побег из-под vm ware

крис касперски ака мыщъх, no-email

многие хакеры и программеры ~~системные администраторы~~ гоняют сомнительные программы под VM Ware и прочими эмуляторами, считая, что они надежно защищены, однако, это не так! зловредный код может вырваться из эмулятора и покоцать основную систему. мыщъх детально исследовал этот вопрос и предлагает несколько ~~эффективных~~ сценариев возможных атак

введение

Во времена MS-DOS/9x для экспериментов с вирусами приходилось держать на столе несколько компьютеров или переключаться на специальный жесткий диск, что было крайне ~~неудобно~~утомительно. Народ с тоскою поглядывал в сторону NT, гибкая система безопасности которой позволяла творить чудеса, например, разрешала процессу изменять только специально подсаженные файлы-дрозофилы. Увы! Большинство вирусов не работало под NT! К тому же, ~~подсистема~~ защит~~ы~~а оказалась крайне ненадежной и хакеры научились ее обходить (например, эмулировать ввод с мыши/клавиатуры, посылая команды более привилегированному окну).

Рисунок 1 несколько компьютеров при работе с вирусами в эпоху ранней MS-DOS были не роскошью, а необходимостью

С появлением виртуальных машин (VM Ware, Virtual PC) появился и соблазн использовать их как "загон" для вирусов и червей, что очень удобно. Вместо возни с мониторами, корпусами, жесткими дисками и проводами, десяток "системных блоков" свободно размещается в нашей хакерской норе, к тому же некоторые эмуляторы (например, BOCHS) содержат встроенные отладчики, уверенно работающие там, где soft-ice и olly уже не справляются.

Рисунок 2 загон для вирусов по-американски

Весь вопрос в том — насколько это надежно. Гонять живого червя на эмуляторе. А вдруг он вырвется за его пределы? Анализ червей, выловленных в дикой природе, показывает, что многие из них уверенно распознают наличие эмулятора, отказываясь на нем запускаться, в результате чего червь имеет хорошие шансы пройти незамеченным. Но хакерская мысль не стоит на месте, пытаясь вырываться из-за стенок виртуальной машины.

Рисунок 3 особенности национальной охоты на вирусы или загон для вирусов II

Теоретически это вполне возможно. Эмуляторы (особенно динамические, т. е. такие, которые часть команд выполняют на "живом" процессоре) не свободны от ошибок. Привилегированные команды (типа обращения к портам ввода/вывода) отлавливаются эмуляторами достаточно надежно и никаких граблей здесь по обыкновению нет, но существует реальная угроза записи в адресное пространство процесса-эмулятора при выполнении "обычных" инструкций. Конечно, модификации подвергается не код, а данные, но если среди этих данных окажется хотя бы один указатель (а он ~~наверняка там~~ окажется там наверняка), нашу хакерскую задачу можно считать решенной.

Рисунок 4 разработка вирусов требует глубоких познаний системы, вдумчивого подхода к делу и глубокой медитации

Единственная проблема в том, что такая дыра (даже если она действительно будет обнаружена), успеет заткнуться быстрее, прежде чем получит большое распространение, к тому же многообразие существующих эмуляторов значительно уменьшают шансы червя на успех.

Рисунок 5 червь, вырвавшийся из застенок виртуальной машины на свободу

Отбросим гипотетические дыры и сосредоточимся на универсальных методиках, работающих практически под любым эмуляторо~~в~~м и ~~эксплуатирующих~~эксплуатирующие уязвимости концептуального уровня, которые не так-то просто закрыть. Мыщъх предлагает три сценария атаки: а) ***проникновение через виртуальную сеть***, б) ***back door-интерфейс эмулятора*** и в) ***внедрение в folder.htt в shared folders***.

Рассмотрим эти механизмы поподробнее.

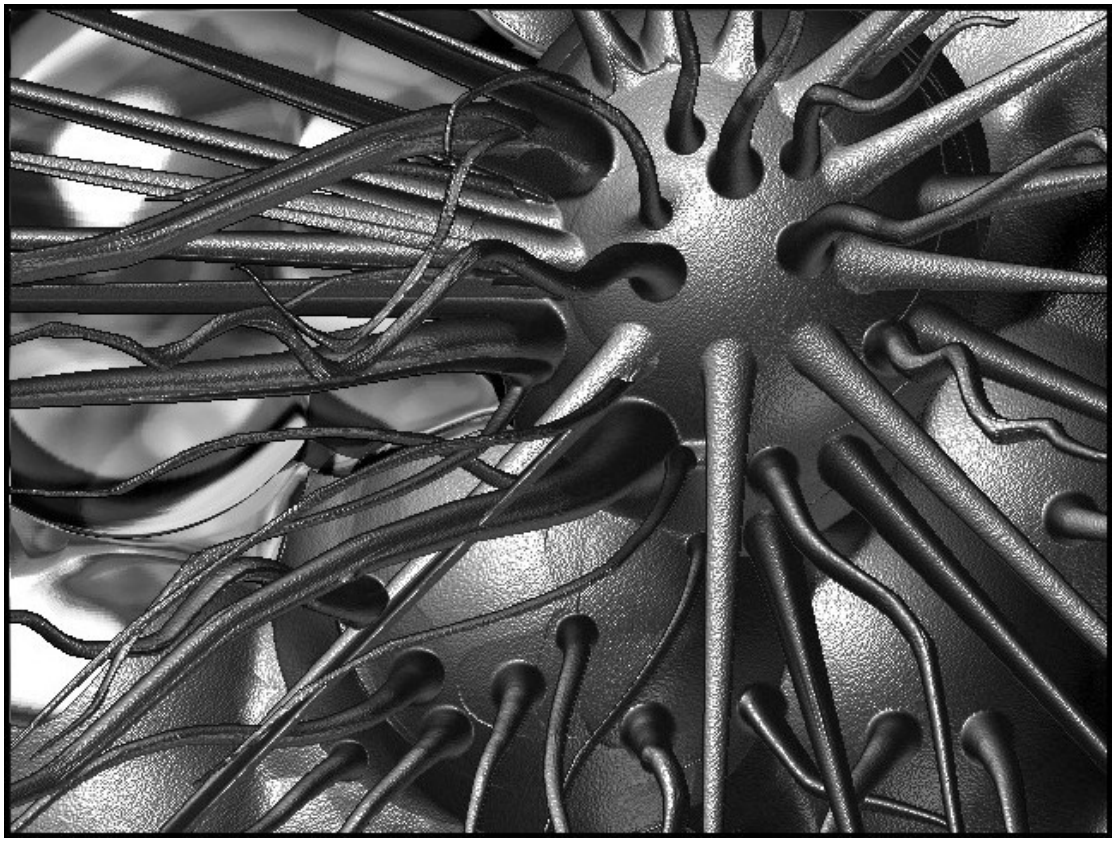

Рисунок 6 в центре вируса

атака через виртуальную сеть

Практически все эмуляторы поддерживают виртуальную сеть, связывающую гостевую (guest) и основною (host) системы невидимым кабелем. В эмуляторах типа QEMU она поднимается сразу, в VM Ware — только после соответствующей настройки виртуальной машины, но обычно эмулятор конфигурируется с сетью, потому что это самый удобный способ обмена данными. К тому же, на базе той же VM Ware можно легко ~~построить~~ воздвигнуть honeypot, своеобразный "капкан" для вирусов и червей, заползающих из Интернета.

Рисунок 7 виртуальный сервер виртуальной сети, существующей только в сознании эмулятора

Если основная операционная система доступна по сети и в ней имеются не залатанные дыры (типа дыр в DCOM RPC или TCPIP.SYS), ее можно свободно атаковать из-под эмулятора так же, как и по настоящей сети. Разница лишь в том, что большинство персональных брандмауэров не отслеживают локальные подключения и не препятствуют им, то есть эмулятор позволяет хакеру подключаться к тем ресурсам, доступ к которым извне компьютера надежно закрыт! При организации honeypot'ов это очень актуально! Допустим, основная система содержит shared-ресурсы, доступные только изнутри локальной сети, и для удобства не имеющие паролей, тогда виртуальная машина становится своеобразным "мостом" (или, если угодно proxy-сервером) между хакером/червем и основной системой!

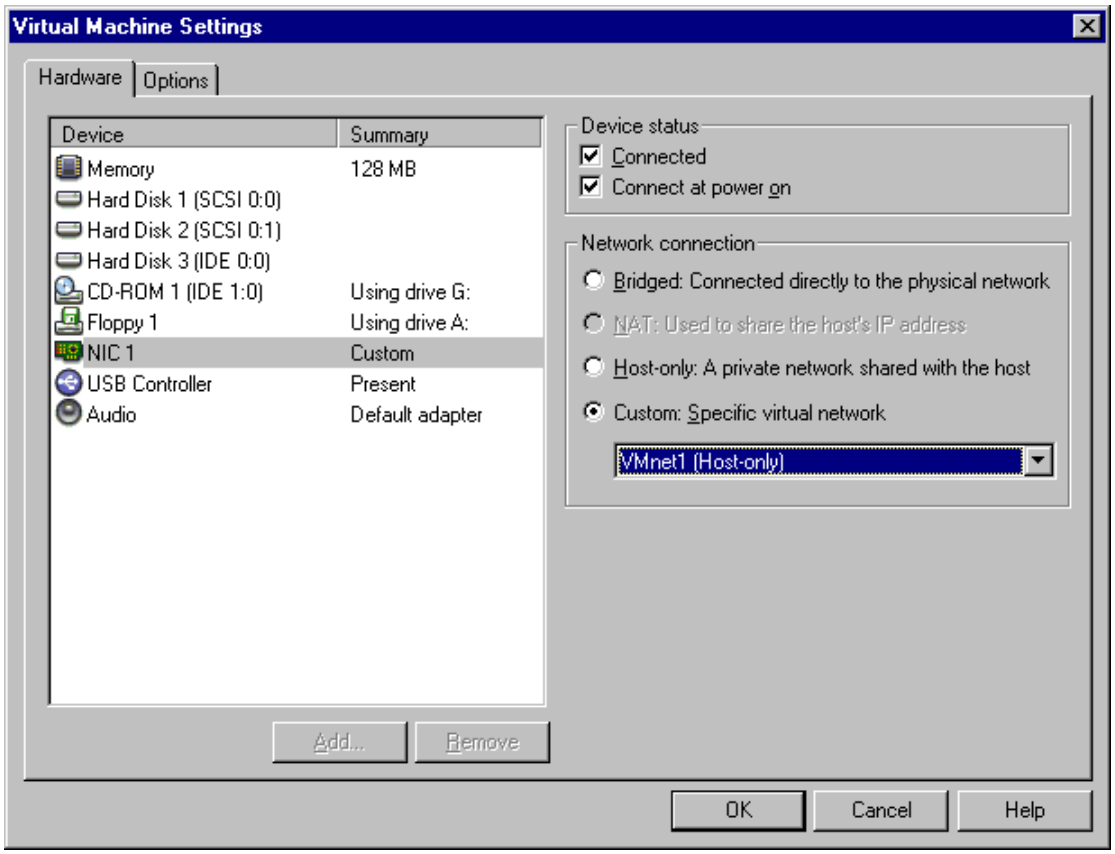

Рисунок 8 настройка виртуальной сети в среде эмулятора VM Ware

Как защититься от этой атаки? Самое простое — снести виртуальную сеть, а весь обмен данными с гостевой системой вести через дискету/cd-rom. Чтобы не возиться с прожиганием CD-R/RW болванок, можно использовать виртуальные iso-образы, только это все равно не выход! Значит, потребуется своевременно устанавливать свежие заплатки на основную систему, установить пароли на все shared-ресурсы и удалить с основной машины все службы, доступ к которым нежелателен, либо же убедиться, что персональный брандмауэр отслеживает локальные подключения и блокирует их.

Рисунок 9 еще один виртуальный сервер

атака через folder.htt

Эмулятор VM Ware предоставляет еще один способ обмена данных между виртуальной машиной и основной операционной системой — shared folders (общие папки). При настойке гостевой машины, администратор открывает доступ к одному или нескольким каталогам основной системы и виртуальная машина "видит" их в своем сетевом окружении.

Рисунок 10 folder.htt-файлы позволяют червям вырываться на волю

Механизм общих папок работает в обход виртуальной сети (которой, может быть вообще не установлена) и в плане защиты очень надежен, однако, атаковать его все-таки возможно! Как известно, начиная с Windows 98, "проводник" поддерживает пользовательский стиль папок, управляемый файлом folder.htt. Это обыкновенный http-шаблон, "переваривающий" не только теги, но и скрипты. Известно множество VBS-вирусов, размножающихся именно этим путем.

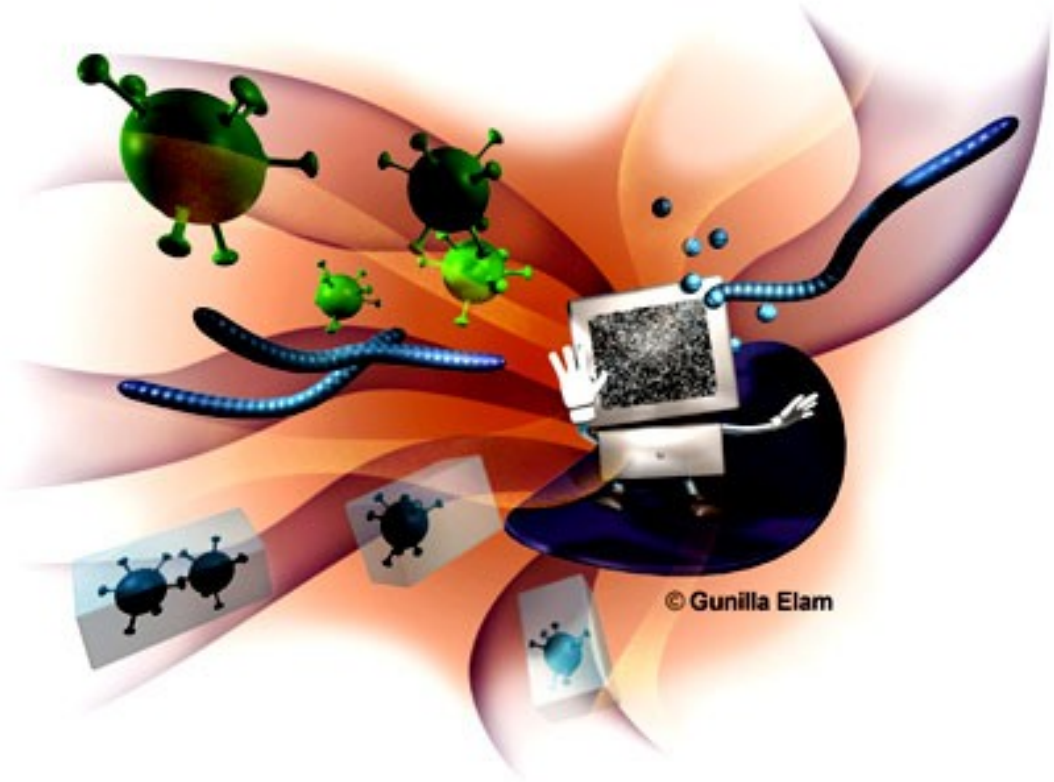

Рисунок 11 виртуальный компьютер – рассадник вирусов

Что произойдет, если зловредный код, исполняющийся под эмулятором, создаст собственный folder.htt файл (или внедриться в уже существующий)? При первом же открытии общей папки Проводником основной системы, скрипт, содержащейся в folder.htt, получит управление, запуская вируса в свои владения! И это не единственный путь!

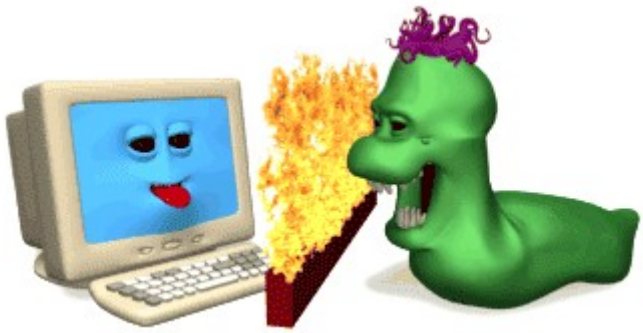

Рисунок 12 надежный брандмауэр должен защищать основной компьютер от атаки через виртуальную сеть и shared-folders, но увы… ни один из известных мыщъх'у брандмауэров надежным не является

Вирус может создать desktop.ini, указав, что папка используется для хранения изображений, тогда при ее открытии, Проводник автоматически отображает миниатюры. Известно по меньшей мере три фатальные ошибки Windows, приводящие к возможности передачи управления на машинный код — в bmp, jmp и wmf файлах. И хотя, соответствующие заплатки были выпущены еще черт знает когда, множество машин остаются уязвимыми и по сегодняшний день.

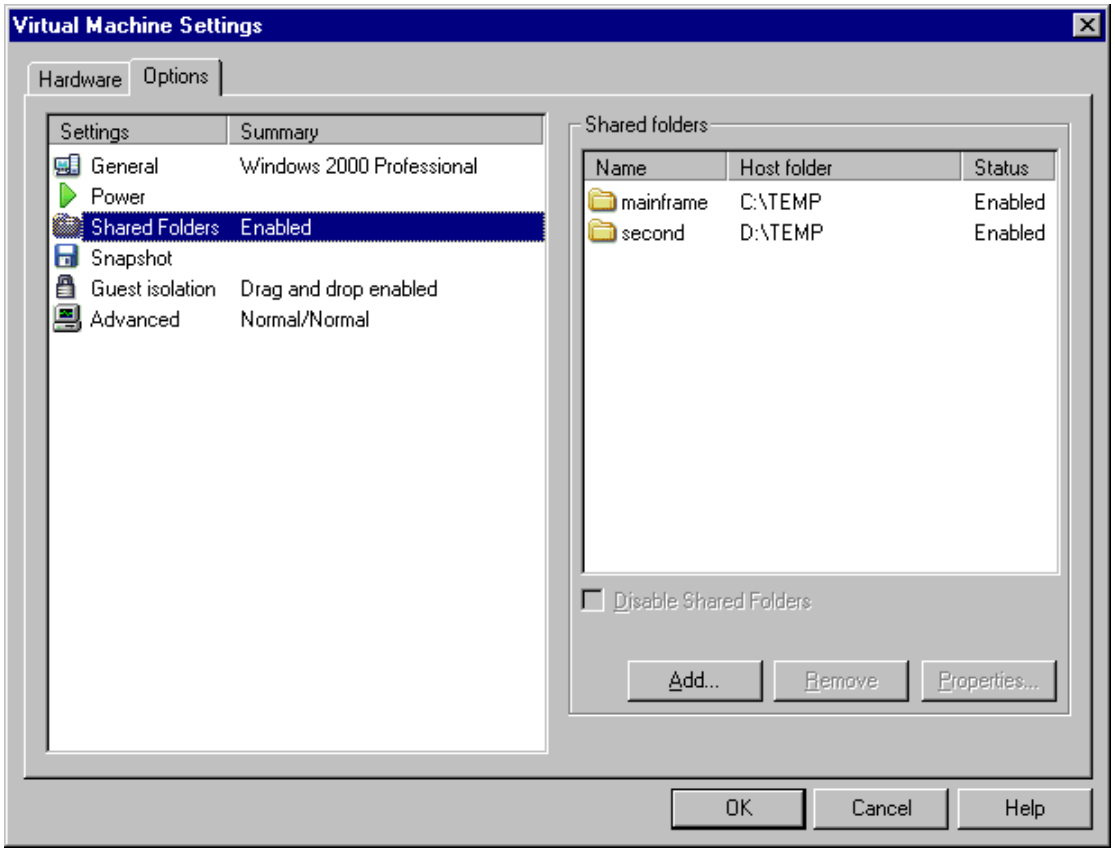

Рисунок 13 настройка общих папок в среде VM Ware

Защититься от атак данного типа очень просто — забейте на Проводник и пользуйтесь только FAR'ом (или на худой конец — Total Commander'ом) и периодически проверяете общие папки на вшивость (даже если лично вы никогда не пользуетесь Проводником, это еще не означает, что им не пользуются остальные и существует вероятность, что общую папку откроет кто-то другой).

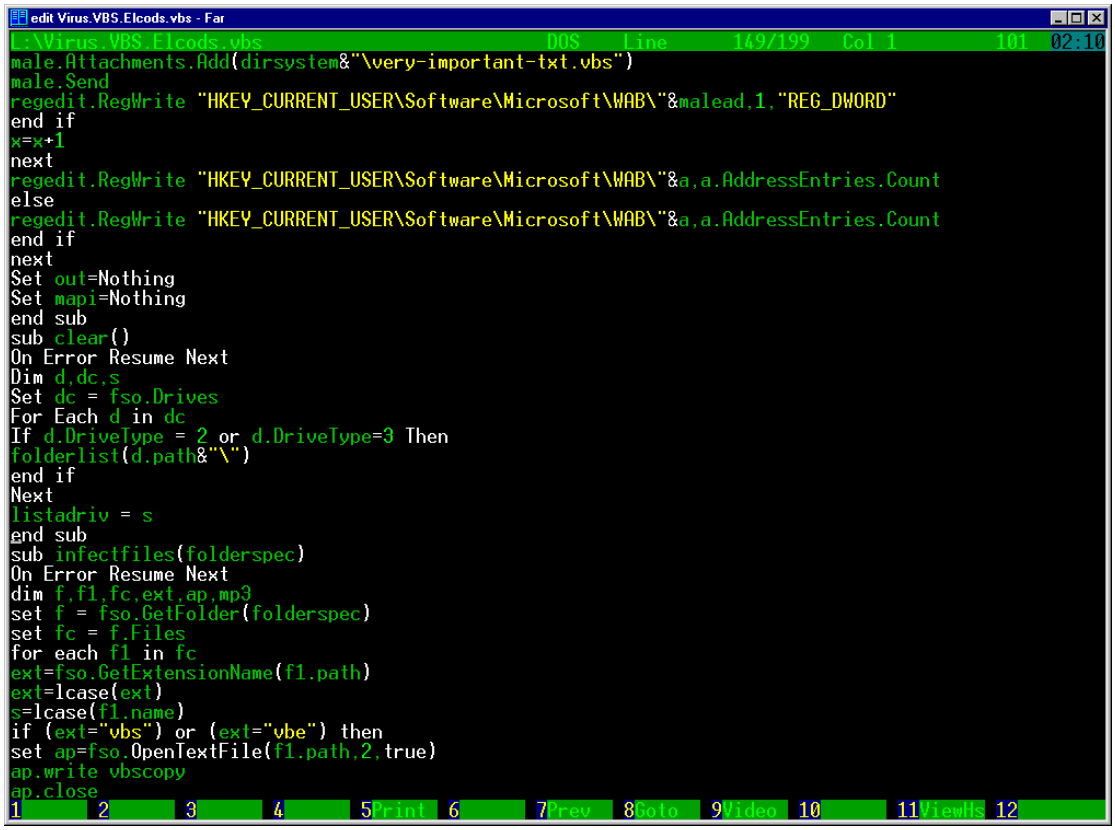

Рисунок 14 фрагмент исходного текста вируса, написанного на VBS

атака через back door

Для управления виртуальной машиной многие эмуляторы используют специальный (и, по обыкновению, недокументированный) back door механизм вроде того, что есть в soft-ice (см. INT 03h в Interrupt List'e Ральфа Брауна). Virtual PC использует для той же цели инвалидные инструкции процессора (например, 0Fh 3Fh 07h 0Bh), а VM Ware "магический" порт ввода/вывода.

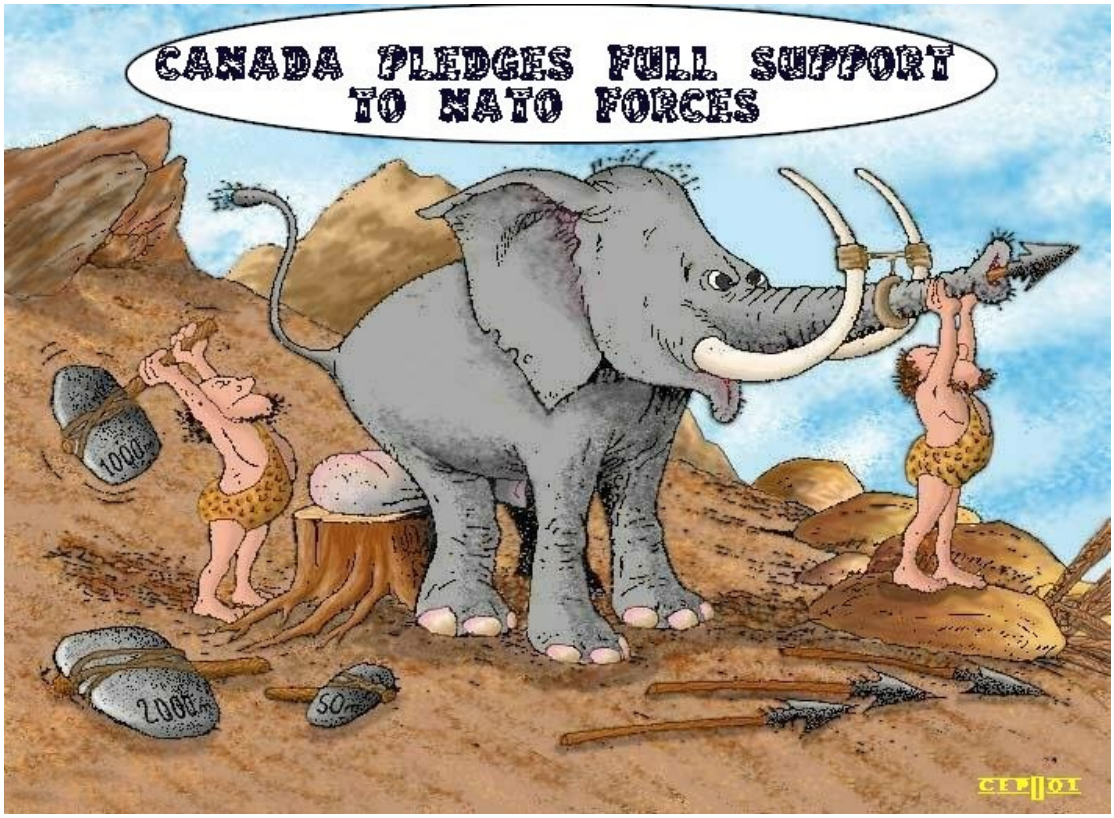

Рисунок 15 back-door интерфейс – мощное оружие, бьющие точно в цель

Остановимся на VM Ware как на самом популярном эмуляторе. Чтобы передать back door команду на выполнение, необходимо выполнить следующие действия:

- в регистр EAX занести магическое число 564D5868h ('VMXh' в ASCII-представлении);
- в регистр DX занести магическое число 5658h (номер порта, 'VX' в ASCII);
- в регистр CX занести номер команды, а в регистр EBX ее параметры;
- выполнить команду IN EAX, DX (or OUT DX, EAX);
- если программа исполняется не под VM Ware (или VM Ware был предварительно пропатчен) на прикладном уровне защищенного режима возникнет исключение типа "нарушение доступа";
- при выполнении под VM Ware регистр EBX будет содержать магическое число 564D5868h ('VMXh' в ASCII-представлении), а в остальных регистрах — возвращенные данные (если они есть);

VM Ware поддерживает большое количество самых различных команд, подробно исследованных Ken'ом Kato и описанных в его статье "VMware's back" (http://chitchat.at.infoseek.co.jp/vmware/backdoor.html). Здесь можно найти и установку даты/времени, и работу с буфером обмена и даже механизм удаленного вызова процедур (RPC), но… потенциально опасных команд среди них нет. Вирус не может просто взять и вырываться из виртуальной машины! Или… все-таки сможет? Свыше двух десятков команд еще остаются неисследованными и неясно зачем они и почему. Никто не знает какие возможности нас ждут…

Рисунок 16 вороне где-то бог послал персональный компьютер

Из всех команд, исследованных на сегодняшний день, самой опасной была и остается 0Ch (Connect/disconnect a device), отвечающая за подключение/отключение IDE, SCSI и USB устройств. У вируса существует шикарная возможность подключить физический диск основной системы и нагадить на нем по полной программе (VM Ware позволяет создавать виртуальные диски на основе физических). Еще вирус может дотянуться до USB-"свистка" и заразить все имеющиеся на нем исполняемые файлы, которые кто-нибудь обязательно запустит на основной машине.

Рисунок 17 виртуальная машина по сути своей черепаха

Короче, возможностей много. Для защиты рекомендуется пропатчить VM Ware, изменив магический номер на что-то еще. Неофициальная заплатка лежит здесь: http://honeynet.rstack.org/tools/vmpatch.c, официальных пока нет и, по-видимому, в обозримом будущем и не предвидится. (Однако, даже залатанная система по-прежнему остается уязвимой, поскольку подобрать нужные магические числа можно и брут-форсом, возможных вариантов не так уж и много — 16-битный номер порта, плюс 32-битный "пирожок" дают менее 48-значимых битов! "менее" — это за вычетом стандартных номеров портов, которые нельзя использовать).

Ниже в качестве примера приводится программа, определяющая версию VM Ware.

```
#include <windows.h>

// строковые константны
#define VM                      "vmware"
#define VM_DETECTED             "detected"
#define VM_NOT_DETECTED         "not detected"
#define VM_NOT_RECOGNZD         "detected, but not recognized"

// под vm-ware функция возвращает версию vm-ware в регистре eax
// (нуль — это не vm-ware или версия неопознана),
// без vm-ware возбуждается исключение
__declspec(naked) get_vm()    // "голая" функция без пролога и эпилога
{
__asm{
        ; подготавливаем аргументы и магические пирожки
        mov ecx, 0Ah            ; номер команды — определение версии
        mov eax, 564d5868h      ; 'VMXh' - магический номер типа "пирожок"
        mov edx, 00005658h      ; '..VX' - магический порт back-door интерфейса

        ; дергаем за "веревочку"
        in eax, dx              ; вызываем команду по back-door интерфейсу
                                ; возвращенные параметры помещаются в EAX/EBX/ECX

        ; внимание!
        ; в среде чистой Windows без vm-ware при обращении к порту ввода-вывода
        ; произойдет исключение и управление будет передано SEH-обработчику,
        ; который должен быть заранее установлен (иначе выполнение программы
        ; будет завершено системой)
        ; если же мы еще здесь, следовательно, исключения не произошло
        ; и, либо какой-то хитрый драйвер открыл порты ввода-вывода,
```

```
        ; либо мы находимся под управлением непатченной vm-ware
        ; или чего-то очень на нее похожего
        cmp ebx, 'VMXh'         ; анализируем возвращенный магический пирожок
        je under_VMware         ; если пирожок возвращен, мы под vm-ware

        xor eax,eax             ; возвращаем ноль в знак того, что мы не под vm-ware!
        ret                     ; выходим из функции

under_VMware:
        ret                     ; мы под непатченной vm-ware, в eax номер версии
        }
}

main()
{
        // вызываем функцию get_vm из блока __try,
        // чтобы отлавливать возникающие исключения
        // для простоты и наглядности, версия vm-ware, возвращенная get_vm,
        // не выводится на экран и сообщается лишь о том, был ли обнаружен
        // непатченный эмулятор или нет (если исключения не произошло,
        // vm-ware считается обнаруженной)
        __try { printf("%s %s\n",VM, (get_vm())?VM_DETECTED:VM_NOT_RECOGNZD);}

        // обработчик исключения, получающий управление при выполнении программы
        // в среде чистой Windows, под патченной vm-ware или на другом эмуляторе
        __except(1) {printf("%s %s\n",VM, VM_NOT_DETECTED);}
}
```

Листинг 1 программа, демонстрирующая взаимодействия с виртуальной машиной через back-door интерфейс и определяющая версию VM Ware (на экран она не выводится)

прочие способы

Для обмена мелкими порциями данных между виртуальной машиной и основной системой удобно использовать старый добрый гибкий диск. Просто даем эмулятору физический доступ к устройству A: (B:)и все! В смысле кранты! Если вирус внедрит в boot-сектор зловредный код, дискета окажется забытой в дисководе и этот дисковод будет первым загрузочным устройством в BIOS Setup, когда-нибудь зловредный код получит управление и сможет поразить жесткий диск основной системы.

Существуют и другие сценарии проникновения, однако они еще менее жизнеспособны и потому здесь не рассматриваются.

Рисунок 18 дьявол, юзеры, системные администраторы и все-все-все

заключение

Эмулятор — это очень удобная вещь, однако, от разведения вирусов в недрах виртуальной машины мыщъх советует воздержаться: "скорлупа", отделяющая гостевую систему от реального мира, слишком тонка и против грамотно спланированной атаки ей не устоять. Можно, конечно, шутки ради, запустить эмулятор в эмуляторе (например, BOCHS внутри VM Ware), только это все равно не решит всех проблем, а вот производительность упадет колоссально!

Отельный жесткий диск в этом плане намного надежнее, да и удобнее. Кстати говоря, отключать основной диск необходимо чисто физически — путем отрубания кабеля. Диски, перечисленные в основном разделе BIOS, актуальны только на стадии первичной загрузки, а дальше весь обмен идет через драйвер защищенного режима, работающий напрямую с контроллером. Отключение каналов интегрированного контролера через BIOS Setup, как правило, делает диски невидимыми и штатными средствами Windows до них будет не дотянуться, однако, зловредный код при большом желании со своей стороны, может перенастроить контроллер на ходу, подцепив все каналы. Естественно, это системно-зависимая операция и все контроллеры программируются по-разному, однако, поддержать пару-тройку самых распространенных чипсетов вполне реально!

Короче говоря, "дедовские" способы — самые надежные, но неудобные. Виртуальные машины — удобные, но ненадежные. Вот и выбирай!

Рисунок 19 будущее виртуальных машин