И. Ю. Коркин

ТЕХНОЛОГИИ СОКРЫТИЯ ВРЕДОНОСНЫХ ПРОГРАММ И НОВЫЕ СПОСОБЫ ПРОТИВОДЕЙСТВИЯ ИМ

Вредоносное программное обеспечение (ВПО) создается для обеспечения постоянного неопределяемого присутствия на компьютере пользователя и активного сетевого взаимодействия, например, для передачи добытой информации на выделенный компьютер. При этом для сокрытия ВПО используются технологии, в которых применяются стеганографические и руткит-механизмы. С учетом этого, а также новейших технологий сокрытия ВПО предлагается следующая классификация механизмов сокрытия ВПО (рис. 1).

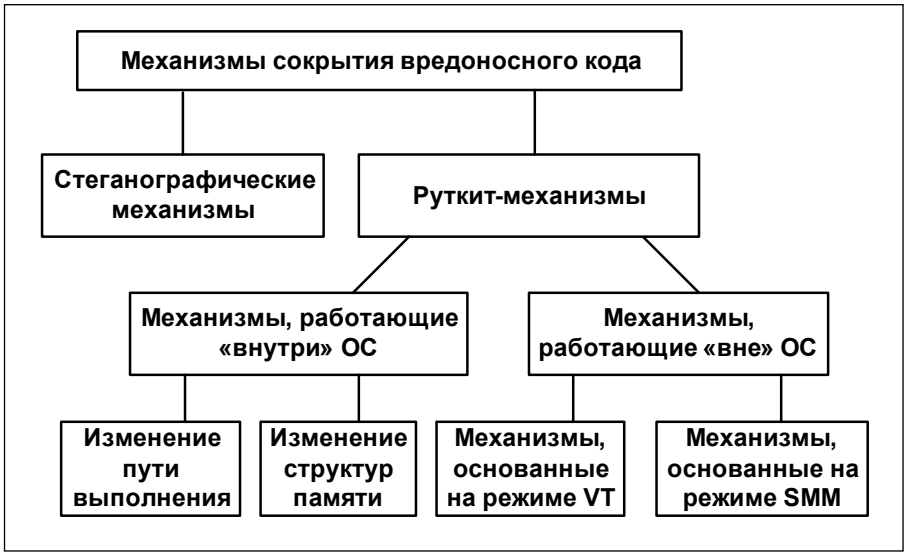

Рис. 1. Схема классификации механизмов сокрытия ВПО

Стеганографические механизмы скрывают истинное предназначение внедренных объектов маскировкой их под легитимные, например схожестью их имен с именами системных файлов. В результате вредоносные файлы видны, но могут не вызывать подозрения.

ВПО, использующее только стеганографические механизмы сокрытия, можно отнести к нулевому типу по классификации Рутковской [1].

Заметим, что стеганографические механизмы по сравнению с другими не используют технических средств сокрытия, не «вмешиваются» в работу системы, что обеспечивает поведенческую необнаруживаемость для антивирусных средств.

Кроме того, для работы таких вредоносных программ не требуется прав администратора, что существенно расширяет аудиторию потенциально заражаемых компьютеров. Такие средства обладают переносимостью на различные версии ОС Windows и оборудования.

Однако из-за отсутствия технических средств сокрытия такое ВПО может быть легко обнаружено и удалено. Большую опасность представляют комбинации стеганографических механизмов с другими средствами сокрытия.

Ко второй группе относятся вредоносные программы, использующие для сокрытия руткит-механизмы по принципу: не виден объект, значит, его и нет.

Объектом может выступать процесс, поток, драйвер, файл, папка на диске, сетевой порт, ключ в реестре и т. д. [2]. Руткит-механизмы могут изменять как пути выполнения, так и структуры памяти (DKOM). При этом они могут работать как «внутри» ОС, так и «вне» ее.

Руткит-механизмы разделяются следующим образом:

- руткит-механизмы, работающие «внутри» ОС, которые основываются на изменении пути выполнения либо на изменении структур памяти;
- руткит-механизмы, работающие «вне» ОС.

Для изменения пути выполнения ВПО осуществляет перехват функции штатного обработчика и передает управление вредоносному обработчику, который вносит целенаправленные изменения в возвращаемый результат (заметим, кстати, что подобным образом функционируют многие антивирусные средства при сигнатурном поиске вредоносного кода). Такие ВПО можно отнести к первому типу по классификации [1].

Способы обнаружения описанного механизма сокрытия приведены в работе [3].

Работающие «внутри» ОС руткит-механизмы не добавляет новых обработчиков в систему, а особым образом изменяют структуры памяти. Данные структуры могут быть расположены как в ядерной памяти [4], так и в памяти пользовательского режима [5]. Подобные ВПО можно отнести ко второму типу по классификации [1].

Для уяснения разработанного автором способа противодействия таким руткитам рассмотрим их функционирование на примере работы сервисов (заметим, что этим способом можно выявлять и некоторые объекты, сокрытые способом изменения пути выполнения).

В ОС Windows после инсталляции сервиса в систему в контексте процесса Servises в пользовательской части памяти создается структура, хранящая информацию об этом сервисе. После его деинсталляции соответствующая структура частично очищается. Таким образом, каждому инсталлированному сервису однозначно соответствует определенная структура.

Система сама выбирает адрес в памяти, по которому загружается описываемая структура для нового инсталлированного сервиса. Новая структура может размещаться в памяти как выше, так и ниже ранее загруженной. Структуры работающих сервисов связаны двунаправленным списком, в образовании которого участвуют поля Flink и Blink структуры ServiseList, хранящие адреса структуры ServiseList, следующей и предыдущей структуры соответственно, как показано на рис. 1.

Проведенные нами исследования показали, что штатные средства ОС Windows, например функция EnumServicesStatus, для получения списка зарегистрированных сервисов используют проход по двусвязному списку структур, хранящемуся в контексте процесса Services. Схематично вышеописанный список представлен на рис. 2. На нем прямоугольниками обозначены структуры сервисов, стрелками — двунаправленная связь между ними.

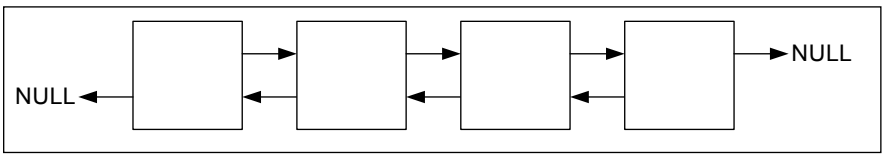

Рис. 2. Связи в списке структур сервисов без внедренного ВПО, сокрытого руткит-механизмом

С целью сокрытия связи между структурами можно изменять (на рис. 3 это показано изменением стрелок для второго прямоугольника). Поскольку функционирование не нарушается, штатные средства не обнаруживают этот сервис.

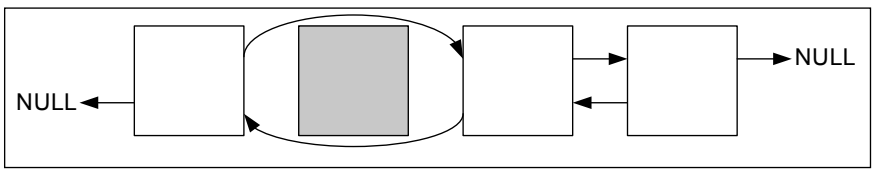

Рис. 3. Сокрытие сервисов путем изменения связей между их структурами

Автором разработан способ для выявления ВПО, сокрытого с использованием таких руткит-механизмов [6, 7].

Способ основан на динамических битовых сигнатурах структур объектов и имеет следующие достоинства:

- способ достаточно универсален и обладает переносимостью на различные версии Windows;
- ему трудно противодействовать, поскольку для получения готового списка объектов функции ОС не используются;
- способ может быть интегрирован с другими антивирусными средствами.

Однако предложенный способ не позволяет выявить ВПО, работающее «вне» ОС на компьютерах, оснащенных новейшими процессорами Intel и AMD, поддерживающими новую технологию — аппаратную виртуализацию (АВ).

Отличительными особенностями АВ являются:

- возможность запустить код монитора виртуальных машин (МВМ), работающий между ОС и аппаратным обеспечением, что исключает возможность контроля этого кода из ОС;
- поскольку АВ поддерживается процессором, то обеспечивается необходимый функционал и быстродействие;
- не существует штатных средств, позволяющих определить, запущена ли система под виртуализацией или нет [8].

Перечисленное выше позволяет создавать ВПО, работающее «вне» ОС, обнаружить которое пока не представляется возможным.

При разработке средств противодействия необходимо, во-первых, учитывать, что функционирующие в среде АВ руткиты аналогичны по скрытности более раннему режиму системного управления SMM (ВПО, использующее эти механизмы сокрытия, можно отнести к третьему типу классификации [1]), и, во-вторых, учитывать различия этих режимов, а именно:

- режим SMM обеспечивает независимое от ОС окружение, которое может функционировать как на более старых процессорах, без поддержки АВ, так и на новейших процессорах, поддерживающих виртуализацию;
- ВПО на базе технологии АВ работает только на новых процессорах (младше 1–2 лет), в то время как ВПО, построенное на базе режима SMM, функционирует на более старых процессорах;
- ВПО на базе технологии АВ работает в страничной области памяти, в то время как ВПО, построенное на базе режима SMM, оперирует физической памятью без страниц;
- ВПО на базе технологии АВ более гибкое, поскольку может перехватывать события более высокого уровня, такие как выполнение привилегированных инструкций, запись и чтение регистров, доступ к памяти и прерывания. В то же время у ВПО, построенного на базе режима SMM и контролирующего периферийное оборудование, возможности перехвата ограничены низкоуровневыми событиями [9].

Из изложенного следует, что ВПО, построенное на технологии АВ, является наиболее опасным — его трудно выявлять, поскольку необходимо определять не только факт того, что система запущена под виртуализацией, но и является ли МВМ вредоносным. Поэтому разработка средств противодействия таким ВПО является приоритетной задачей.

О состоянии проблемы обнаружения вредоносных программ, сокрытых описанными выше способами, можно судить по приведенной ниже таблице 1.

Таблица 1. Технологии сокрытия ВПО и возможности его обнаружения

<table>
<tr><td colspan="3">Механизм сокрытия</td><td>Технические средства или способы обнаружения ВПО</td></tr>
<tr><td rowspan="4">Руткит-механизмы</td><td rowspan="2">Механизмы, работающие «внутри» ОС</td><td></td><td>Имеются, они рассмотрены в [3]</td></tr>
<tr><td></td><td>Имеется авторская разработка [6]</td></tr>
<tr><td rowspan="2">Механизмы, работающие «вне» ОС</td><td></td><td>Отсутствуют</td></tr>
<tr><td></td><td>Отсутствуют</td></tr>
<tr><td colspan="3">Стеганографические</td><td>Отсутствуют</td></tr>
</table>

Детальное описание классификации механизмов сокрытия, работающих «внутри» ОС, представлено в работе [3].

СПИСОК ЛИТЕРАТУРЫ:

1. *Rutkowska J.* Introducing Stealth Malware Taxonomy. URL: http://invisiblethings.org/papers/malware-taxonomy.pdf.
2. *Хоглунд Дж., Батлер Г.* Руткиты: внедрение в ядро Windows. СПб.: Питер, 2007.
3. *Коркин И. Ю.* Исследование способов сокрытия вредоносного кода в ОС Windows. URL: http://sites.google.com/site/igorkorkin/home/ИсследованиеспособовсокрытиявредоносногокодавОСWindows.doc?attredirects=0.
4. *Коркин И. Ю.* Обнаружение скрытых процессов, потоков и драйверов в 32-разрядных ОС линейки Windows. URL: http://sites.google.com/site/igorkorkin/home/Обнаружениескрытыхпроцессов%2Cпотоковидрайверовв32-разрядныхОСлинейкиWindows.doc?attredirects=0.
5. *Коркин И. Ю.*Обнаружение скрытых сервисов в 32-разрядных ОС линейки Windows. URL: http://sites.google.com/site/igorkorkin/home/Обнаружениескрытыхсервисовв32-разрядныхОСлинейкиWindows.doc?attredirects=0.
6. *Коркин И. Ю.* Система обнаружения скрытого программного обеспечения в ОС Windows. URL: http://sites.google.com/site/igorkorkin/home/СистемаобнаруженияскрытогопрограммногообеспечениявОСWindows.doc?attredirects=0.
7. *Коркин И. Ю.* Способ обнаружения скрытых процессов в ОС Windows // Бизнес и безопасность в России. 2009. № 53.
8. Intel® 64 and IA-32 Architectures Software Developer's ManualVolume 3B: System Programming Guide.
9. *Embleton S., Sparks S.* SMM Rootkits. URL: http://www.hakim.ws/BHUSA08/speakers/Embleton_Sparks_SMM_Rookits/BH_US_08_Embleton_Sparks_SMM_Rootkits_WhitePaper.pdf.