Руткиты: проблемы безопасности и тенденции развития

Игорь Коркин (igor.korkin@gmail.com, @Igorkorkin)

Рассмотрим современные тенденции развития руткитов и методов их обнаружения

В настоящее время чётко просматривается смещение вектора компьютерных атак от массового заражения к целевым, точечным атакам. Как заметил Е. Касперский, если девяностые годы были десятилетием киберхулиганов, двухтысячные были десятилетием киберпреступников, то сейчас наступила эра кибервойн и кибертеррора. Иллюстрацией этому являются всем известные примеры вредоносного программного обеспечения (ВПО) – Stuxnet, Duqu, Flamer, Gauss, которые многие антивирусные компании причисляют к кибероружию. Для обеспечения устойчивого и неопределяемого присутствия в компьютерной системе ВПО использует специальные механизмы, называемые руткит-механизмами. В результате ВПО работает незаметно как для пользователя, так и для средств защиты.

XXX ЗАГОЛОВОК XXX
Основные тенденции в компьютерной безопасности

Одним из ярких примеров использования кибероружия является шпионская сеть «Красный октябрь», которая в течение пяти лет занималась активным добыванием информации из правительственных организаций, различных исследовательских институтов, крупных международных компаний. Серьезная защищённость этих объектов не остановила работу вредоносной системы: она была раскрыта всего несколько месяцев назад, что свидетельствует о возрастающей угрозе вмешательства в работу любой компьютерной системы.

Появление новой ОС Windows 8 не изменило ситуацию в лучшую сторону. Новая ОС сохранила от своих предшественников часть прежних механизмов защиты (UAC, ASLR, DEP, SEHOP, SafeSeh, PatchGuard, цифровые подписи для драйверов), для которых существует возможность обхода, что неоднократно упоминалось в открытых источниках.

И хотя в Windows 8 появились новые механизмы – Secure Boot, SMEP и ELAM, однако, защищённости они прибавили мало. Об этом свидетельствуют демо-образец буткита “Stoned Lite” П. Кляйсснера (P. Kleissner) (http://bit.ly/stolite) и UEFI bootkit А. Аллиеви (A. Allievi)

(http://bit.ly/efiall), а также результаты анализа технологии SMEP А. Шишкина из компании «Positive Technologies» (http://bit.ly/smepw8).
Кроме того, на конференции «Hackito Ergo Sum» в 2012 было представлено любопытное средство "Rakshasa" от Дж. Броссара (J. Brossard) (http://bit.ly/brossard), построенное на open-source компонентах.
Из приведенного следует, что в последней версии Windows для борьбы с ВПО ничего революционного не предусмотрено, а достойных механизмов, способных сильно осложнить жизнь разработчикам руткитов, на сегодняшний момент нет.

XXX ЗАГОЛОВОК XXX
Механизмы сокрытия в системе

Для сокрытия ВПО могут применяться различные механизмы. Их классификация предложена в работе Дж. Рутковской (J. Rutkowska) «Introducing Stealth Malware Taxonomy» (http://bit.ly/taxon). В последующем данная классификация была нами расширена (рис. 1).
Стеганографические механизмы скрывают истинное предназначение внедрённых объектов маскировкой их под легитимные, например, схожестью их имён с именами системных файлов. В результате вредоносные файлы видны пользователю, но не вызывают подозрения. Примером стеганографического сокрытия является использование сертификатов доверенных компаний для подписи вредоносных драйверов. Благодаря действительным сертификатам компаний Realtek и JMicron червь Stuxnet долгое время оставался незамеченным, а компоненты червя Flame имели цифровую подпись самой компании Microsoft.
Для работы стеганографических механизмов не требуется повышенных привилегий, и, кроме того, они обладают переносимостью на различные версии ОС Windows и оборудование. Однако из-за отсутствия технических механизмов сокрытия такое ВПО может быть легко обнаружено и удалено. Особую опасность представляют комбинации стеганографических с другими механизмами сокрытия.
Ко второй группе относятся технические механизмы сокрытия, в результате работы которых информация о скрываемом объекте становится недоступной средству обнаружения («не виден объект, значит, его и нет»). Эти механизмы можно разделить на руткит-механизмы, работающие «внутри» и «вне» ОС.

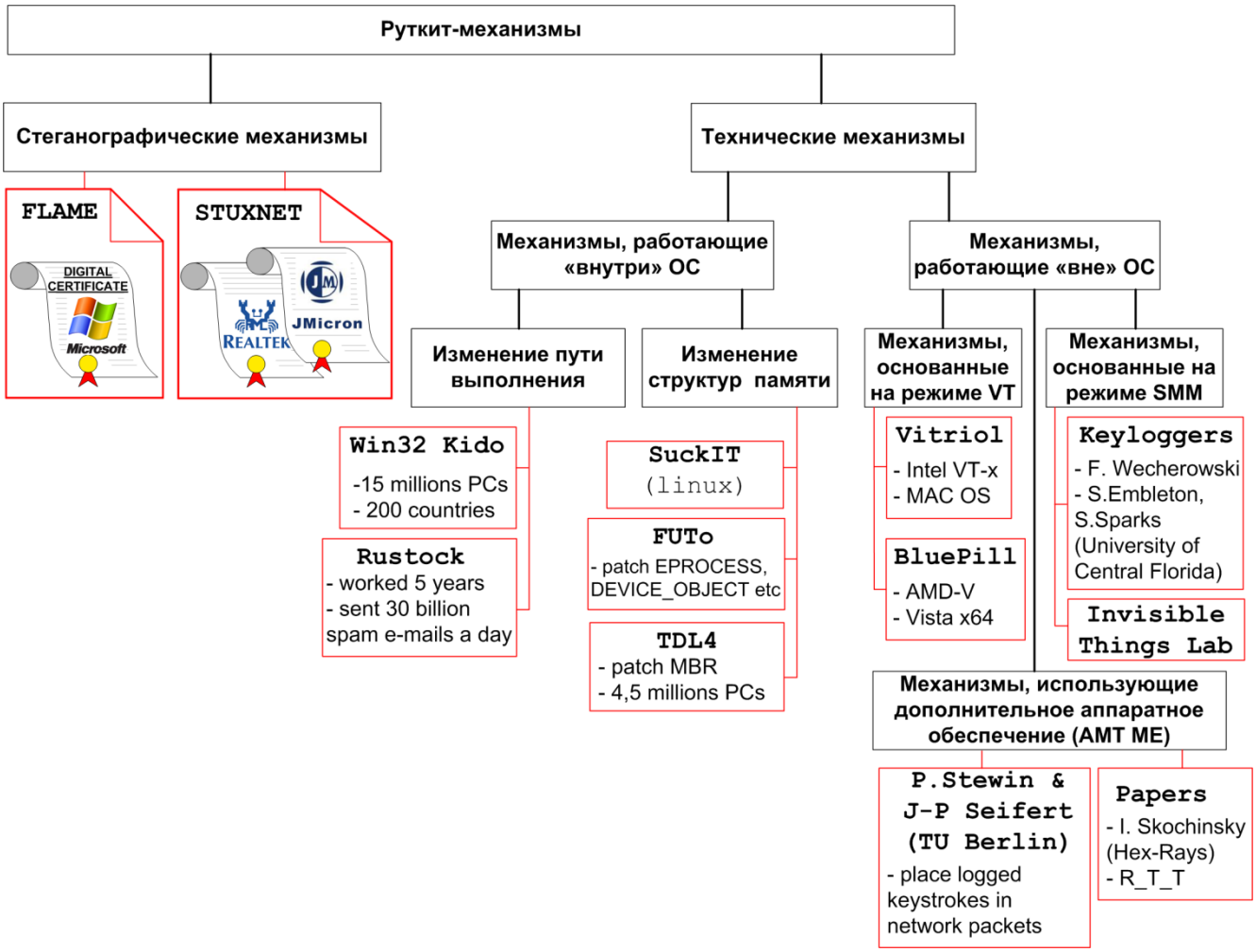


Рисунок 1 – Схема классификации механизмов сокрытия программного обеспечения

В руткит-механизме «внутри ОС» объектом может выступать процесс, драйвер, файл на диске, сетевой порт, ключ в реестре. Для своей работы руткит-механизмы могут изменять как пути выполнения, так и структуры памяти, как в пользовательском, так и в системном адресных пространствах.

Для изменения пути выполнения ВПО осуществляет перехват функции штатного обработчика и передаёт управление вредоносному обработчику, который вносит целенаправленные изменения в возвращаемый результат. Способы обнаружения описанного механизма сокрытия уже освещались в открытых источниках.

Вторая подгруппа руткит-механизмов, работающих «внутри» ОС, не добавляет новых обработчиков в систему, а особым образом изменяет структуры памяти, хранящие информацию о скрываемом объекте. Примером таких структур, расположенных в системном адресном пространстве и представляющих интерес для руткитов, являются _KRPCB, _ETHREAD, _EPROCESS, _MODULE_ENTRY, _DRIVER_OBJECT, плюс БД зарегистрированных драйверов и служб, расположенная в пользовательском пространстве процесса SERVICES.EXE.

Руткит-механизмы «вне ОС» основаны на установке собственного или модификации существующего обработчика событий в том или ином режиме работы

процессора либо дополнительного аппаратного обеспечения. Для их работы необходимо наличие набора микросхем с поддержкой требуемой технологии. Можно выделить руткит-механизмы, построенные на основе режима аппаратной виртуализации, режима системного управления, кода, использующего технологии Active Management Technology (AMT) и Management Engine (ME) и др. Широко известный в узких кругах автор R_T_T в своих работах «Кремневый беспредел» описывает возможности и угрозы информационной безопасности не только от указанных технологий, но и от механизма обновления микрокода процессора (http://bit.ly/VRQD6O и http://bit.ly/104EsRB).

Угрозы технологии ME также освещались в докладе И. Скочинского (I. Skochinsky) "Rootkit in your laptop: Hidden code in your chipset and how to discover what exactly it does" (http://bit.ly/igorsko) на конференции "Breakpoint" в 2012 году.

В Берлинском техническом университете была выполнена работа по разработке руткита на основе технологий AMT и ME – "In God We Trust All Others We Monitor" (http://bit.ly/amtberlin). Её авторы П. Стивен (P. Stewin) и Ж.-П. Сейферт (J.-P. Seifert) создали клавиатурного шпиона, отправляющий собранные данные по сети скрытно от средств защиты.

Сравнительный анализ руткитов, построенных на базе технологии аппаратной виртуализации и режима системного управления, приведён в работах Ш. Эмблитона (Sh. Embleton) и Ш. Спаркс (Sh. Sparks) "SMM Rootkits: A New Breed of OS Independent Malware" (http://bit.ly/smmbh08) и "SMM Rootkits" (http://bit.ly/smmbh08).

XXXX

ВРЕЗКА

XXXX

Интересная техника сокрытия руткитов

XXXX

На конференции ZeroNight в 2012 году была представлена работа Д. Олексюка (aka Cr4sh) с описанием способа размещения руткита не в файлах, а в реестре с помощью Differentiated System Description Table (DSDT). Согласно докладу автора, преимущество способа в том, что ни одно средство для обнаружения руткитов не учитывает такую возможность.

XXXX

XXX ЗАГОЛОВОК XXX

Антируткиты

Большинство образцов нового ВПО, о которых шла речь ранее, использовали для своей работы драйвера. Поэтому рассмотрим наиболее распространенные антируткит средства, способные обнаруживать наличие скрытых драйверов.
Сокрытие драйвера от штатных средств ОС хорошо описано в популярных источниках, таких как «Rootkits: Subverting the Windows Kernel» и новой книге «The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System».
Из популярных внештатных средств с поддержкой Windows 8 можно выделить следующие: Gmer, XueTr, PowerTool, TDSSKiller (kaspersky labs).
С позиции обнаружения скрытых драйверов первые три средства имеют схожие алгоритмы работы, использующие для обнаружения побайтовый поиск в памяти фрагментов структур драйверов. Средство TDSSKiller использует несколько иной список, информация в который заносится при загрузке драйверов через штатные средства Windows.
Изменение полей в необходимых структурах и удаление из соответствующих списков обеспечит сокрытие драйвера от этих средств, без нарушения работы системы и самого ВПО. Это позволяет констатировать отсутствие в открытом доступе антируткитных средств, стойких к противодействию.

XXX ЗАГОЛОВОК XXX
Программно - аппаратные руткиты

Программно-аппаратные руткиты функционируют «вне ОС». Отдельно уделим внимание руткитам, построенным на основе технологии аппаратной виртуализации. Во-первых, их можно установить с помощью драйверов – штатного механизма различных ОС. Во-вторых, такие руткиты могут перехватывать события более высокого уровня, чем другие. В-третьих, они лучше документированы.
С 2006 г. компании Intel и AMD выпускают процессоры с поддержкой технологии аппаратной виртуализации. ПО, использующее технологию аппаратной виртуализации (гипервизор, ПОАВ) работает в новом режиме, более привилегированном, чем ОС. Технология аппаратной виртуализации позволяет запускать во вложенном виде несколько различных гипервизоров.

XXXX
ВРЕЗКА

XXXX

Исходные коды гипервизоров-драйверов для ОС Windows x86

XXXX

В приложении можно найти следующие исходники

1. BluePill (версии 0.11 и 0.32) – демонстрационный образец гипервизора для систем AMD, после публикации которого и началось широкое обсуждение угроз ИБ от аппаратной виртуализации.
2. VMXCPU – исходный код заглушки гипервизора Ш. Эмблитона (Sh. Embleton) для процессоров Intel, который уже готов к использованию.
3. Invisible Lane (il) – исходный код авторского скрытого гипервизора, построенного на основе vmxcpu. Сокрытие осуществляется путём компрометации процессорного счётчика тактов TSC, величина компрометации может задаваться с точностью до 1 такта.

XXXX

XXXX

ВРЕЗКА

XXXX

Средства отладки гипервизоров

XXXX

Специфика работы гипервизора не всегда позволяет использовать популярные средства отладки, такие как vBox (VmWare) вместе с Windbg, вместо этого можно использовать эмуляторы Bochs или AMD SimNow, однако они сложны в настройке для новичка.

Что же можно использовать:

1. Вывод отладочных сообщений через DbgPrint и просмотр их с помощью DbgView от М. Руссиновича (M. Russinovich) скорее используется для демонстрации корректной работы гипервизора, чем для его отладки. Сбой в работе гипервизора приводит к нарушению работы системы и всех отладочных средств запущенных в ОС. В этом случае DbgView ничего не покажет.
2. Отправлять отладочные сообщения на COM-порт. Этот способ использовали авторы BluePill, сохранив в исходнике реализации этих функций.
3. Использовать отладочную плату, например, «PTI8 Diagnostic Post Test Card Debug Card PCI Analyzer». При включении компьютера, на LCD-дисплее этой платы можно будет увидеть POST сообщения BIOS. Для

создания таких сообщений со значением «value» можно использовать сишную конструкцию outb(value, 0x80).

XXXX

Гипервизор сочетает в себе возможности как средства защиты, так и угрозы внедрения программных закладок.

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 (рис. 2). Но, с другой стороны, гипервизор можно негласно внедрить как программную закладку с бесконтрольными возможностями, несущими угрозу информационной безопасности.

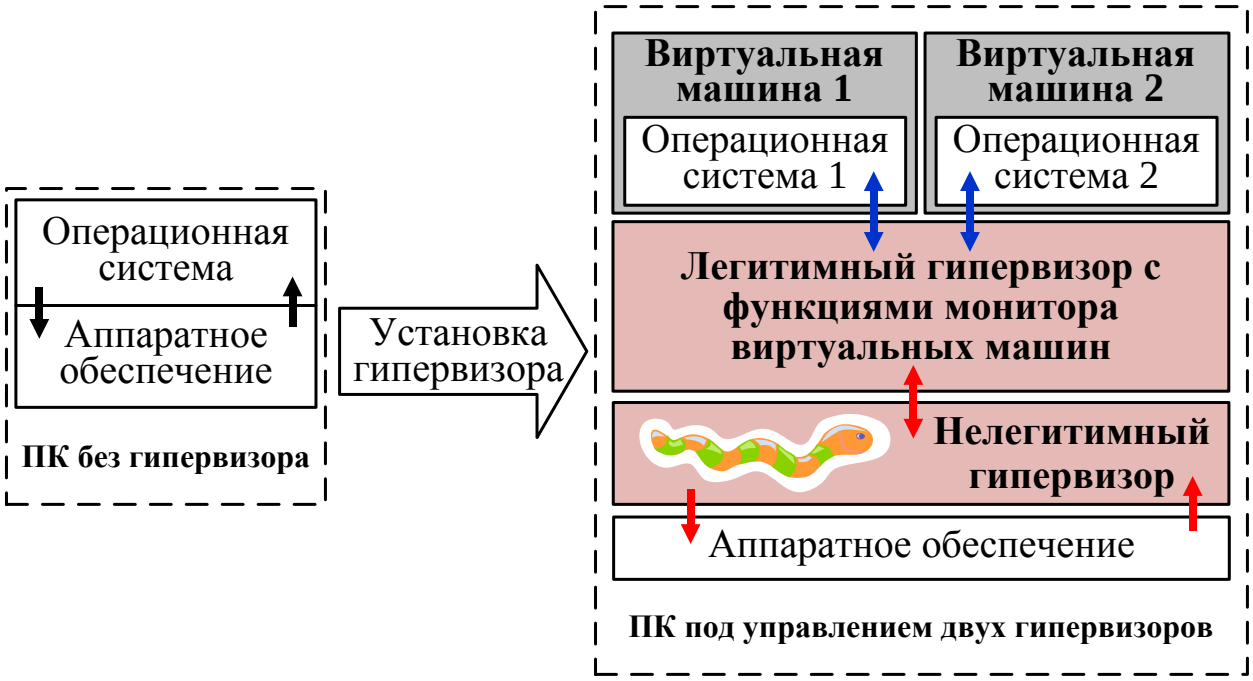


Рисунок 2 – Схема взаимодействия операционной системы с аппаратным обеспечением в случаях отсутствия (присутствия) двух гипервизоров: легитимного с функциями монитора виртуальных машин и нелегитимного, находящихся во вложенном виде

В открытом доступе имеются два программных средства, «BluePill» и «Vitriol», реализованные в виде драйверов, которые устанавливают гипервизор прозрачно для пользователя.

Программное средство «BluePill» было разработано в 2006-2007 гг. исследователями Д. Рутковской (J. Rutkowska), А. Терешкиным (A. Tereshkin), Р. Войтчуком (R. Wojtczuk) и Р. Фаном (R. Fan) из компании Invisible Things Labs для процессоров AMD.

Другим примером гипервизора является программное средство «Vitriol», разработанное для процессоров Intel исследователем Дино А. Даи Зови (Dino

A. Dai Zovi) из компании Matasano Security одновременно с BluePill в 2006 году.

Обнаружением гипервизоров занимались как целые компании: Komoku, North Security Labs и др., так и отдельные специалисты. Даже сама компания Microsoft опубликовала интерфейс для обнаружения гипервизоров (http://bit.ly/hyperms), согласно которому необходимо выполнить инструкцию CPUID, предварительно записав в регистр EAX единицу. Далее необходимо проверить значение 31 бита регистра ECX; если он выставлен, то в системе присутствует гипервизор, а информация о его возможностях передаётся в структуре HV_CPUID_RESULT. Однако этот интерфейс не защищён от компрометации (о её преодолении см. далее).

Несмотря на широкую распространённость гипервизоров, штатные средства для их обнаружения отсутствуют, а опубликованные имеют существенные недостатки, такие как отсутствие возможности выявить гипервизор в случае его противодействия обнаружению, а также неудобство использования и тиражирования ряда средств. Под удобством тиражирования понимается отсутствие в средстве обнаружения внешнего аппаратного компонента, необходимого на протяжении всего времени работы.

В связи с широким распространением различного ПО, использующего технологию аппаратной виртуализации, особую опасность представляет нелегальный гипервизор, который для своего сокрытия использует санкционировано установленный пользователем (легитимный) гипервизор с помощью вложенной виртуализации. В открытых источниках отсутствуют сведения о наличии способов обнаружения нескольких вложенных гипервизоров.

XXX ЗАГОЛОВОК XXX

Обзор и классификация способов обнаружения гипервизоров

В открытых источниках вопрос обнаружения гипервизора многократно обсуждался. На рис. 3 представлена классификация способов обнаружения гипервизоров, согласно которой все способы делятся на проактивные и сигнатурные.

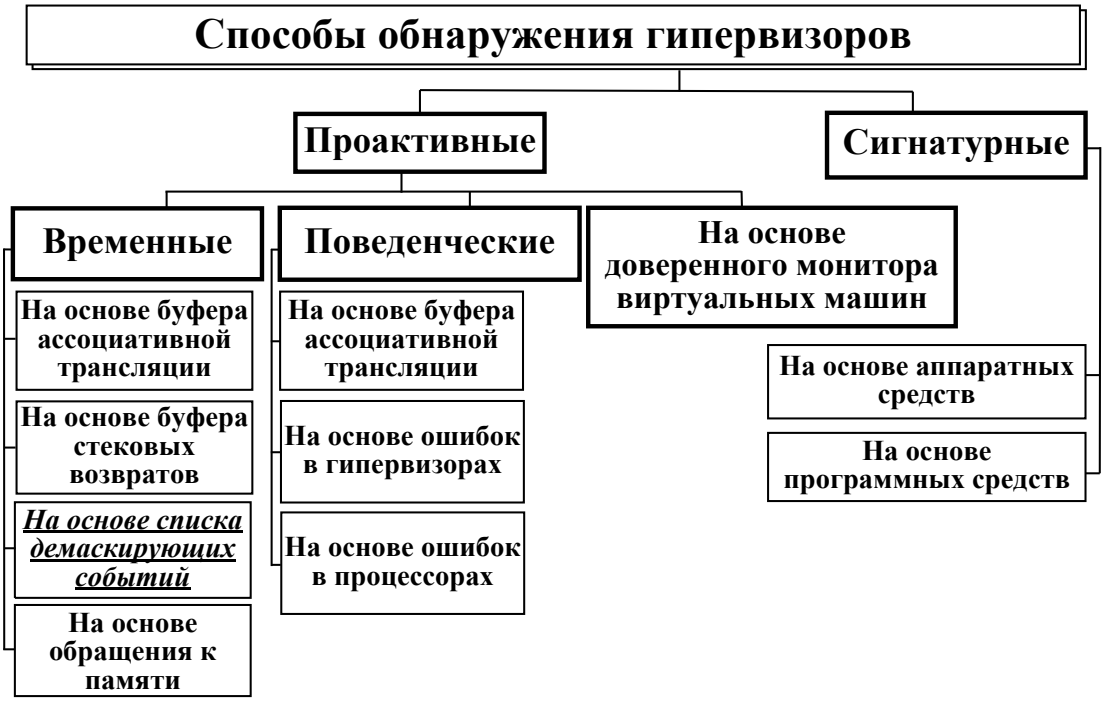


Рисунок 3 – Классификация способов обнаружения гипервизоров

Входящие в группу проактивных временны́е способы обнаружения основаны на том, что статистики времени обработки заданных событий гостевой ОС существенно зависят от того, загружен гипервизор или нет: в присутствии гипервизора длительность обработки событий значительно больше. Данная особенность была использована автором R_T_T при обнаружении китайского гипервизора (www.xakep.ru/post/58104/). Однако простые статистики позволяют сравнительно легко выявлять гипервизоры только в тех случаях, если нарушителем не предприняты меры по противодействию их обнаружения. В ситуациях, когда осуществляется целенаправленная компрометация счётчика, либо временная выгрузка гипервизора из памяти (так называемая технология «BlueChicken», которая использовалась в «BluePill»), известные временны́е способы не позволяют обнаружить гипервизор в системе.

Детальное описание и сравнительный анализ указанных способов обнаружения представлен в работе www.bit.ly/ik volume. Уделим внимание временному способу обнаружения с использованием списка демаскирующих событий.

Для выбранного способа таким событием гостевой ОС является выполнение инструкции, при котором управление всегда передаётся из ОС гипервизору. Одной из таких инструкций является CPUID. Система обнаружения гипервизоров (подчёркнутый курсив на схеме), которая будет описана далее, использует именно этот способ обнаружения.

Были проанализированы существующие средства обнаружения гипервизоров, результаты их сравнения представлены в табл.1. Под не скрытым гипервизором

подразумевается отсутствие в этом образце компонента, обеспечивающего противодействие обнаружению. Под скрытым образцом подразумевается наличие в этом образце такого компонента. Знаки «+» и «-» показывают наличие (отсутствие) соответствующей характеристики.

Таблица 1 – Сравнение существующих средств обнаружения гипервизоров

Наименование средства	**Способ обнаружения гипервизора**	**Возможность обнаружения гипервизора**		**Удобство использования и тиражирования**	**Обнаружение нескольких гипервизоров**
		не скрытого	**скрытого**		
Hypersight Rootkit Detector (North Security Labs, 2007–2011)	На основе доверенного монитора виртуальных машин	+	–	+	–
«Красная пилюля» (Луценко А. 2010 г.)		+	–	+	–
Symantec Endpoint Protection 12.1 2011 г.		+	–	+	–
McAfee Deep Defender 2011 г.		+	–	+	–
DeepWatch (Булыгин Ю. 2008 г.)	Сигнатурный на основе аппаратных средств	+	+	–	–
Copilot (Komoku, 2008 г.)		+	+	–	–
Экспериментальные образцы ПО	Временные и поведенческие способы на основе буфера ассоциативной трансляции и др.	+	–	+	–

С помощью анализа было установлено, что существующие способы обнаружения гипервизоров обладают рядом следующих недостатков:

- временны́е способы не позволяют выявить гипервизоры в случае использования компрометации счётчика тактов или временной выгрузки гипервизора из памяти;
- поведенческие способы не могут обнаруживать новые гипервизоры и не работоспособны на новых моделях процессоров;
- способы на основе доверенного монитора виртуальных машин уязвимы к атаке «человек-посередине» («Man-In-The-Middle»);

- сигнатурные аппаратные средства неудобны в использовании и тиражировании, а программные средства – нестойки к противодействию гипервизора;
- известные способы и средства обнаружения не позволяют обнаружить несколько вложенных гипервизоров.

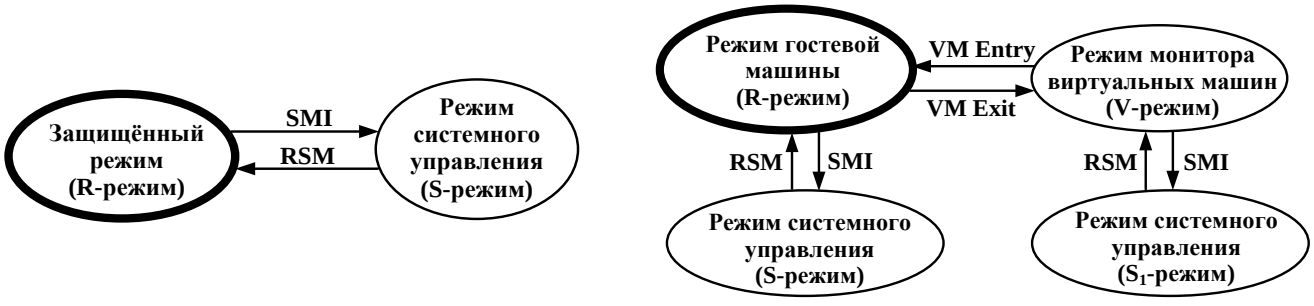
Далее представлена авторская методика обнаружения нелегитимного гипервизора, которая лишена указанных недостатков. Предполагается, что гипервизор внедрён одним из следующих способов:

- установкой драйвера операционной системы;
- модификацией главной загрузочной записи жёсткого диска;
- внесением изменений в микропрограмму аппаратного обеспечения, например, в BIOS (EFI).

При этом учитывается, что реализованный нарушителем гипервизор может противодействовать обнаружению посредством компрометации процессорного счётчика тактов, вре́менной деинсталляции из памяти, и недопущения копирования дампа памяти, содержащей структуры гипервизора.

ХХХ ЗАГОЛОВОК ХХХ
Предпосылки для обнаружения

Для выявления факторов, которые могут быть использованы для обнаружения гипервизора был проведён сравнительный анализ работы процессора с поддержкой аппаратной виртуализации при выполнении набора безусловно перехватываемых гипервизором инструкций (трассы) для случаев его отсутствия и присутствия (рис. 4, *а* и 4, *б*).



а *б*

Рисунок 4 – Схемы переключения между режимами работы процессора при выполнении набора безусловно перехватываемых инструкций в случаях отсутствия (а) и присутствия (б) гипервизора

В случае присутствия гипервизора возрастает не только среднее время выполнения трассы, но и его вариабельность. При этом, как показали выполненные нами эксперименты, если среднее время выполнения трассы для сокрытия гипервизора легко исказить (по средством искажения показаний счётчика тактов, то характеристики вариабельности целенаправленно исказить практически не возможно. Это обстоятельство и легло в основу предлагаемой методики обнаружения.
Подробный анализ схем переключения между режимами работы процессора и математическое обоснование приводится в работе «Модели выполнения процессорных инструкций в условиях противодействия со стороны нарушителя для компьютерных систем с поддержкой технологии аппаратной виртуализации» (www.bit.ly/10nPPlY).

XXX ЗАГОЛОВОК XXX
Методика обнаружения и её анализ

Суть данной методики обнаружения состоит в расчёте статистических характеристик длительности выполнения трассы и их сравнении с пороговыми величинами.
Измерения длительности выполнения трассы из 10-и инструкций CPUID проводились с помощью процессорного счётчика тактов TSC на повышенном 31-м уровне приоритета IRQL. Каждый опыт представлял собой выполнение трассы во вложенном цикле (внутренний цикл состоял из 1000 измерений, который повторялся 10 раз во внешнем цикле). Такие опыты выполнялись на различных ПК в двух режимах – при наличии и отсутствии гипервизора. Результатами опытов являлись матрицы размером 1000x10, содержащие данные измерений длительности выполнения трассы, по которым рассчитывались различные статистические характеристики и их пороговые значения для двух указанных режимов ПК для различных уровней фильтрации полученных данных. (Фильтрация выполнялась для недопущения искажений получаемых статистик выбросами в получаемых рядах значений длительности выполнения трассы).
Эксперименты также выявили наличие дрейфа получаемых данных по дням проведения опытов. Для обеспечения воспроизводимости в таких условиях опыты повторялись в течение нескольких дней до стабилизации получаемых статистик.

Для иллюстрации в табл. 2 приведены пороговые значения дисперсии $\bar{D}_f$ и момента 4-го порядка $\bar{M}_f$ времени трассы полученных на различных ПК для случаев отсутствия (ОТ) и присутствия (ПТ) гипервизоров.

Таблица 2 – Пороговые значения тест-статистик длительности трассы и их доверительная вероятность, полученные на различных ПК.

ПК	Тест-статистика	Уровень фильтрации f	Пороговое значение		Вероятности ошибок	
			ОТ	ПР	I рода, α	II рода, β
1	$\bar{D}_f$	0	≤ 14	≥ 18	0.02	0
	$\bar{M}_f$	0.1	≤ 679	≥ 947	0.02	0
2	$\bar{D}_f$	0.2	≤ 100	≥ 101	0.08	0.1
	$\bar{M}_f$	0.2	≤ 168	≥ 13030	0.14	0.02
3	$\bar{D}_f$	0	≤ 216	≥ 5478	0	0
	$\bar{M}_f$	0.02	≤ 54	≥ 956	0	0

В первом столбце табл. 2 номерами обозначены модели процессоров обследованных ПК: 1 – Intel Core 2 Duo E8200 с ОС Windows 7, 2 – Intel Core 2 Duo E6300 с ОС Windows 7, 3 – AMD Phenom X4 945 с ОС Windows Live CD XP (DDD). В первых двух ПК использовался разработанный автором гипервизор, реализованный в виде драйвера ОС, в 3-ем случае – специализированный гипервизор, получающий управление при загрузке ПК из BIOS. Исходный код авторского гипервизора находится на компакт диске.

Предлагаемая методика обнаружения нелегитимного гипервизора состоит из двух этапов: предварительного и оперативного (табл. 3). Детальное описание методики на сайте – http://bit.ly/ik_volume

Таблица 3 – Пошаговая методика обнаружения нелегитимного гипервизора

Название этапа	Содержание шагов
Предварительный	(1) Аппаратным образом записать доверенную микропрограмму в BIOS. (2) Установить операционную систему. (3) Получить пороговые значения для обнаружения гипервизора с помощью соответствующего алгоритма.
Оперативный, на стадии эксплуатации ПК	(4) Начать проверку ПК на отсутствие гипервизора с помощью алгоритма обнаружения. (5) Последовательно установить дополнительное ПО (MS Office и др.). (6) Следить за сообщениями об обнаружении гипервизора. (7) Для адаптации средства обнаружения к легитимному гипервизору выполнить шаг 3.

Напомним, что данная методика предназначена для обнаружения гипервизора в условиях противодействия с стороны нарушителя, которое выражается в целенаправленном искажении показаний счётчика тактов «скручиванием» его на

некоторую постоянную или переменную величину при попытке воспользоваться им для определения длительности выполнения трассы. Второе противодействие, если достигается достаточно точное искажение действительных статистических показателей вариабельности длительности выполнения трассы, представляет собой пока непреодолимую угрозу для аналитиков информационной безопасности. Однако, это пока только теоретическая угроза, поскольку реализовать на практике подобное противодействие из-за сложности его алгоритма и других причин не представляется возможным.

Разработка и размещение в нелегальном гипервизоре подпрограммы перехвата команд определения показаний счётчика тактов и с немедленным искажением их на постоянную величину – вполне реализуемая угроза: в наших исследованиях она имитировалась при отработки представляемой методики обнаружения гипервизора.

Недостатки методики и меры их преодоления приведены в табл. 4.

Таблица 4 – Недостатки и возможные решения методики обнаружения

Недостаток методики	**Комментарий / предлагаемое решение**
Вероятностное обнаружение гипервизоров	Путём повышения числа измерений можно добиться практически достоверного обнаружения.
Необходимость получения пороговых значений	Для работы средства обнаружения необходимо получать пороговые значения. Это типичная практика для информационной безопасности – какой-либо уровень считать доверенным и опираться на него.
Подсчёт числа вложенных гипервизоров	Для каждого устанавливаемого дополнительно гипервизора необходимо получать пороговые значения. Эксперименты проводились на двух гипервизорах. Не исключено что при повышении уровня вложенности, существующая схема обнаружения не сможет различить гипервизоры. Для решения задачи надо увеличивать число инструкций и повторов измерений.

Спектральный метод обнаружения гипервизоров в системах.
Детальный анализ получаемых массивов длительности выполнения трассы t на различных ПК выявил ещё одну закономерность: значения t распределяются, в основном, по постоянным уровням (рис. 5). Если отфильтровать редкие значения t, то в большинстве случаев оказывается, что в режиме ОТ таких уровней не более трёх, а в режиме ПР – три и более уровня.

Это значит, что каждый режим КС имеет свой частотный спектр длительности выполнения трассы, что можно использовать для выявления присутствия гипервизора в обследуемом ПК. Как и в предыдущей методике, для повышения достоверности уровневого метода следует выявлять пороговые значения режимов ОТ и ПР компьютерных систем.

Нетрудно видеть, что спектральный метод существенно проще рассмотренного выше метода вариационных статистик.

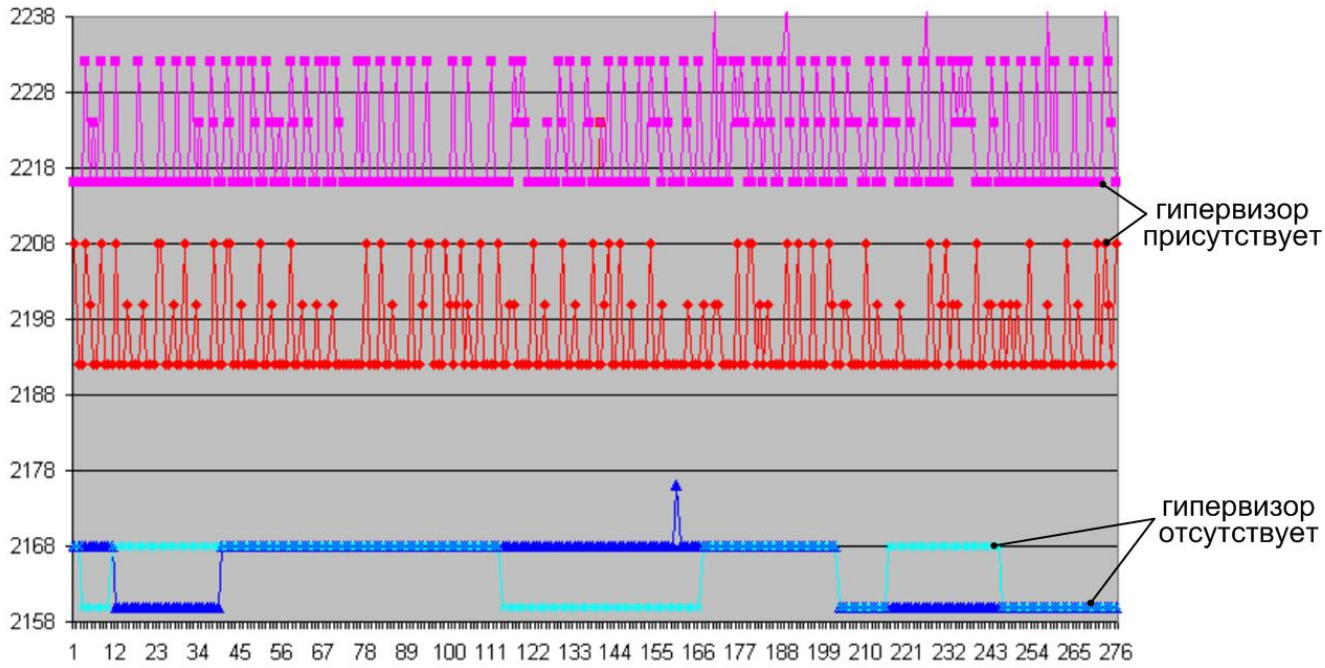


Рис 5 – Графики длительности выполнения трассы

а – для ПК в режиме ОТ и б – в режиме ПР

Видео, демонстрирующее работу данной методики при обнаружении авторского гипервизора в условиях компрометации счётчика тактов, представлено здесь - https://vimeo.com/65251365.

XXX ЗАГОЛОВОК XXX

Взгляд в будущее

Ясно, что предложенные методики обнаружения гипервизоров в системах актуальны на текущий момент. А что нам ждать от дня завтрашнего? По какому пути развития пойдут технологии создания руткитов? Можно дать волю воображению и с большой вероятностью предположить, что скоро можно будет увидеть следующее:

1. Руткиты в облаках и суперкомпьютерах

Зачастую для повышения производительности облачных вычислений используют аппаратную виртуализацию. При этом отсутствует информация о методах проверки таких систем на отсутствие нелегитимных гипервизоров. На недавней конференции РусКрипто'2013 в докладе А. Никольского "Уязвимости гипервизоров и систем облачных вычислений" (http://bit.ly/nikolvmm) отмечается, что существующие гипервизоры небезопасно использовать в облаке, а в докладе Д. П. Зегжды "Суперкомпьютеры и безопасность – новые задачи и новые возможности" (http://bit.ly/litezzz) отмечаются угрозы безопасности средств виртуализации в суперкомпьютерах.

Т. Келлерман (T. Kellermann), вице президент по кибер безопасности компании TrendMicro, отмечает в своей статье – "Observations on the Evolution of Cyber Tactics in 2013", что одним из главных векторов атак в 2013 году будут атаки на гипервизоры, используемые для организации облаков (http://bit.ly/150VFSu).

2. Руткиты во встраиваемых системах

В последнее время расширяется использование таких систем, как «Умный дом», «Умный город», системы управления в автомобилях и многие другие. Новая инициатива IBM Smarter Cities требует особого внимания, так как многочисленные компоненты системы обеспечивают жизнедеятельность большого количества людей, при этом представляя собой добычу для нарушителей. В статье "Manufacturers respond to car-hacking risk" из "Financial Times" (http://bit.ly/ftcarha) описывается угроза от внедрения вирусов в компьютерную систему автомобиля, так называемый «Car-hacking».

3. Руткиты в мобильных ОС

Внедрение руткитов в мобильные операционные системы уже давно не является новинкой.

4. Руткиты в виде предустановленного ПО в планшетах

Не могу снова не упомянуть работу R_T_T, посвящённую закладкам в военных ноутбуках Getac (bit.ly/Sf23yP). Там программные закладки были выполнены в виде софта от компании Compuware, именно той, которая выпускала мощный отладчик SoftICE. В настоящее время аналогичные закладки этой компании можно встретить в планшетах. К примеру, новые ThinkPad 2 с продвинутым уровнем защиты продаются уже с предустановленным ПО – "Enterprise-level security, with Trusted Platform Module and Computrace® Mobile" (http://bit.ly/lenovotmp).

5. Руткиты в комплектующих компьютерных систем

Странная ситуация складывается с этой проблемой. Весьма показательна в этом плане статья «Китайские закладки: непридуманная история о виртуализации, безопасности и шпионах» в журнале Хакер №12 (155) 2011. Ни на момент публикации этой статьи, ни потом должной реакции не последовало. Складывается впечатление, что от проблемы нашпигованности гипервизорами комплектующих для компьютерных систем просто отмахиваются или её целенаправленно замалчивают.

Ясно, однако, что данная проблема сама по себе не разрешится и её рано или поздно придётся решать. Остаётся вопрос времени и цены упущений.

XXX ЗАГОЛОВОК XXX

Вместо послесловия

Специалисты по ИБ, как и все люди, подвержены одним и тем же порокам: среди тех и других бытует мнение, схожее с поговоркой «Пока гром не грянет, мужик не перекрестится». Аналогичная ситуация и с нелегальными гипервизорами. Нередко можно слышать, что пока не набралась критическая статистика инцидентов, заниматься проблемой преждевременно, забывая при этом, что действие на опережение – это самый мудрый подход. Ясно же, что на всякую уязвимость в системе неизбежно найдётся свой нарушитель.

Поскольку угрозы использования нелегальных гипервизоров возрастают, этой проблемой в мире занимается всё большее число специалистов. Исследованиями в области сокрытия и обнаружения программных закладок занимаются в DARPA и IARPA (USA), DSTL (UK), DRDC (Canada), COSTIND (China) и других странах, существует множество частных разведок, таких как Kroll и G4S.

Следует также учитывать и то обстоятельство, что в мире набирает обороты кибервойны, так, например, в прошлом году Пентагон принял решение увеличить контингент киберкомандования с 900 до 4900 сотрудников. Министерство обороны Великобритании выделило £400 000 (более полумиллиарда долларов США) лабораториям DSTL на исследования в области киберугроз.

Россия похвастаться этим не может. Показательны в этом отношении работы R_T_T, в которых освещались случаи некомпетентности и нежелания российских структур разбираться в проблеме гипервизоров.

Российский аналог DARPA (фонд перспективных исследований) был создан в прошлом году. В начале 2013 года было приняты решения о разработке системы борьбы с кибератаками (В. В. Путин) и создании кибервойск (С. К. Шойгу).

Для ликвидации объективного отставания России в данной области необходимо опережающее совершенствование систем и средств защиты, параллельная разработка механизмов внедрения вредоносного кода и противодействия ему. Обеспечить такую работу может **специализированный центр**, основной задачей которого будет компенсировать недостатки в развитии и применении существующих и новых технологий с позиции информационной безопасности. Конкретные предложения по структуре и комплектованию данного центра имеются, но выходят за рамки представленной работы.

Впрочем, некоторые подвижки, хотя и робкие, в противодействии руткитам начались (в том числе с участием автора публикации).

XXXX

ВРЕЗКА

XXXX

Об авторе

XXXX

Игорь Коркин - кандидат технических наук по специальности 05.13.19 «Методы и системы защиты информации, информационная безопасность». Работает в МИФИ, является научным руководителем студентов, готовит аспирантов. Занимается обнаружением программных закладок более 6 лет, автор более 10 научных работ, победитель конкурса «Хакеры против форензики» на форуме «Positive Hack Days 2012».

XXXX

XXXX

БОКОВЫЕ ВЫНОСЫ

XXXX

INFO

XXXX

Согласно недавнему отчёту британской Национальной аудиторской службы (NAO) наблюдается рост числа киберпреступлений, которые одной Великобритании обходятся в £18 – £27 млрд. ежегодно (http://bit.ly/ukaudit).

XXXX

XXXX

WWW

XXXX

Интересную работу, посвящённую EFI руткитам, но под MAC OS выполнил Loukas K (bit.ly/Pe1Dkl).

XXXX

XXXX

WWW

XXXX

В 2012 центр Майкрософт по защите от вредоносных программ подготовил краткий отчёт об угрозе руткитов – http://bit.ly/ms_rootkit12

XXXX

Post scriptum

В дополнении хочется затронуть вопрос, возникающий у многих – "Что же делать после обнаружения руткита? Как его удалить? Можно ли программно вылечить систему?"

Однозначного ответа на эти вопросы нет.

Для руткитов, выполненных в виде компонентов ОС, например, процесса, драйвера, библиотеки dll, можно предложить частные решения для каждого случая отдельно.

Однако для руткитов, получающих управление из BIOS либо использующих иные аппаратные компоненты для работы, программные способы удаления предложить нельзя. Для анализа каждого случая необходима **специализированная лаборатория** с высококвалифицированными экспертами. Осуществляемые действия зачастую связаны с аппаратным вмешательством в работу заражённого оборудования.