УДК 004.3

Особенности работы в среде с памятью readonly

В. В. Кравец
ФГУП «Всероссийский научно-исследовательский институт проблем вычислительной техники и информатизации», ЗАО «ОКБ САПР», Москва, Россия

Описаны сложности, с которыми можно столкнуться при работе программ и ОС в среде, в которой память доступна только на чтение. Предложены некоторые способы преодоления этих проблем.

Ключевые слова: МАРШ!, доверенная загрузка, доверенный сеанс связи, readonly память.

Когда я был студентом, только изучающим основы защиты информации, я был крайне удивлен тем фактом, что защита информации зачастую строится не на базе криптографии, а на достаточно большом комплексе защитных мероприятий. Особенно важным фактом, который я уяснил, оказался тот, что иногда для защиты информации не требуется использовать сильную криптографию, а достаточно положить носитель в сейф, а рядом поставить человека с автоматом.

Прошло уже много лет, но данное обстоятельство не потеряло своей актуальности. Однако я не хочу рассказывать о том, как выбирать автоматы для защиты информации, но все же расскажу о том, что последняя живет не криптографией единой.

Одним из краеугольных понятий в сфере защиты информации является "доверенная вычислительная среда" (ДВС). Без нее никуда — какой смысл всех защит, если сама система изначально скомпрометирована. Нынешним вирусам не составит труда подменять результаты защитных операций в нужную им сторону — заменить сильную криптографию на шифр Цезаря, проверку контрольной суммы свести к перманентному ответу "Да, все сходится".

Есть множество способов обеспечения ДВС. В продукте "МАРШ!" мы используем следующий — фиксирование эталонного образа ОС таким образом, чтобы исключить влияние пользователя на этот эталон. Фактически каждая новая загрузка восстанавливает эталонное состояние, которое априори проверено и считается безопасным.

В физике есть одно интересное эмпирическое правило — "чем воздействие порождается, тем и обнаруживается" (в качестве примера приводятся магнитные и электрические поля). Применив это правило к защите информации, получаем следующее: "если защита основана на софте, то и обойти ее можно будет софтом". Поэтому для исключения большинства угроз и атак мы фиксируем образ не с помощью "софта", а с помощью "железа" — контроллер памяти, содержащий эталон (загружаемый образ ОС), который находится в специальном режиме "readonly", что означает запрет изменений на аппаратном уровне.

Доверенную среду мы обеспечили, но вот работоспособна ли она? Оказывается, что не очень. Любая операционная система, будь то Windows, либо Linux, во время загрузки проводят достаточно большое количество операций записи — ведут журналы, фиксируют какие-то настройки. Здесь возможны три пути решения:

1. Изменение исходного кода операционной системы таким образом, чтобы операции записи не вызывались.

2. Изменение поведения памяти таким образом, чтобы при попытке записи не возвращалась ошибка, а изменения молча, игнорировались.

3. Использование оперативной памяти как временный RAM-диск, на который разрешены операции записи.

К сожалению, первые два способа не очень состоятельны. Для изменения кода операционной системы нужны довольно-таки глубокие знания и понимание всего того, что происходит в процессе загрузки, кроме того, не понятно, что делать с ОС Windows, исходных кодов которой у нас нет. Игнорирование попыток записи тоже не помогает из-за того факта, что опытным путем выясняется, что операционные системы не только хотят что-то записать, но они потом еще и пытаются считать то, что записали. Обычно ничего хорошего из этого не выходит.

Зато последний вариант — использование оперативной памяти как временный RAM-диск, на который разрешены операции записи — предлагает действительно аккуратный путь работы.

Кравец Василий Васильевич, аспирант, программист.
E-mail: vkravec@okbsapr.ru

Статья поступила в редакцию 14 июня 2014 г.

Чаще всего такой временный диск работает в виде фильтра, работа которого схематично показана на рисунке.

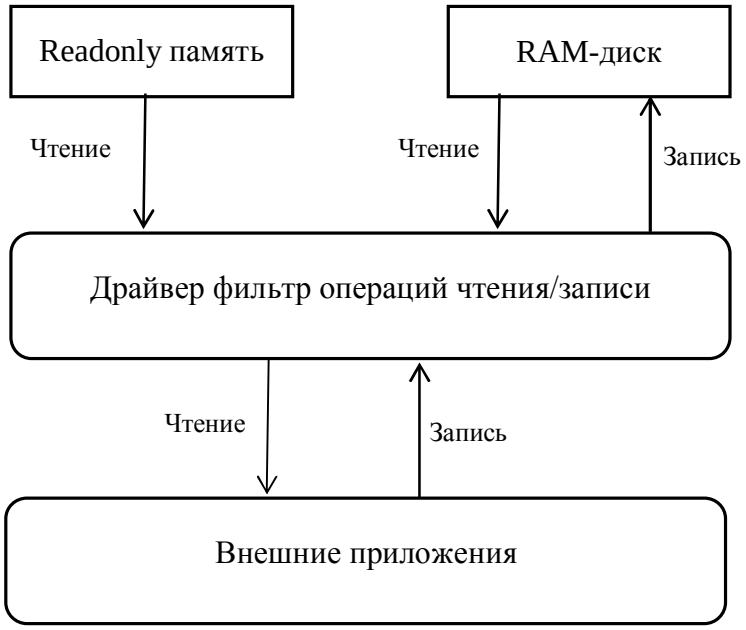

Схема работы с памятью при использовании фильтра записи

Использование такого драйвер-фильтра решается большое количество проблем — ОС загружается, поскольку теперь у нее нет препятствий в виде запрета записи, аналогично работают и все прочие приложения. Но все возможные изменения сохраняются только в оперативную память и на эталонный образ не влияют, а значит, достаточно перезагрузиться и все внесенные изменения пропадут — снова будет загружен эталонный образ ОС.

Но работа пользователя не ограничивается загрузкой операционной системы — ему нужны еще приложения. Прикладное ПО очень часто производит запись той или иной информации:

- ведение журнала работы;
- создание временных файлов для разнообразных операций;
- запись настроек в виде файлов или в реестр;
- операционная система заносит в свои журналы факт запуска ПО;
- запись в атрибуты файла время последнего открытия;

Все эти изменения, так или иначе, будут попадать в RAM-диск, поэтому важно понимать, какие программы какую информацию записывают — RAM диск не бесконечного размера.

Отдельно стоит указать на тот факт, что у большого количества программ первый запуск и все последующие отличаются логически — при первом запуске приложение может создавать настроечные файлы, проверять свои компоненты, выполнять запись в реестр. Особенно неприятно, если после всех этих действий программа откажется работать до перезагрузки компьютера — она же не знает, что после перезагрузки все вернется к исходному эталону. В таких ситуациях программу нужно установить, настроить, несколько раз запустить и перезагрузиться перед фиксированием эталона. Кроме того, возможно включение в эталон portable программ — они обычно уже настроены так, чтобы запускаться сразу в "рабочем режиме".

Еще один вопрос, который остро стоит во время работы с readonly памятью — связь с "внешним миром", т. е. использование внешних данных и сохранение результатов работы. Мы можем предложить самые разные варианты решения:

- Использование внешнего устройства памяти.

В таком случае стоит понимать, что такое внешнее устройство может принести с собой угрозу безопасности, хоть перезагрузка и "вылечит" ОС, но не стоит подвергать такой опасности рабочий сеанс.

- Устройства со специальным интерфейсом для работы с памятью, исключающий перенос произвольной информации.

Вот так замысловато я назвал носители ключевой информации (но не только их). Одной из основных целей работы с устройством "МАРШ!" может являться установление защищенного соединения с некоторым сервером. Для реализации такой задачи в эталонный образ можно включить ПО, обеспечивающее защиту канала (например VPN), а сертификаты и ключи подключать на отчуждаемом устройстве. Хотя если требование отчуждения ключей нет, то сам "МАРШ!" может выступать в роли такого ключевого хранилища.

Не стоит думать, что ключевыми носителями все ограничивается. Можно использовать другой наш продукт — "Секрет". Поскольку "Секрет" привязывается к рабочим местам, мы не только используем безопасный механизм переноса файлов, но и можем быть уверены, что информация будет доступна только на заведомо определенных компьютерах.

- Использование открытого раздела в самом "МАРШе! ".

Такой способ вызывает существенные трудности в плане обеспечения безопасности, но, тем не менее, все равно возможен. Для компенсации возникающих при этом уязвимостей необходимы правильные организационные меры.

В своей статье я постарался выделить основные особенности и ограничения, а также методы их героического принятия и преодоления (соотнести объекты и действия предлагается читателю) при работе со средой readonly. Это далеко не все, а только верхушка айсберга, но даже сейчас видно, что такой метод создания временной ДВС (чтобы получить доверенный сеанс связи в дальнейшем) вполне жизнеспособен и может активно применяться.

Peculiar properties of operating on the system with readonly memory

V. V. Kravec
All-Russia Scientific Research Institute of Computer Technology and Informatization Problems (VNIIPVTI), OKB SAPR JSC, Moscow, Russia

The article describes the problems might occur in the developing programs and operating systems for the environment which has read-only memory.

Keywords: MARSH!, trusted startup, trusted communication session, readonly memory.

Received June 14, 2014