построенных на основе штатного механизма обратного вызова реестра.

Экспериментальное обоснование уязвимости в штатном механизме контроля создания и удаления процессов.

Первая серия испытаний заключается в экспериментальном обосновании возможности использования штатного механизма отслеживания создания и удаления процессов для обнаружения факта создания нового процесса и его удаления. Для проведения испытаний был создан драйвер, использующий рассматриваемый штатный механизм, который вел аудит состояния в ходе создания всех новых процессов и их удаления, представляя результаты такого аудита в виде отладочной информации. В результате серии испытаний осуществлялась попытка создания процесса Консоль Windows (cmd.exe) из оболочки системы (explorer.exe) и его удаление. Факт создания процесса протоколировался используемым драйвером.

Вторая серия заключалась в обнулении (отключении) массивов отвечающих за работу штатного механизма контроля создания и удаления процессов, рассматриваемых во второй главе (PspCreateProcessNotifyRoutine, PspCreateProcessNotifyRoutineCount) и в повторении первой серии экспериментов. В ходе этих экспериментов, процесс explorer.exe надежно создавал и удалял приложение cmd.exe, однако, этот факт не протоколировался с помощью используемого драйвера. Полученный результат подтверждает уязвимость средств защиты контроля создания и удаления процессов, построенных на базе рассматриваемого штатного механизма.

Экспериментальное обоснование уязвимости в штатном механизме фильтрации сетевого трафика.

Первая серия испытаний заключалась в экспериментальном обосновании возможности использования штатного механизма фильтрации сетевого трафика для блокирования всего трафика. Для проведения таких экспериментов был сконфигурирован брандмауэр Windows, использующий рассматриваемый штатный механизм в режим блокирования, который вел учет всех попыток подключений, представляя результаты такого аудита в виде отладочной информации. В ходе экспериментов этой серии осуществлялась попытка подключения к telnet-сервису, которая блокировалась, со стороны брандмауэра Windows. Результаты протоколировались средствами аудита брандмауэра и анализатором сетевого трафика.

Вторая серия испытаний заключался в создании NDIS-драйвера, передающего и получающего скрытую в полях заголовка IP-пакета информацию. В ходе этих опытов брандмауэр Windows был, как и в первой серии, сконфигурирован в режим блокирования всего сетевого трафика. В результате экспериментов данной серии принимающий NDIS-драйвер в обход брандмауэра Windows получал скрытую информацию в поле заголовка IP- пакета и выводил эту информацию на экран. Результаты подтверждали уязвимость средств фильтрации сетевого трафика (в виде брандмауэра Windows), построенных на базе штатного механизма фильтрации сетевого трафика на уровне ядра ОС Windows.

В заключении изложены основные теоретические и практические результаты работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

В результате диссертационного исследования были решены следующие поставленные задачи:

- разработан метод анализа штатных механизмов обнаружения скрытых информационных воздействий на уровне ядра ОС Windows, основанный на локализации закрытого программного кода, реализующего эти механизмы;
- представлено решение задачи восстановления логической схемы штатных механизмов управления доступом к реестру, контроля создания и удаления процессов, фильтрации сетевого трафика, полученное на основе анализа этих механизмов предложенным автором методом;
- представлено экспериментальное обоснование теоретических исследований по поиску уязвимостей в штатных механизмах защиты ядра ОС Windows.

На основе полученных результатов был сделан следующий вывод.

Проанализированные в работе штатные механизмы обнаружения скрытых информационных воздействий уровня ядра Windows позволяют создавать эффективные средства обнаружения скрытых информационных воздействий, осуществляющие контроль доступа и аудит на уровне исполнительной системы на уровне ядра ОС Windows. Однако данные механизмы содержат в себе уязвимости, позволяющие отключать (обходить) данные механизмы и средства обнаружения скрытых информационных воздействий, построенные на их основе. Следовательно, нельзя гарантировать осуществление аудита и контроля доступа со стороны средств защиты, основанных (использующих) исключительно на данных штатных механизмах.

ОСНОВНЫЕ ПУБЛИКАЦИИ АВТОРА ПО ТЕМЕ ДИССЕРТАЦИИ

1. *Левыкин М.В.* Методы блокирования скрытых каналов по памяти в стеке протоколов TCP/IP. // Труды XXXIV Международной конференции и дискуссионного научного клуба «Информационные технологии в науке, образовании, телекоммуникации и бизнесе (весенняя сессия)», Украина, Крым, Ялта-Гурзуф, 25 мая – 4 июня 2007 г., 2 с.

2. *Левыкин М.В.* Анализ защищенности штатного механизма построения изолированной программной среды в ОС Линукс. Труды XXXV Международной конференции и дискуссионного научного клуба «Информационные технологии в науке, образовании, телекоммуникации и бизнесе (весенняя сессия)», Украина, Крым, Ялта-Гурзуф, 25 мая – 4 июня 2008 г., 2 с.

3. *Левыкин М.В.*: Обход штатного межсетевого экрана ОС Windows XP. // Научный журнал Вестник РГГУ №10/09: Серия «Информатика. Защита информации. Математика» Москва 2009, 11 с.

4. *Левыкин М.В.*: Скрытая загрузка объектов уровня ядра ОС Windows. Труды XXXVI Международной конференции и дискуссионного научного клуба «Информационные технологии в науке, образовании, телекоммуникации и бизнесе (осенняя сессия)», Украина, Крым, Ялта-Гурзуф, 1–10 октября 2009 г., 2 с.

5. *Левыкин М.В.*: Анализ защищенности штатного механизма контроля доступа к реестру на уровне ядра ОС Windows XP. // Научный журнал Вестник РГГУ №12/55: Серия «Информатика. Защита информации. Математика» Москва 2010, 9 с.

Подписано в печать: 10.07.2010 г.
Печать офсетная. Усл. печ. л. 1,5.
Тираж: 100 экз. Заказ №216.
Отпечатано в типографии РИА «Медиа-пресс»
г. Москва, Покровский б-р., д.20/3 стр.4.