Я не профессионал в области информационной безопасности, моя область интересов — это высокопроизводительные вычислительные комплексы. В тему ИБ я пришел совершенно случайно, и именно об этом дальше пойдет речь. Думаю, эта невыдуманная история гораздо лучше осветит проблемы, связанные с аппаратурой виртуализации, нежели сухое изложение фактов.

Китайские закладки

НЕПРИДУМАННАЯ ИСТОРИЯ О ВИРТУАЛИЗАЦИИ, БЕЗОПАСНОСТИ И ШПИОНАХ

Еще до официального анонса новых процессоров Intel с поддержкой аппаратной виртуализации (в начале 2007 года) я задумал использовать эти чипы для создания единой вычислительной системы на базе нескольких серверов, которая стала бы для ОС и прикладных программ единой вычислительной установкой с SMP-архитектурой. Для этого требовалось написать компактный гипервизор с нестандартным функционалом, главной особенностью которого было бы не разделение ресурсов единой вычислительной установки между разными ОС, а наоборот, объединение ресурсов нескольких вычислительных машин в единый комплекс, которым бы управляла одна ОС. При этом ОС не должна была даже догадываться, что имеет дело не с единой системой, а с несколькими серверами. Аппаратура виртуализации предоставляла такую возможность, хотя изначально не предназначалась для решения подобных задач. Собственно, система, в которой аппаратура виртуализации применялась бы для высокопроизводительных вычислений, не создана до сих пор, а уж в то время я вообще был первопроходцем в этой области.

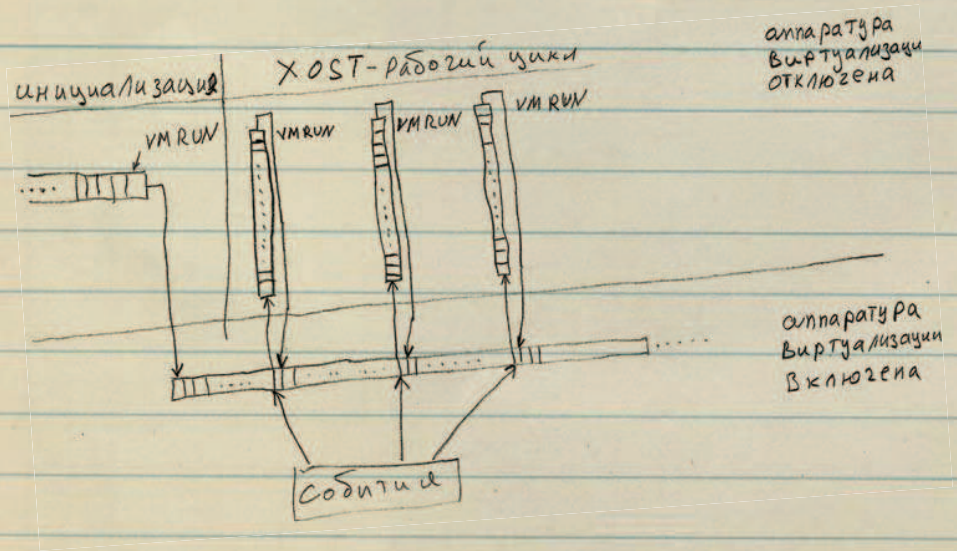

Схема 1. Архитектура виртуализации AMD

Гипервизор для этой задачи, конечно, писался с нуля. Принципиально важно было запускать ОС уже на виртуализированной платформе, чтобы с первых команд загрузчика ОС все работало в виртуальной среде. Для этого пришлось виртуализировать реальную модель и все режимы работы процессора и запускать виртуализацию сразу после инициализации платформы до загрузки ОС.

Поскольку система виртуализации для этой цели получилась нестандартной и выглядела как полностью автономный компактный программный модуль (объем кода не более 40–60 Кб), язык как-то не поворачивался называть ее гипервизором, и я стал использовать термин «гипердрайвер», поскольку он более точно передавал суть функционального предназначения системы.

Серийного оборудования с аппаратурой виртуализации в то время еще не было, однако благодаря сотрудничеству с фирмой «Крафтвей» я имел доступ к предсерийным образцам процессоров и материнских плат с поддержкой виртуализации, которые еще не выпускались официально (так называемым сэмплам, которые фирма Intel любезно предоставляет своим партнерам по бизнесу). Поэтому работа закипела на этом «сэмпловом» оборудовании.

Макет был собран, гипердрайвер написан, все заработало, как и было задумано. Нужно сказать, что на тот момент аппаратура виртуализации было очень «сырой», из-за чего она не один раз отказывалась работать так, как написано в документации. Приходилось разбираться буквально с каждой ассемблерной командой, а сами команды для аппаратуры виртуализации писать в машинных кодах, поскольку тогда не существовало компиляторов с поддержкой команд виртуализации.

Я гордился полученными результатами, чувствовал себя чуть ли не властелином виртуальных миров... но моя эйфория продлилась недолго, всего месяц. К тому времени я уже собрал макет на основе серверов с аппаратурой виртуализации, первые серийные образцы которых тогда как раз появились, но макет не работал.

Я начал разбираться и понял, что моя система виснет при выполнении команд аппаратной виртуализации. Создавалось такое впечатление, что они или совсем не работают, или работают как-то нестандартно. Зависание происходило только во время работы аппаратуры виртуализации в реальном режиме, если же моя система запускалась из защищенного режима, после загрузки ОС, то все было нормально.

Профессионалы знают, что в первых ревизиях аппаратура виртуализации Intel не поддерживала работу процессора в реальном режиме. Для этого требовался дополнительный слой достаточно большого объема для эмуля-

ции виртуального x86. Поскольку гипердрайвер запускался до загрузки операционной системы, чтобы она могла полностью поверить в новую виртуальную конфигурацию, то небольшой кусок загрузочного кода ОС выполнялся в реальном режиме работы процессора. Система умирала как раз на обработчиках эмуляции реального режима в гипердрайвере.

Сначала я подумал, что где-то ошибся, что-то не понял, о чем-то забыл. Проверил все до последнего бита в своем коде, никаких ошибок не нашел и начал грешить уже не на себя, а на коллег из-за бугра.

Первым делом заменил процессоры, но это не помогло. На материнских платах в то время аппаратура виртуализации была только в биосе, где она инициализировалась во время включения сервера, поэтому я начал сравнивать биосы на материнских платах (однотипных платах с сэмплами) — все совпадало до байта и номера самого биоса.

Я впал в ступор и, уже не зная, что делать, применил последнее средство — «метод тыка». Чего я только не делал, уже не думая, а просто комбинируя, и в конце концов тупо скачал биосы с официального сайта Intel и переписал их заново в материнские платы, после чего все заработало...

Моему удивлению не было предела: номер биоса был тем же самым, образы биоса совпадали побайтно, но по какой-то причине серийные материнские платы заработали только тогда, когда я залил в них такой же биос, взятый сайта Intel. Значит, причина все-таки в материнских платах? Но единственное их отличие было в маркировке: на сэмплах было написано Assembled Canada, а на серийных платах — Assembled China.

Стало ясно, что платы из Китая содержат дополнительные программные модули, прошитые в биосе, а стандартные программы анализа эти модули не увидели. Они, видимо, тоже работали с аппаратурой виртуализации и, соответственно, имели возможность скрыть истинное содержимое биоса. Стала понятна и причина зависаний моего гипердрайвера на этих китайских платах: две программные системы одновременно работали с одной и той же аппаратурой виртуализации, которая не позволяла разделять свои ресурсы.

Мне захотелось разобраться с этим зловредным биосом, причем без всякой задней мысли о «закладках», «бэкдорах», «недокументированных возможностях», был просто академический интерес, и не более того.

Нужно сказать, что параллельно с внедрением аппаратуры виртуализации Intel радикально обновила чипсет. Этот чипсет, получивший номер 5000x, выпускается в нескольких модификациях до сих пор. Южный мост этого чипсета, 631xESB/632xESB I/O Controller Hub, к которому подключены флеш-микросхемы с биосом, практически в неизменном виде выпускается с 2007 года и используется в качестве базового чипа почти для всех серверов в двухсокетном исполнении. Я скачал даташит на южный мост, прочел описание и просто обалдел. Оказывается, к этому новому южному мосту подключаются три микросхемы флеш-памяти: первая представляет собой стандартный биос, вторая выделена под программы процессора сетевого контроллера, а третья предназначена для интегрированного в южный мост блока BMC.

Блок менеджмента системы (BMC) — это средство удаленного управления вычислительной установкой и ее мониторинга. Он незаменим для больших серверных комнат, где из-за шума, температуры и сквозняков просто невозможно долго находиться.

То, что блоки BMC имеют собственный процессор и, соответственно, флеш-память для его программ, конечно, не новость, но до сих пор такие процессор и память выносились на отдельную плату, которая подключалась к материнской плате: хочешь — ставь, не хочешь — не ставь. Теперь Intel внедрила эти компоненты в южный мост, более того, подключила этот блок к системной шине и не стала использовать выделенный сетевой канал (как предусмотрено стандартом IPMI, описывающим функции блока BMC) для работы сервисной сети, а туннелировала весь сервисный сетевой трафик в основные сетевые адаптеры.

Далее я узнал из документации, что программы на флеш-микросхеме блока BMC зашифрованы, а для их распаковки используется специальный аппаратный криптографический модуль, также интегрированный в южный мост. Такие блоки BMC мне раньше не попадались. Чтобы не быть голословным, привожу выдержку из документации на этот южный мост:

- ARC4 processor working at 62.5 MHz speed.
- Interface to both LAN ports of Intel® 631xESB/632xESB I/O Controller Hub allowing direct connection to the net and access to all LAN registers.
- Cryptographic module, supporting AES and RC4 encryption algorithms and SHA1 and MD5 authentication algorithms.
- Secured mechanism for loadable Regulated FW.

Применение иностранных криптографических средств с длиной ключа более 40 бит запрещено на территории России законодательно, а тут — пожалуйста! — в каждом сервере Intel криптомодуль с неизвестными ключами длиной 256 бит. Более того, эти ключи использовались для шифрования программ, зашитых в микросхемы материнской платы на этапе производства.

Выходит, что блоки BMC в России на серверах Intel, которые имеют в своем составе чипсет 5000x, должны быть отключены. Однако эти блоки, напротив, всегда находятся в рабочем состоянии, даже если сама вычислительная установка отключена (для функционирования BMC достаточно дежурного напряжения, то есть вставленного в розетку кабеля питания сервера).

Все это казалось мне на тот момент второстепенным, поскольку для начала нужно было выяснить, в какой из флеш-микросхем находился программный модуль, работающий с аппаратурой виртуализации и мешающий моему гипердрайверу, и я начал экспериментировать с прошивками.

После ознакомления с документацией я насторожился, а когда обнаружил, что работоспособность гипердрайвера восстанавливается как раз после перепрошивки флеш-микросхемы блока BMC, даже не удивился.

Разбираться дальше без специальных стендов было невозможно, так как криптография полностью перекрывала возможности реверса кода для BMC. Документации на внутреннюю

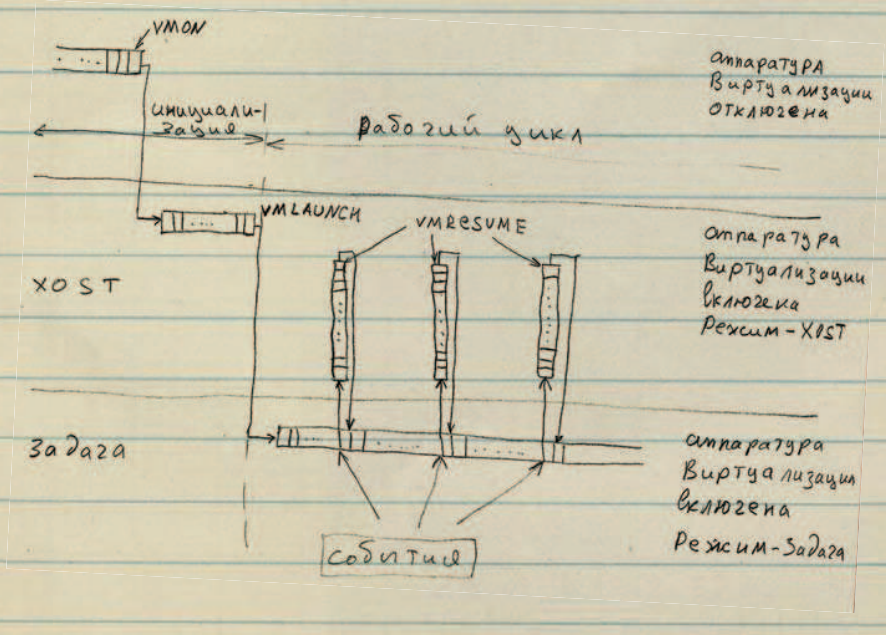

Схема 2. Архитектура виртуализации Intel

архитектуру этого интегрированного BMC я не нашел, в даташите на южный мост Intel описала только интерфейсные регистры для управления этим блоком по стандартным методам доступа, в результате чего получился классический «черный ящик».

Совокупность фактов настораживала и наводила на параноидальные мысли в стиле шпионских детективов. Эти факты однозначно говорили о следующем:

- В новых серийных серверных платах Intel на базе чипсета 5000 имеются программы, прошитые в флеш-памяти блока BMC и исполняемые на центральном процессоре, причем эти программы работают с использованием аппаратуры виртуализации центрального процессора.
- Образы флеш-памяти с официального сайта Intel не содержат таких программных модулей, следовательно, мешающие мне программные модули были нелегально прошиты в материнские платы на этапе производства.
- Флеш-память блока BMC содержит зашифрованные программные модули, которые невозможно собрать и залить в флеш-память без знания ключей шифрования, следовательно, тот, кто вставил эти нелегальные программные модули, знал ключи шифрования, то есть имел доступ фактически к секретной информации.

Я сообщил руководству «Крафтвей» о проблеме с прошивкой флеш-памяти блока BMC и сомнительной с точки зрения законодательства ситуации с новыми чипсетами Intel, на что получил вполне ожидаемый ответ в стиле «не мути, мешаешь бизнесу». Пришлось угомониться, поскольку против работодателей особо не попрешь.

Руки были связаны, но «мои мысли, мои скакуны» не давали мне покоя, было непонятно, зачем эти сложности и как все это сделано.

Если у тебя есть возможность разместить собственное ПО в памяти блока BMC, зачем тебе вся эта морока с центральным процессором? Разумной причиной могло быть только то, что решаемая задача требовала контролировать текущий вычислительный контекст на центральном процессоре. Очевидно, что уследить за обрабатываемой информацией на основной вычислительной системе, используя только периферийный низкоскоростной процессор с частотой 60 МГц, невозможно.

Таким образом, похоже, задача этой нелегальной системы состояла в съеме информации, обрабатываемой на основной вычислительной установке, с помощью средств аппаратуры виртуализации. Дистанционно управлять всей нелегальной системой, конечно, удобнее с процессора блока BMC, поскольку он имеет собственный независимый доступ к сетевым адаптерам на материнской плате и собственные MAC- и IP-адреса.

Вопрос «как это сделано?» имел более академический характер, поскольку кто-то умудрился создать гипервизор, умеющий разделять ресурсы аппаратуры виртуализации с другим гипервизором и делающий это корректно для всех режимов, кроме реального режима работы ЦП. Сейчас такими системами уже никого не удивишь, но тогда, пять лет назад, они воспринимались как чудо, кроме того, скорость эмуляции поражала — программно эмулировать хост без значительных потерь в быстродействии было невозможно.

```
----------------СТАРТ ТЕСТА-----------------
--------ИМЯ ТЕСТА----------СРЕДНЕ ВЗВЕШЕННОЕ ВРЕМЯ-------РЕЗУЛЬТАТ АНАЛИЗА---
Когерентность таймеров                299          Гипервизор не обнарухен
Команды MOV (CR0..Cr3)                235          Гипервизор не обнарухен
Команды MOV (DR0..Dr7)                237          Гипервизор не обнарухен
Команды RDMSR, WRMSR                  235          Гипервизор не обнарухен
Очистка буферов TLB                   234          Гипервизор не обнарухен
Команды виртуализации                 233          Гипервизор не обнарухен
Команды Ввода/Вывода                  233          Гипервизор не обнарухен
Монотонность доступа к ОП             259          Гипервизор не обнарухен
Монотонность доступа к БИОС           236          Гипервизор не обнарухен
Доступ в APIC                         235          Гипервизор не обнарухен
Доступ в NIC RAM                      234          Гипервизор не обнарухен
Выполнение прерываний                 236          Гипервизор не обнарухен
----------------Конец ТЕСТА-----------------
```

Рис.3 Время выполнения системных команд на сэмпловых платах

Для пояснения придется немного углубиться в теорию. Архитектура систем виртуализации Intel и AMD не предполагает наличия на платформе сразу нескольких гипервизоров, однако запущенный первым гипервизор может эмулировать для гипервизоров, которые запускаются после, работу на реальном оборудовании виртуализации. В этом случае все гипервизоры, запущенные вслед за первым, работают в эмулируемой среде хоста. Этот принцип я называю «правом первой ночи».

Его можно легко реализовать с помощью специальных обработчиков в корневом хосте, при этом режим задачи существенно не изменится, а хосты вторичных гипервизоров будут выполняться в режиме задачи для корневого хоста.

Режим эмуляции организовать не сложно, однако с быстродействием возникают проблемы. Аппаратура виртуализации работает в основном с блоком VMCB (VMCS), программы хоста постоянно обращаются к этому блоку, а на каждое такое обращение требуется 0,4–0,7 мкс. Скрыть такую программную эмуляцию хоста для системы виртуализации Intel практически невозможно, слишком много команд виртуализации придется эмулировать программно через выходы в корневой хост, вместо того чтобы выполнять их на реальном оборудовании.

Расскажу немного о различиях между архитектурами виртуализации. Системы аппаратной виртуализации от Intel и AMD совершенно не похожи друг на друга. Главное архитектурное отличие этих систем состоит в режиме работы хоста. В системе AMD хост работает с отключенной аппаратурой виртуализации, то есть его программы выполняются на реальном процессоре.

Виртуализация вторичного хоста в системах от AMD требует виртуализации только команды VMRUN (можно считать, что других команд нет). Работа с управляющим VMCB-блоком в архитектуре AMD происходит через обычные команды обращения к оперативной памяти, что позволяет контролировать с помощью вторичного хоста только выполнение команд VMRUN и подправлять при необходимости VMCB-блок перед реальным входом в режим задачи. Удлинить цикл обработки события в два раза еще можно, и на платформе AMD такая эмуляция жизнеспособна.

В системе виртуализации Intel все гораздо сложнее. Для доступа к VMCB-блоку используются специальные команды VMREAD и VMLOAD, которые нужно обязательно виртуализировать. Обычно обработчики хоста десятки, если не сотни раз обращаются к полям VMCB-блока, и каждую такую операцию нужно эмулировать. При этом заметно, что скорость падает на порядок, это очень неэффективно.

Стало ясно, что для эмуляции неизвестные коллеги использовали другой, более эффективный механизм. И подсказки насчет того, какой именно, я нашел в документации. Хост у Intel сам является виртуальной средой, то есть ничем, по сути, в этом плане не отличается от среды выполнения задачи и просто управляется другим VMCB (см. схему 2).

Кроме того, в документации описана концепция «дуального монитора» для виртуализации SMM-режима (режима системного менеджмента), когда фактически активны сразу два хоста и, следовательно, два VMCB-блока, причем хост, виртуализирующий режим системного менеджмента, контролирует основной хост как задачу, но только в точках вызова прерываний системного менеджмента.

Эта совокупность косвенных фактов говорит о том, аппаратура виртуализации Intel, вероятно, имеет механизм контроля множественных вторичных хостов, управляемых корневым хостом, хотя этот механизм нигде не описан. Кроме того, моя система именно так и работала, и другого объяснения практически незаметным действиям корневого гипервизора у меня до сих пор нет.

Стало еще интереснее: похоже, кто-то имел доступ к этим недокументированным возможностям и использовал их на практике.

Примерно за полгода до окончания сотрудничества с «Крафтвеем» я занял позицию пассивного наблюдателя, продолжая тем не менее регулярно запускать свою систему на новых партиях серийных материнских плат из Китая и новых сэмплах. На сэмплах все продолжало стабильно работать. Когда я переходил к китайским платам, в системе возникали все новые и новые чудеса. Было похоже, что коллеги из-за рубежа активно улучшали работу своего корневого гипервизора. Последние подозрительные партии плат вели себя практически нормально, то есть первый запуск моего гипердрайвера приводил к перезагрузке системы во время старта ОС, но все последующие запуски гипердрайвера и ОС проходили без сучка и задоринки. В конце концов случилось то, чего я давно ожидал: поступила новая партия серийных материнских плат, при использовании которых мой гипердрайвер вообще не зависал. Я уже начал сомневаться в своих параноидальных подозрениях, однако новый случай укрепил их.

Надо заметить, что фирма Intel активно совершенствует аппаратуру виртуализации. Если первая ревизия аппаратуры, с которой я начал работать, имела номер 7, то описываемая ситуация произошла на 11-й ревизии, то есть приблизительно за год ревизия обновлялась дважды (ревизии почему-то имеют только нечетные номера). Так вот, на ревизии с номером 11 условия выходов в хост по состоянию задачи для аппаратуры виртуализации существенно расширились, в соответсвтии с чем в VMCB-блоке даже было введено новое управляющее поле.

Когда появились сэмпловые процессоры с этой ревизией аппаратуры виртуализации, мне захотелось испытать новые возможности на практике. Я усовершенствовал гипердрайвер за счет новых возможностей 11-й ревизии аппаратуры виртуализации, установил сэмпловый процессор на серийную плату из Китая, в которой все уже работало без замечаний, и начал отладку.

Новые возможности аппаратуры никак себя не проявили, и я опять впал в прострацию, греша на сэмпловый процессор и документацию. Через некоторое время материнская плата потребовалась для другой задачи, и я, возобновив эксперименты, для подстраховки переставил процессоры с 11-й ревизией аппаратуры виртуализации на канадский сэмпл. Каково же было мое удивление, когда на этом сэмпле все заработало!

Сначала я подумал, что где-то накосячил с серийной платой, поскольку новые выходы в хост не имели к материнской плате ну совершенно никакого отношения, это чисто процессорная функция. Для проверки я переставил сэмпловый процессор на серийную плату, и все опять перестало работать. Значит, я ничего не накосячил, а проблема крылась в том, что материнская плата каким-то образом влияла на новые возможности аппаратуры виртуализации процессора.

С учетом моих подозрений напрашивался единственный вывод — нелегальный корневой хост коллег из-за рубежа, прошитый в флеш-памяти материнской платы, не знал о новой ревизии аппаратуры виртуализации. Когда эта неизвестная ему аппаратура начинала работать, он переставал корректно пропускать выходы из состояния задачи в мой вторичный хост через собственный обработчик событий.

Уже зная, как бороться с этой напастью, я залил на серийную плату прошивку для блока BMC с сайта Intel, включил систему в уверенности, что все сразу заработает, и снова выпадал в осадок, так как зависания остались. Это было что-то новенькое.

Согласно моей теории, нелегальный гипервизор обнаглел и уверился в своей неуязвимости. Видимо, его авторы посчитали, что этап обкатки их детище прошло и маскировать неотлаженное ПО под сбой биоса больше не нужно.

После того как была включена функция защиты кода инициализации от перезаписи во флеш-памяти, закладка стала практически неудалимой.

Уверенности в своей правоте у меня не было, нужны были контрольные эксперименты. Пришлось изобретать собственный метод для обнаружения аппаратного гипервизора. Потом, правда, оказалось, что я изобрел велосипед. Метод позволял контролировать время выполнения системных команд, требующих обязательной эмуляции в хосте гипервизора.

В качестве таймера я использовал циклический счетчик фреймов в аппаратуре USB-контроллера, а программу написал для реального режима работы, чтобы минимизировать побочные и неконтролируемые прерывания, которые маскировали истинное время выполнения системных команд. Первую проверку я провел для чистой системы на базе сэмпловых материнских плат из Канады (см. рис 3).

Время выполнения, указанное на фото, — это некоторое условное значение, приблизительно соответствующее такту процессора.

Затем я запустил тот же тест на серийной материнской плате и убедился в своих параноидальных предположениях — циклы выполнения команд существенно удлинились (см. рис. 4).

То есть во флеш-памяти блока BMC серверных плат из Китая, выпускаемых под лейблом Intel, имелся установленный на этапе производства недекларированный программный модуль, работающий как хост гипервизора.

Осталось убедить в этом окружающих. Первым делом я вышел на российского представителя Intel. Это было совсем нетрудно, поскольку сотрудники российского офиса часто появлялись в «Крафтвее».

Я все рассказал и показал, однако не был уверен, что технический специалист все понял. Эти так называемые технические специалисты по уровню своей компетенции мало отличаются от менеджеров. Однако он обещал доложить обо всем руководству. Не знаю, сделал ли он это, но никакой ответной реакции от Intel так и не последовало, все ушло как в песок.

Работа в «Крафтвее» к тому времени завершилась, и я начал новый проект в фирме, связанной с системами информационной безопасности.

Руководитель этой фирмы, с которым я поделился своими «открытиями», принял мои слова всерьез. В связи с этим было решено выйти на

```
----------------СТАРТ ТЕСТА----------------
--------ИМЯ ТЕСТА----------СРЕДНЕ ВЗВЕШЕННОЕ ВРЕМЯ-------РЕЗУЛЬТАТ АНАЛИЗА---
Когерентность таймеров              8273        Присутствие Гипервизора
Команды MOV (CR0..Cr3)              14023       Присутствие Гипервизора
Команды MOV (DR0..Dr7)              16191       Присутствие Гипервизора
Команды RDMSR, WRMSR                15831       Присутствие Гипервизора
Очистка буферов TLB                 19046       Присутствие Гипервизора
Команды виртуализации               14502       Присутствие Гипервизора
Команды Ввода/Вывода                16593       Присутствие Гипервизора
Монотонность доступа к ОП           14436       Присутствие Гипервизора
Монотонность доступа к БИОС         16787       Присутствие Гипервизора
Доступ в APIC                       17898       Присутствие Гипервизора
Доступ в NIC RAM                    13443       Присутствие Гипервизора
Выполнение прерываний               14055       Присутствие Гипервизора
```

Рис. 4. Время выполнения системных команд на серийных платах

руководство Центра защиты информации и спецсвязи ФСБ. Эта структура в составе ФСБ занимается обеспечением информационной безопасности в стране и регулирует деятельность государственных и коммерческих организаций, которые имеют отношение к защите информации. Она также регламентирует меры по защите информации для госструктур и коммерческих фирм, обрабатывающих секретную и конфиденциальную информацию.

Фирма, в которой я в то время работал, поддерживала с Центром официальные контакты, чтобы сертифицировать и лицензировать свои коммерческие проекты, поэтому организовать встречу на уровне специалистов было достаточно просто. Предполагалось, что эксперты Центра сообщат о своем мнении руководству, и если после этого руководство сочтет нужным выслушать нас, то следующим этапом будет встреча на более высоком уровне.

Встреча состоялась, я рассказал и показал все, что удалось выяснить, затем продемонстрировал наличие нелегального программного модуля на примерах плат из Канады и Китая. Кстати, тогда я в первый раз услышал и профессиональный термин «закладка», обозначающий такой модуль. Когда разговор зашел про BMC, в глазах у коллег из Центра появилось непонимание. Пришлось проводить ликбез. По ходу дела выяснилось, что они даже не подозревали о существовании специального микропроцессора в южном мосте с выходом на сетевой адаптер и о наличии в блоке BMC криптографического модуля, нарушающего российское законодательство.

В заключение мы совершенно неожиданно услышали, что эта модель угроз уже исследована, в их отношении применяется комплекс мер противодействия, и вообще, нам закладки не страшны, поскольку наши системы не имеют выхода в интернет.

Дальнейшие расспросы ни к чему не привели, все упиралось в секретность, типа, мы умные и суперграмотные, а вам ни о чем знать не положено. Однако я сильно сомневался в их технической грамотности, поскольку они просто не поняли большую часть того, что я рассказал и показал. Расстались на том, что они доложат своему начальству, а уж оно примет решение о дальнейших действиях.

Чуть позже я узнал, что это за «секретный метод» обнаружения программ хоста. Причем узнал совершенно случайно, во время переговоров на фирме — лицензиате Центра, уполномоченной проверять биос на закладки. Технические специалисты этой фирмы, проводящие исследования биоса, рассказали, что его программные модули, использующие аппаратуру виртуализации, надо искать по сигнатурам команд виртуализации.

Действительно, команды процессора для аппаратуры виртуализации содержат три-четыре байта и в программном коде, но кто сказал, что этот программный код они обнаружат в незашифрованном виде на флеш-микросхеме? Как они отсканируют этот код в оперативной памяти, если эти области памяти защищены аппаратно от просмотра?

В общем, результат первой встречи оставил неприятный осадок, и я в самом мрачном настроении ожидал развития событий. Месяца через полтора нас пригласили уже в сам Центр защиты информации и спецсвязи, чтобы мы продемонстрировали обнаруженную нами закладку.

На этот раз послушать нас собрались не рядовые сотрудники, а руководители и ведущие специалисты (по крайней мере, так они представились). Встреча превратилась в лекцию, меня внимательно слушали практически три часа, было видно, что они впервые слышат то, о чем я им рассказываю. Я перечислил новые уязвимости платформы x86, показал закладку и рассказал, как ее детектировать, ответил на множество вопросов. В конце нас поблагодарили, сказали, что тему нужно развивать в рамках специальных НИР, и на этом мы и расстались.

Эйфория улетучилась, когда по неофициальным каналам до нас дошла информация, что нам просто не захотели поверить. Однако это не охладило моего желания доказать свою правоту.

Как мне тогда казалось, решение лежало на поверхности: нужно было самому написать такой программный модуль закладки. Мне бы не удалось поместить закладку во флеш-память BMC, но в основной биос запихнуть ее я вполне мог. Я решил оснастить гипервизор модулем собственной безопасности для маскировки в памяти и на флеш-микросхеме, а также заблокировать запись во флеш-микросхему, где будет размещен код закладки, после чего удалить ее получится только путем выпаивания биоса и его перепрограммирования на внешнем программаторе.

Оставалось только определиться со «зловредной» функцией, которую следовало выполнять гипервизору. Я вспомнил утверждение одного из специалистов ФСБ о том, что им не страшны закладки, поскольку их системы отключены от глобальной Сети. Но информация из внешнего мира как-то должна попадать в эти защищенные локальные сети, хотя бы через одноразовые оптические диски. Таким образом, я пришел к очевидному выводу и решил анализировать входящий информационный поток в закладке средствами гипердрайвера, чтобы реализовать, так сказать, оружие судного дня, то есть использовать закладку для уничтожения вычислительной системы по внешней команде, передавая ее через входной информационный поток, стеганографически.

Просканировать информационный поток скрытно, без потери быстродействия, по зубам только аппаратуре виртуализации. В какой точке сканировать, тоже понятно: на буферах ввода/вывода дисковых систем и сетевого адаптера. Сканирование буферов ввода/вывода — плевая задача для аппаратуры виртуализации.

Сказано — сделано! Такой гипердрайвер размером около 20 Кб был прописан в биос материнской платы и оснащен функцией защиты от обнаружения. Он блокировал попытки его перезаписи при обновлении биоса и выполнял единственную функцию: обнулял флеш-микросхему биоса при поступлении команды на уничтожение.

Сама команда для простоты реализации была зашита в текстовый файл DOC-формата в тегах настройки.

Когда все было готово, руководство фирмы опять вышло на ФСБ с предложением посмотреть работу нашей собственной закладки и убедиться в том, что технологии виртуализации представляют реальную угрозу. Но посмотреть на нашу закладку в деле никто не захотел, с самого верха поступила команда (я так и не узнал, чье именно это было распоряжение) с нами больше не общаться.

Главные борцы за информационную безопасность не захотели нас слушать. Тогда, уже практически ни на что не надеясь, фактически для очистки совести, мы попытались донести информацию о проблеме до пользователей систем информационной безопасности. Мы связались с «Газпромом», чтобы проинформировать специалистов компании о современных угрозах для распределенных систем управления технологическими процессами. Удалось организовать встречу с руководством службы корпоративной защиты и управления комплексными системами безопасности этой корпорации. Специально для них была подготовлена более наглядная версия закладки с упрощенным командным интерфейсом. Закладка активировалась после загрузки на компьютер текстового файла, содержимое которого включало два слова — «Газпром» и «стоп», — расположенных в произвольном порядке. После этого компьютер умирал, но не сразу, а с задержкой в пять минут. Естественно, можно было сделать задержку и на сутки, но тогда мы бы не уложились во время, отведенное для демонстрации. Сотрудники «Газпрома» посетовали на низкий уровень информационной безопасности и заявили, что это не их дело, поскольку они руководствуются требованиями и правилами, которые устанавливает ФСБ. Круг замкнулся, стало понятно, что эту монолитную систему «информационной безответственности» не пробить.

За три с лишним года, которые прошли с тех пор, я ни разу не слышал, чтобы кто-нибудь говорил об аппаратуре виртуализации как об инструменте проникновения в целевые системы. Парадокс? Не думаю. Специфика темы в том, что мы узнаем только о провальных технологиях. О технологиях, которые не обнаружены, мы не знаем, а их авторы, конечно, молчат.

Нужно учитывать, что надежное размещение закладок в биосе возможно только в заводских условиях. В условиях эксплуатации для этого придется ориентироваться на определенную модель материнской платы, а такие варианты не слишком интересны хакерам. Им нужна массовость, они работают, что называется, «по площадям». Однако существуют и те, кто атакует прицельно, «по-снайперски». Технологии размещения закладок в биосе, да еще и с активацией аппаратуры виртуализации, которая позволяет эффективно скрыть их, — это, конечно, удобный инструмент для таких «снайперов».

Один раз их почти поймали, причем практически случайно. Думаю, сейчас этого сделать уже не удастся, да и ловить, как ты, наверное, понял, некому. ⊐⊏