УДК 004.056.57

Фомин И. П., Маткин И. А.

ТЕХНОЛОГИЯ АППАРАТНОЙ ВИРТУАЛИЗАЦИИ В КОНТЕКСТЕ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

Технология аппаратной виртуализации предоставляет широкие возможности для программистов. Многие из этих возможностей интересны в контексте компьютерной безопасности. Целью данной работы является исследование технологии аппаратной виртуализации и выявление ее возможностей, применимых для решения задач в сфере компьютерной безопасности. В частности интересна возможность применения аппаратной виртуализации в руткитах. В результате проделанной работы были исследованы следующие возможности аппаратной виртуализации: контроль обращения к портам ввода вывода, возможность контроля таблицы страниц, а так же контроль обработки прерываний. Был разработан прототип руткита, использующего эти возможности для скрытного функционирования в системе и сбора информации.

Ключевые слова: *аппаратная виртуализация, руткит, гипервизор.*

Fomin I., Matkin I.

HARDWARE VIRTUALIZATION TECHNOLOGIES IN THE CONTEXT OF COMPUTER SECURITY

Tehnologiya hardware virtualization provides great opportunities for programmers. Many of the features of interest in the context of computer security. The aim of this work is to investigate hardware virtualization technology and the identification of its possible application to solve problems in the field of computer security. Particularly interesting possibility of using hardware virtualization rootkits. As a result of this work have been investigated following hardware virtualization: control access to ports of the output, the ability to control the page table, as well as control interrupt handling. Developed a prototype rootkit using these opportunities for covert operation of the system and gather information.

Keywords: *hardware virtualization, rootkit, hypervisor.*

На данный момент наиболее широкое распространение на рабочих станциях имеют две технологии аппаратной виртуализации: IntelVMXв процессорах семейства x86 от Intel, а так же AMD-V в процессорах AMDAthlon, которые появились в 2006 году.

Перед производителями процессоров были поставлены следующие требования:

1. Изоляция — каждая виртуальная машина должна иметь доступ только к тем ресурсам, которые были ей назначены. Она не должна иметь возможности повлиять на работы как монитора, так и других ВМ.

2. Эквивалентность — любая программа, исполняемая под управлением ВМ, должна демонстрировать поведение, полностью иден-

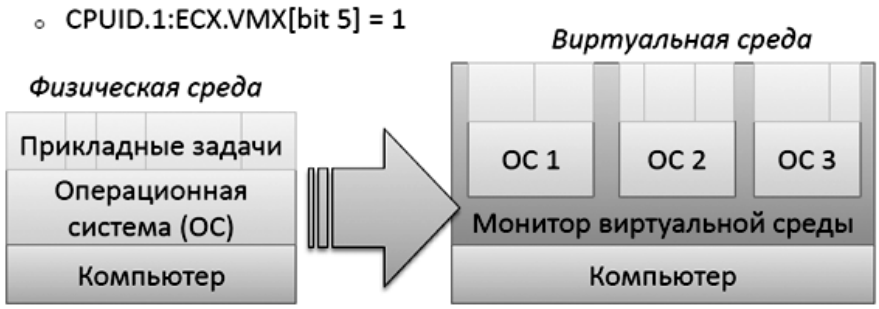

Рис. 1. Компьютер без гипервизора и с гипервизором

тичное её исполнению на реальной системе, за исключением эффектов, вызванных двумя обстоятельствами: различием в количестве доступных ресурсов (например, ВМ может иметь меньший объём памяти) и длительностями операций (из-за возможности разделения времени исполнения с другими ВМ).

3. Эффективность — в оригинальной работе условие сформулировано следующим образом: «статистически преобладающее подмножество инструкций виртуального процессора должно исполняться напрямую хозяйским процессором, без вмешательства монитора ВМ». Другими словами, значительная часть инструкций должна симулироваться в режиме прямого исполнения.

Идея работы аппаратной виртуализации заключена в работе монитора виртуальных машин (VMM) или гипервизора. Гиппервизор — это некий код имеющий уровень привилегий более высокий, чем уровень привилегий операционной системы, а так же неограниченный доступ к аппаратным ресурсам компьютера. При включении гипервизора на машине, все действия операционной системы начинают контролироваться гипервизором. Схематично это изображено на рис. 1.

Из этого следует, что аппаратная виртуализация предоставляет программистам широкий спектр воздействия на систему в самом привилегированном режиме. Это предоставляет очень мощную возможность для сокрытия вредоносного программного обеспечения.Так, с одной стороны, гипервизор, выполняющий функции монитора виртуальных машин, повышает сервисные возможности компьютера и снижает его эксплуатационные расходы. Благодаря монитору виртуальных машин на одном компьютере может быть одновременно запущено несколько ОС в разных виртуальных машинах. Но, с другой стороны, гипервизор можно негласно внедрить как программную закладку несущую угрозу информационной безопасности.

Рис. 2. Схема внедрения нелегитимного гипервизора

Далее рассмотрим интересные в контексте компьютерной безопасности возможности аппаратной виртуализации. Аппаратная виртуализация предоставляет средство контроля обращений к портам, то есть контролируются команды типа IN, INS/INSB/INSW/INSD, OUT, OUTS/OUTSB/OUTSW/OUTSD. Суть этого заключается в перехвате управления гипервизором в случае выполнения операций с портами из непривилегированного режима (в данном контексте привилегированным является режим работы гипервизора, а непривилегированный — пользовательский и режим ядра операционной системы). Из этого следует возможность контроля за операциями ввода и вывода, а также способ контроля взаимодействия программного кода с аппаратным обеспечением. Сказанное выше к примеру предоставляет возможность перехвата управления гипервизором в случае нажатия клавиши. В последствии в зависимости от логики работы гипервизора это событие можно обработать разными способами: можно сохранять нажатую клавишу, подменять код нажатой клавиши, либо скрывать нажатие клавиш. По такому принципу был разработан прототип руткита, представляющий драйвер режима ядра, который после загрузки в систему устанавливает гипервизор, регистрирует обработчики и использует специализированную команду для запуска гипервизора. Суть работы заключается в следующем: при регистрации гипервизора есть способ установить контроль обращений к портам ввода вывода путем установки определенного бита в единицу. Далее указывается гипервизору номер порта для контроля, в данном случае это порт с номером 0x60. В случае, если после установки контроля за портом, отвечающим за клавиатуру, пользователь станет вводить с клавиатуры какие-либо символы, управление перейдет к гипервизору, который сохраняет код нажатой клавиши и возвращает управление пользователю. Все это происходит незаметно для пользователя и антивирусных средств, так как гипервизор напрямую работает с процессором. Схематично этот процесс можно изобразить блок схемой (Рис. 3).

Контроль прерываний. Суть контроля прерываний заключается в перехвате гипервизором управления в случае прерываний, а также установка своих обработчиков для обработки прерываний. Данное средство позволяет перехватывать все виды прерываний, как программные, так и аппаратные. Например, мы имеем способ контроля исключений при обращении к выгруженной странице памяти. Эта возможность предоставляет нам интересный способ контроля операций обращения к памяти. Механизм этого состоит в отключении некоторых областей памяти путем выставления определенных битов, отвечающих за их работу, в нуль. Это вызывает, в случае обращения к соответствующим страницам, возникновение аппаратного прерывания, так как происходит обращение к недоступной памяти. После этого управление получает гипервизор. Подобная функциональность была реализована в прототипе руткита, для обеспечения сокрытия

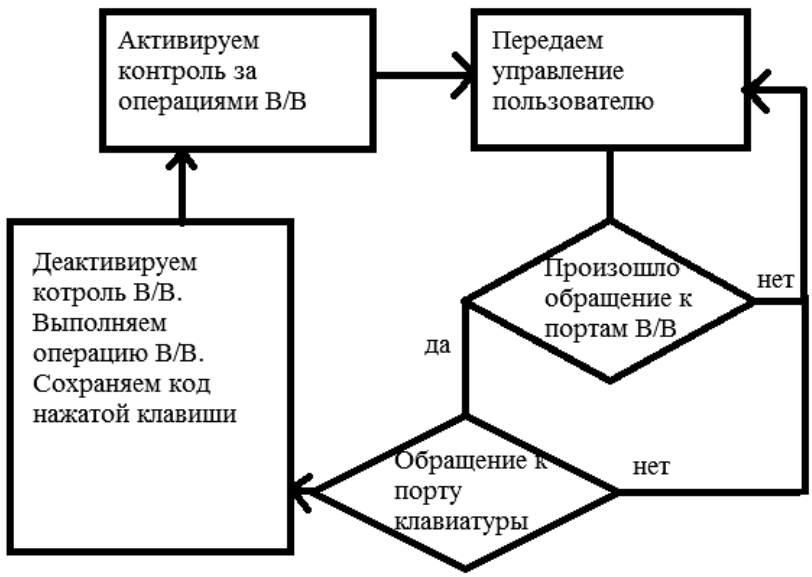

Рис. 3. Блок-схема алгоритма перехвата

руткита в системе. То есть, сокрытие происходит в том случае, если обращение на чтение к области памяти руткита происходит не из кода руткита, а за его пределами. Реализовано это было следующим образом. В гипервизоре устанавливается поддержка контроля прерываний. Далее в таблице страниц, обнуляется бит присутствия соответствующих страниц, чтобы при обращении к областям памяти, где расположен руткит, происходило прерывание, и управление получал гипервизор. В случае возникновения исключения управление передается обработчику гипервизора, в котором происходит проверка, какая операция вызвала исключение. Если это была операция чтения или записи, проверяется адрес инструкции, вызвавшей исключение. Если инструкция принадлежит коду руткита, то выставляется флаг трассировки в гипервизоре, так же бит присутствия страниц руткита вновь выставляется в единицу и управление передается коду, вызвавшему исключение. Далее благодаря установленному флагу трассировки, после выполненной инструкции управление вновь получает гипервизор, который снова устанавливает соответствующие биты присутствия страниц в нули, чтобы не пропустить последующие обращения к областям памяти руткита, снимается флаг трассировки и управление передается пользовательскому процессу. Если инструкция чтения / записи принадлежит не коду руткита, то необходимо скрыть код руткита от нежелательных действий. В данном случае происходит сохранение кода руткита в буфер и заполнение соответствующих руткиту областей памяти нулями. Далее происходит установка бита присутствия страниц в единицу, установка флага трассировки и передача управления вызвавшему исключение коду. После выполнения вызвавшей исключение инструкции код, вызвавший данную инструкцию, получит нули, и управление вновь перейдет к гипервизору, который вновь выставит биты присутствия в нули и снимет флаг трассировки. В случае если была инструкция передачи управления, то биты присутствии выставляются в единицу, выставляется флаг трассировки и управление получает руткит. Таким образом, блокируется внешнее воздействие на код руткита, не мешая деятельности руткита.

В заключении скажем, что поставленные задачи в данной работе были выполнены. В ходе работы были рассмотрены и изучены две интересные возможности аппаратной виртуализации в контексте компьютерной безопасности. Рассмотренные задачи и реализованные прототипы руткита впоследствии могут иметь применение для написания средств обнаружения руткитов. В будущем планируется рассмотрение других интересующих возможностей аппаратной виртуализации. А также планируется рассмотрение методов противодействия этому и написание антируткита.

Примечания

1. Intel® 64 and IA-32 Architectures Software Developer's Manual: в 3 т. //ОфициальныйсайтIntel. URL: http://http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-1-manual.pdf

2. Gerhart. Примеры использования функциональности аппаратной виртуализации процессоров Intel для реализации Anti-VmDetect для Vmware[электронный ресурс]. – Электронные текстовые данные. - Режим доступа http://www.securitylab.ru/analytics/427855.php, свободный.

3. Методика обнаружения нелегитимного программного обеспечения, использующего технологию аппаратной виртуализации [Текст] :автореф. дис. ... канд. техн. наук : 05.13.19 / И. Ю. Коркин. - М., 2011. - 21 с. : ил. - Библиогр.: с. 20-21

4. Александр Самойленко. Технологии аппаратной виртуализации[электронный ресурс]. – Электронные текстовые данные. – Режим доступа http://www.ixbt.com/cm/virtualization-h.shtml, свободный.

Фомин Игорь Павлович, студент, Челябинский государственный университет. E-mail: kraken_pro@mail.ru

Маткин Илья Александрович, ст. преподаватель, Челябинский государственный университет

Fomin Igor Pavlovich, student, Chelyabinsk State University. E-mail: kraken_pro@mail.ru

Matkin Ilya Alexandrovich, senior lecturer, Chelyabinsk State University