Статья Компьютер заBIOSает? Изучаем буткиты

xss.is/threads/65355

Самый популярный совет в случае, если что-то идет не так или работает медленно, — перезагрузить компьютер. Вторая по популярности рекомендация — переустановить ОС, чтобы избавиться от вирусов. Если не помогает ни первое, ни второе, то вам «повезло»: ваш компьютер инфицирован вредоносным ПО опаснейшего класса, которое называется буткит.

Мы начали погружение в тему буткитов и рассмотрели обширную группу так называемых legacy-буткитов (подробнее об этом рассказали на вебинаре от команды PT Expert Security Center «Компьютер заBIOSает? Изучаем буткиты»). Ответили на вопрос, почему вредоносы, нацеленные на системы на базе Legacy BIOS, все еще актуальны, поговорили о наиболее известных представителях буткитов, а также рассказали, как они работают, какие преследуют цели и применяют техники.

Буткит — вредоносная программа, которая выполняет свои действия на стадии предзагрузки ОС. Устанавливается либо на первых секторах диска, либо в прошивке. За счет своей работы до старта операционки буткит имеет возможность обойти ряд защитных механизмов.

Цели буткита:

- незаметно установить основную нагрузку (как правило руткит);

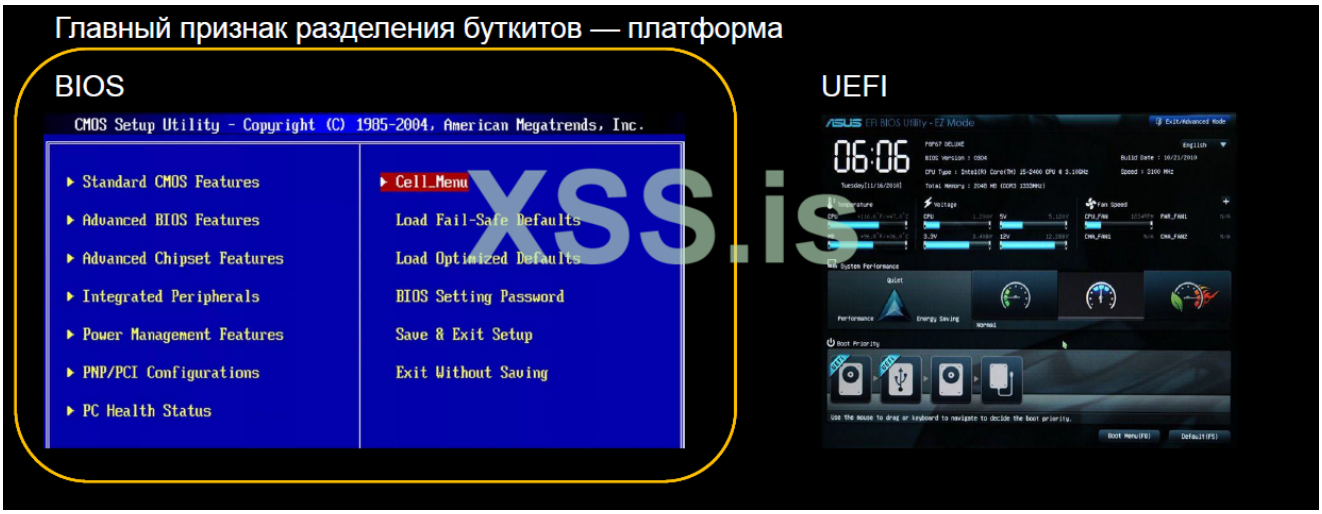
- устойчиво закрепиться на компьютере жертвы (для хакера это более надежный и долговременный способ нежели традиционные зловреды, использующие автозапуски, планировщики задач и др.);
- препятствовать загрузке системы (например, в случае атаки с помощью шифровальщика).

Несколько лет назад появились такие методы защиты от руткитов, как проверка подписи драйверов (Driver Signature) и технология запуска своего легитимного средства защиты раньше всех (Enforcement Early Launch Antimalware), чтобы опередить руткит.

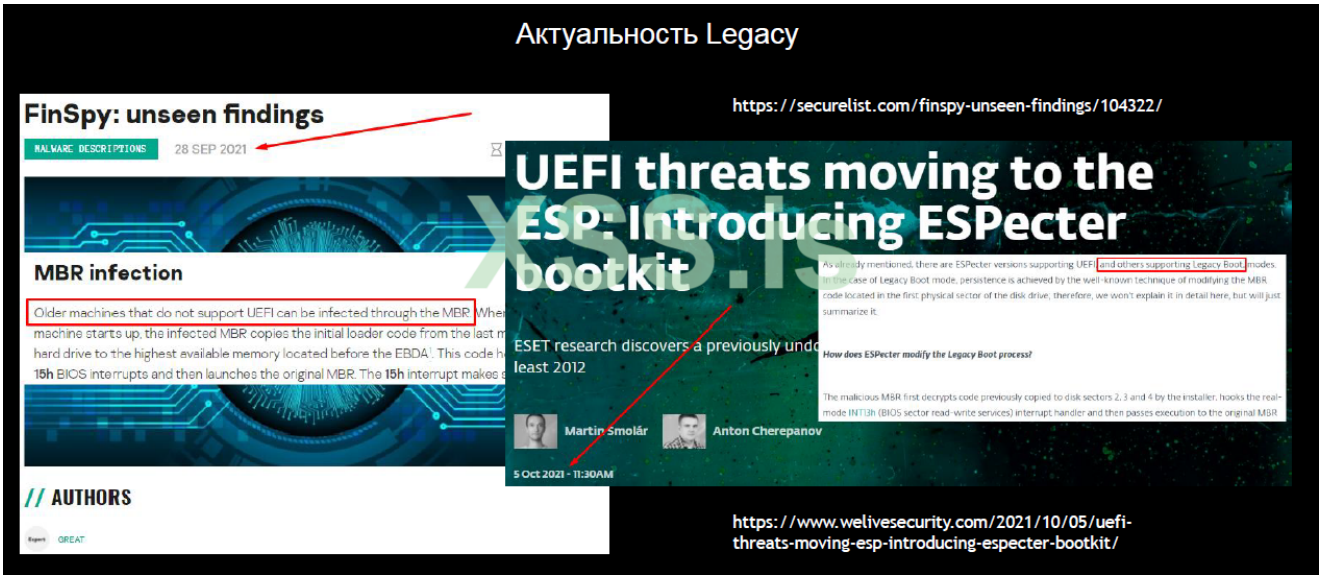
Если у злоумышленника не будет способа зайти в систему глубже, то у него не получится установить руткит, и атака провалится. В таком случае выходом для атакующего становится буткит.

Ниже красным цветом выделены основные этапы, по которым злоумышленники устанавливают буткиты. На третьем этапе, в режиме пользователя, в системе появляются привычные зловреды, такие как средства удаленного доступа и бэкдоры.

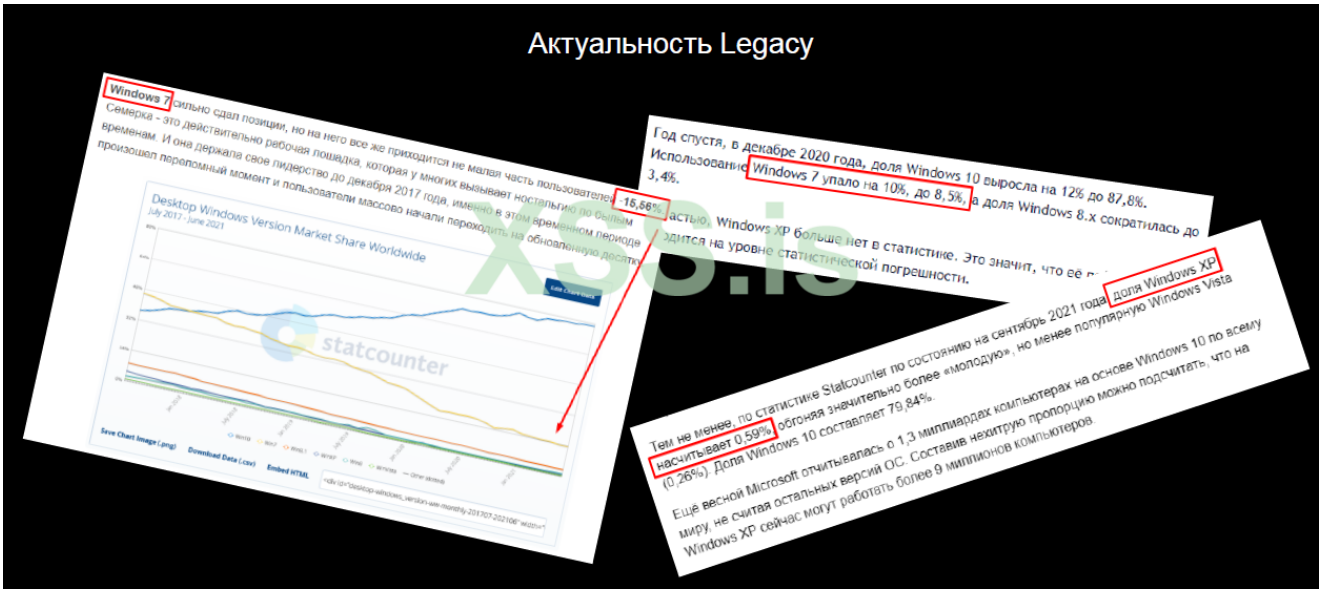
Буткиты бывают двух типов. Одни заточены на так называемые старые Legacy-способы оперировать стадией предзагрузки. Второй тип использует новые подходы, например, основанные на подсистеме UEFI. В статье мы рассматриваем только Legacy-режим



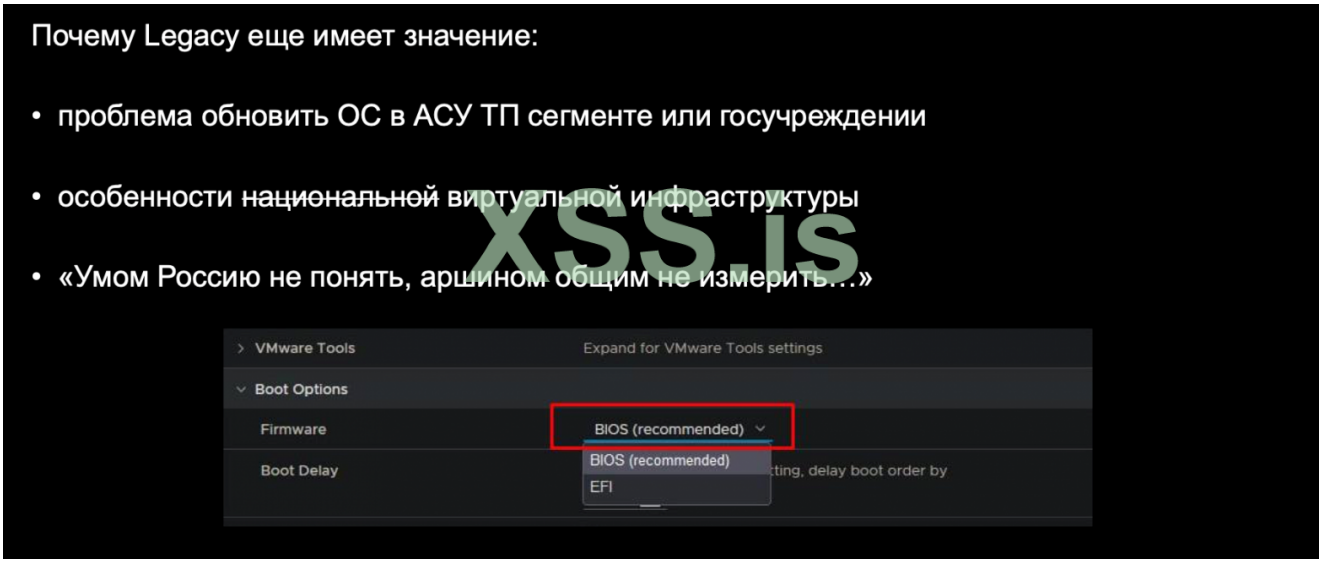
Если посмотреть на исследования коллег, то зловреды, такие как FinSpy, все еще используют в своем арсенале заражение в режиме Legacy наряду с современным UEFI.



Обратите внимание на распространенность операционных систем, таких как Windows 7 (около 16%) и Windows XP (0,59%). Даже полпроцента Windows XP — это более 9 млн компьютеров, многие из которых, по нашему опыту, могут относиться к весьма чувствительным сферам (SCADA, банки и пр.).

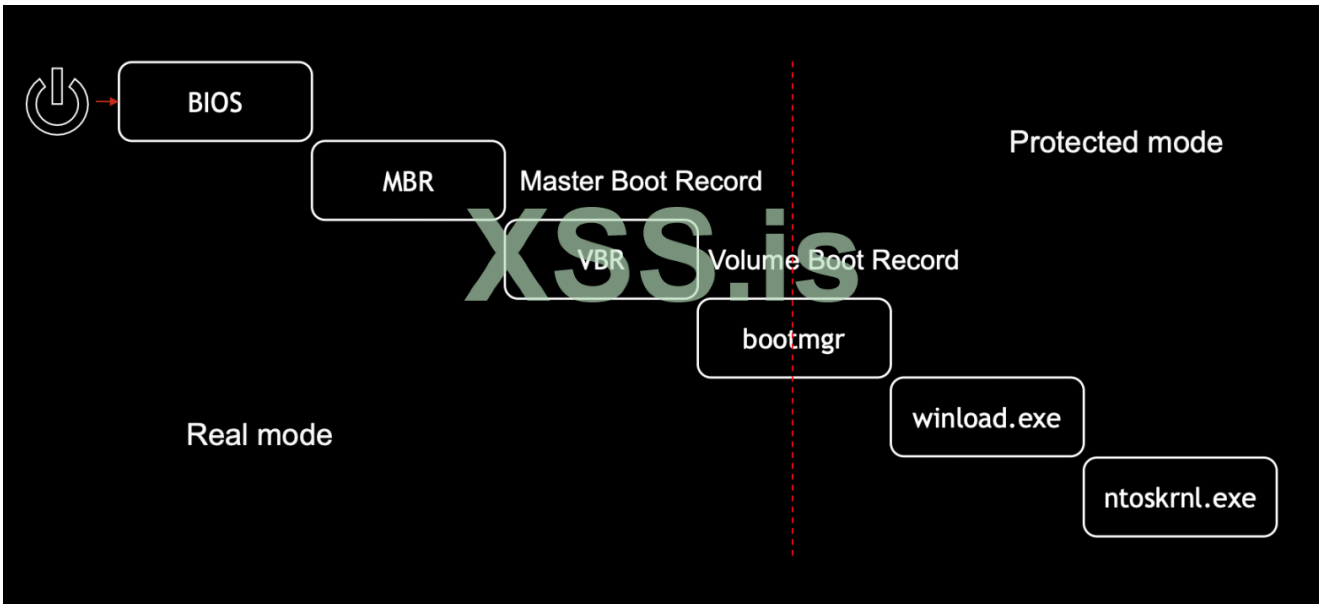


Отметим и виртуальные машины — когда мы устанавливаем такую ОС, сам же vCenter рекомендует Legacy-BIOS, это поле выбрано по умолчанию. А, так как на виртуальной инфраструктуре часто делают критически важные узлы на предприятии, такие как контроллеры домена, то Legacy-BIOS хакеру весьма интересен



Процесс загрузки Windows в режиме Legacy

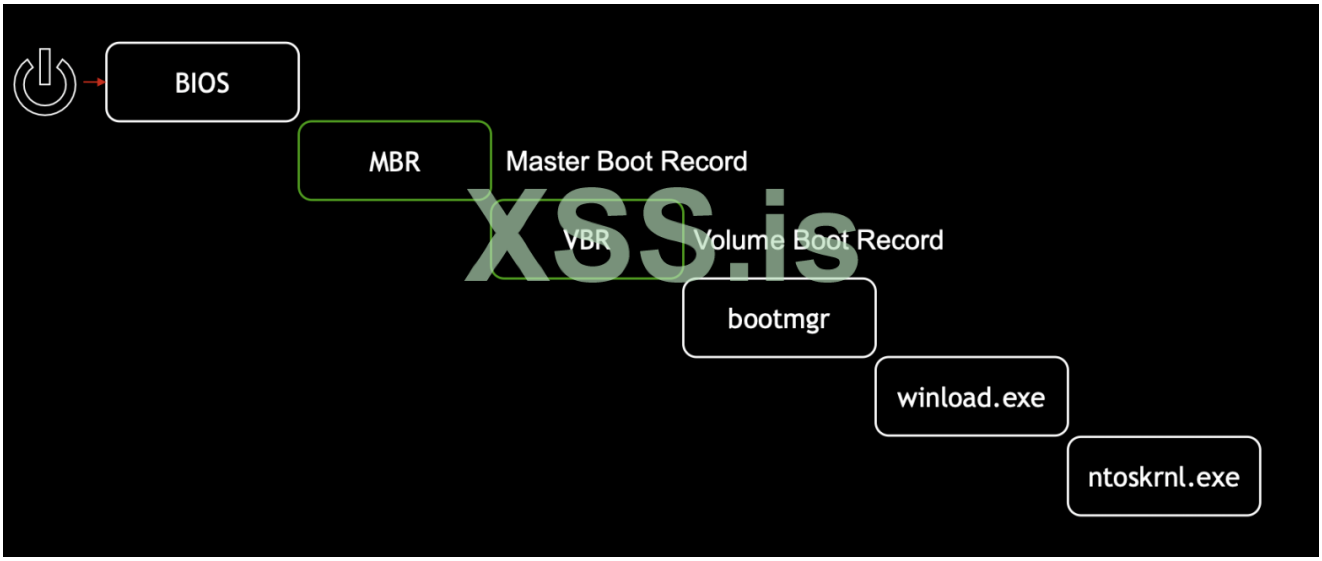
В целом загрузку можно представить как работу цепочки программ, которые стартуют начиная с нажатия клавиши питания ПК. Первым появляется BIOS, который работает на отдельном чипе. Его задача — инициализировать оборудование, определить загрузочный диск, сформировать программный интерфейс для взаимодействия с оборудованием и т.д



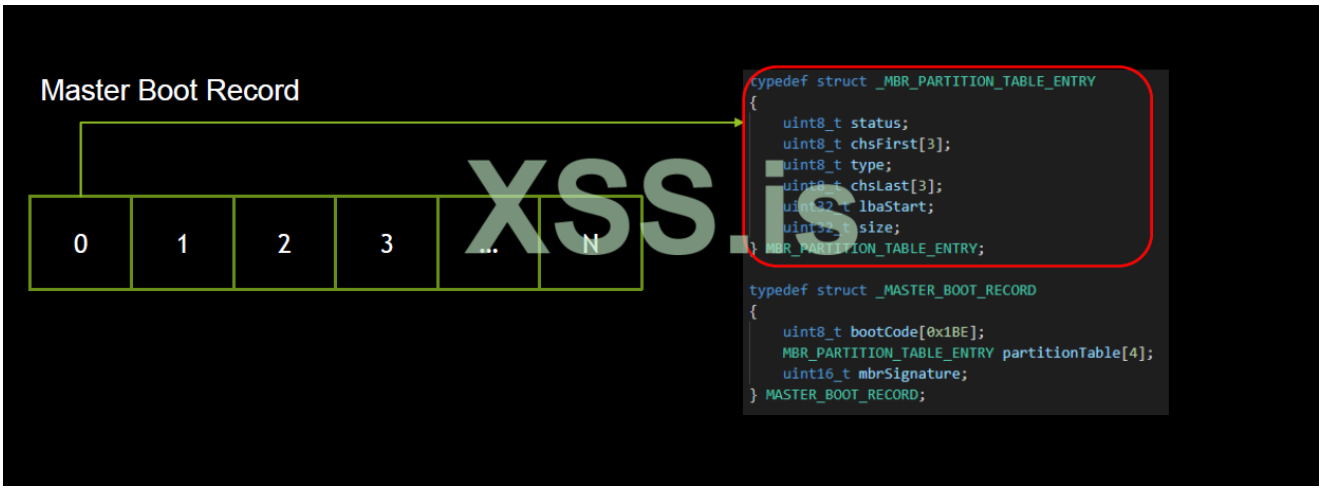
Сразу после загрузки процессор работает в реальном режиме, который остался от старых поколений CPU. Ключевой особенностью такого режима является сегментная

адресация. В этом режиме каждый адрес памяти идентифицируется двумя значениями — это сегмент и смещение. Чтобы перевести в линейный адрес, надо использовать формулу ниже:

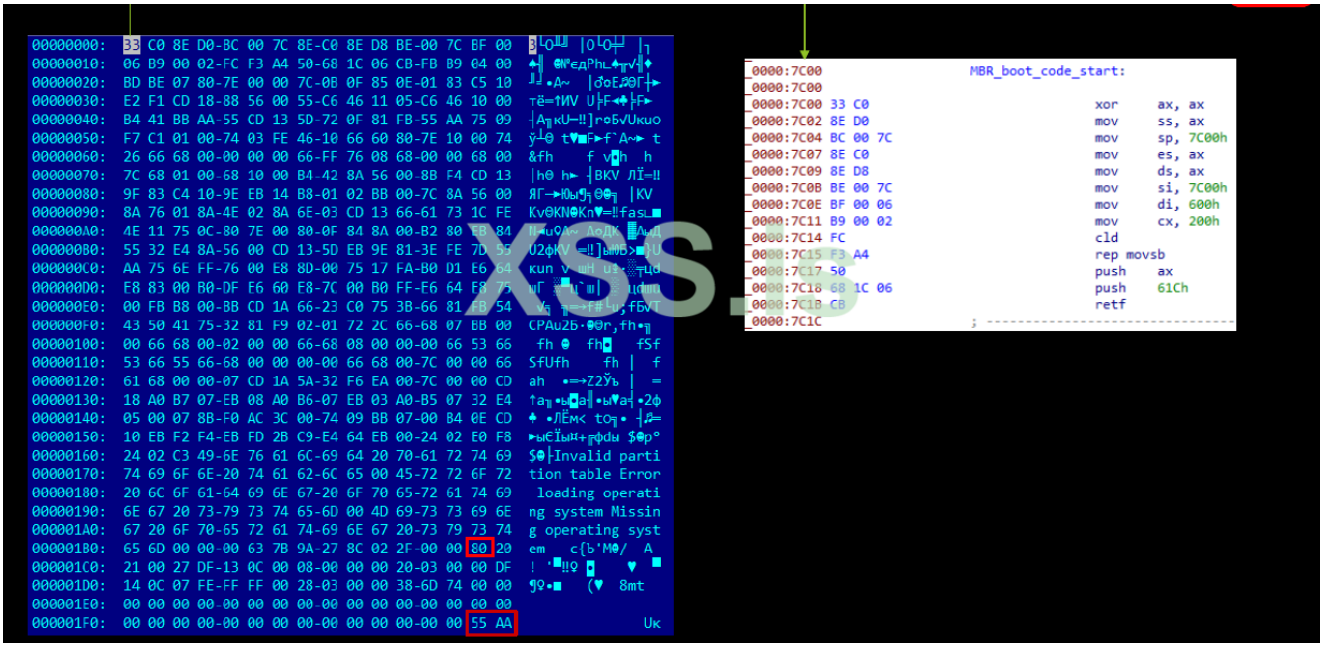
На данном этапе используется физическая память, виртуальная еще не задействована, процессор имеет доступ к физической памяти без ограничений. BIOS после первоначальных операций определяет загрузочный диск и считывает его нулевой сектор в память, на которой располагается MBR (главная загрузочная запись). Получив управление от BIOS, MBR находит VBR (загрузочная запись раздела) и передает туда управление. Оба эти компонента находятся в загрузочных секторах диска, а не внутри файловой системы.



MBR занимает весь нулевой сектор, содержит загрузочный код, таблицу разделов и сигнатуру, которая позволяет проверить валидность MBR

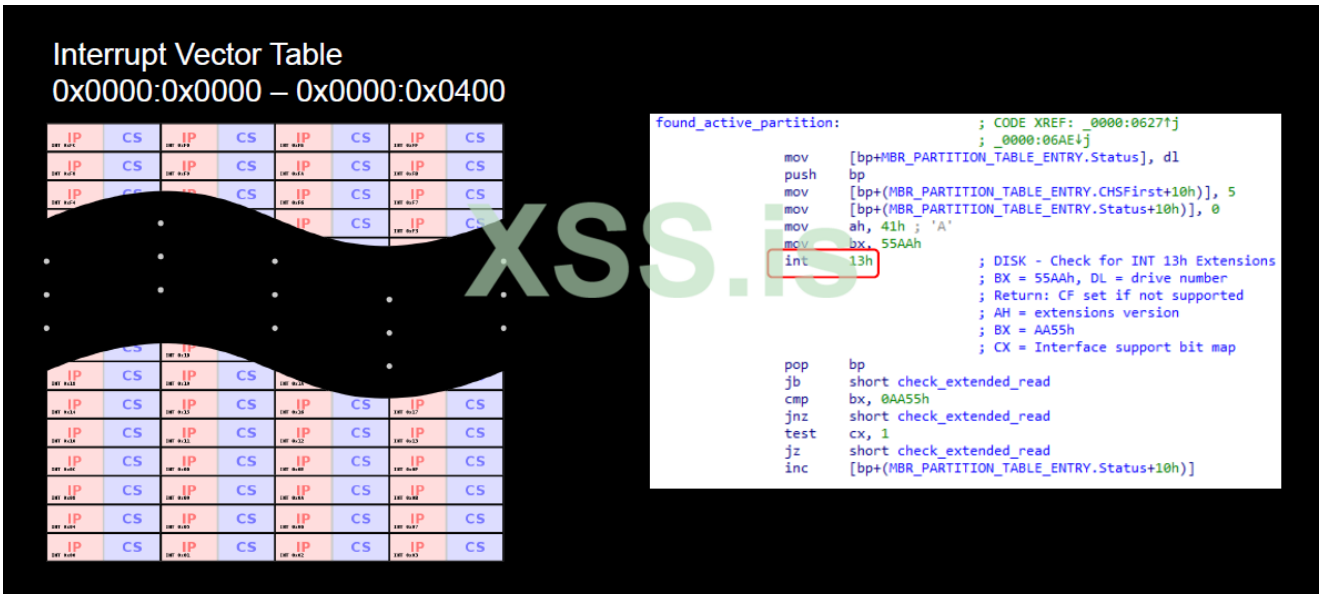


Если посмотрим на реальную MBR (ниже), то в конце увидим сигнатуру 55 AA и выделенное значение 80 в поле статус, которое отмечает активный раздел на диске. Справа мы видим загрузочный код, который в начале своей работы перемещает себя в другое место.

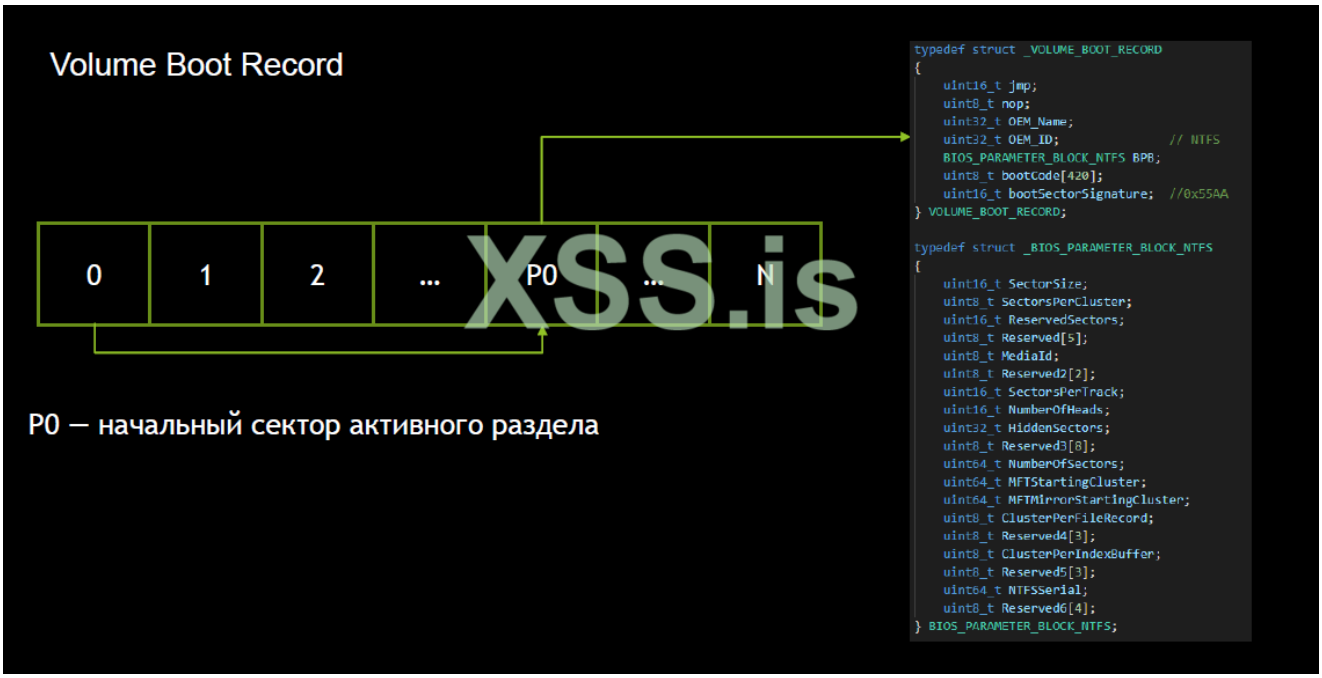


Ниже — программный интерфейс, который предназначен для взаимодействия MBR с диском и другим оборудованием и представлен BIOSом в виде таблицы векторов прерываний. Эта таблица представляет собой набор ячеек по 4 байта, каждая из которых содержит сегмент и смещение обработчика прерывания. В частности, под

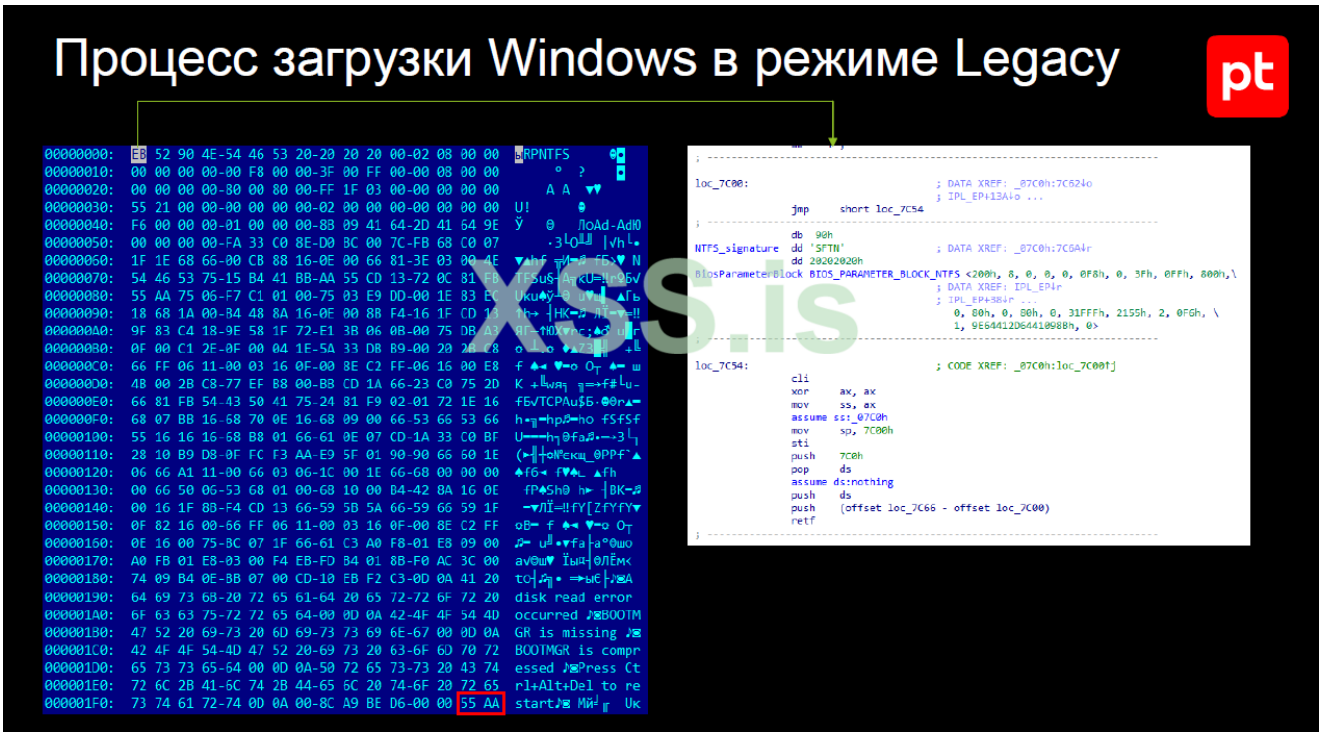
номером 0x13 располагается обработчик прерывания для работы с диском; это не одна функция, а набор сервисов для определения параметров диска, чтения его в различных режимах. Номер конкретного сервиса задается в регистре AH.



Задача MBR состоит в том, чтобы с использованием прерывания 0x13 считать нулевой сектор активного раздела. Его адрес находится в таблице разделов MBR. Этот нулевой сектор содержит главную загрузочную запись раздела (VBR), внутри которой содержатся параметры раздела и фрагмент загрузочного кода. Помимо параметров раздела там есть указатели на структуру, описывающие файловую систему (в данном случае — NTFS), для того чтобы найти таблицы, прочитать их и распарсить файловую систему активного раздела.

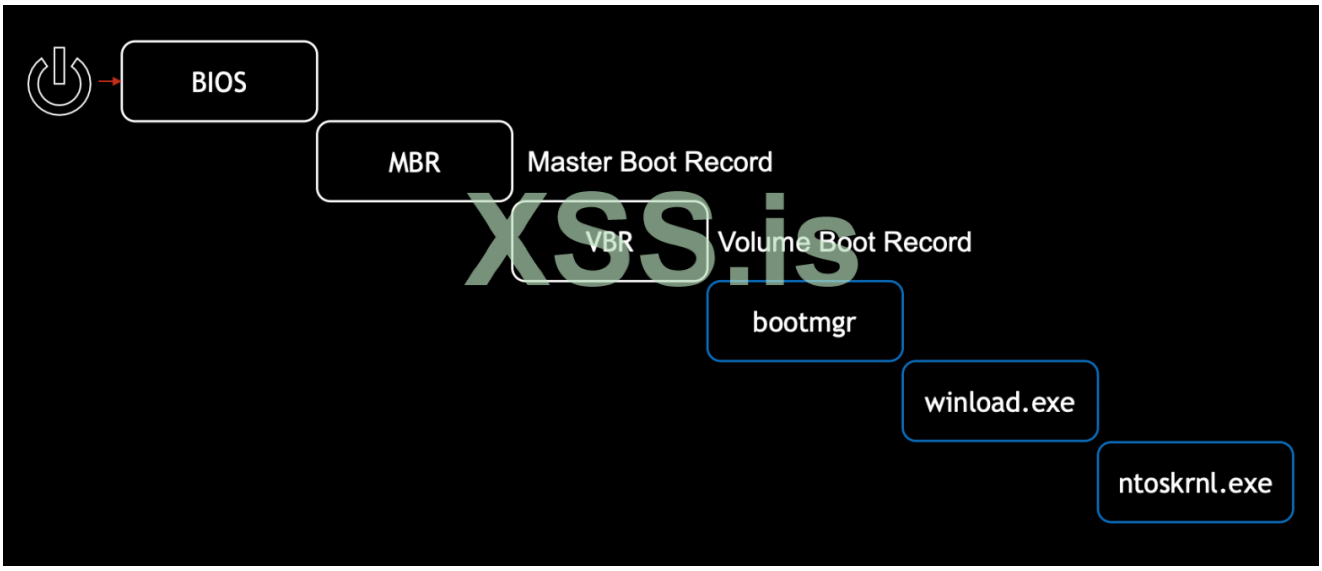


Далее мы видим дамп VBR, он содержит сигнатуру 55 AA для проверки валидности, а справа — листинг загрузочного кода раздела.

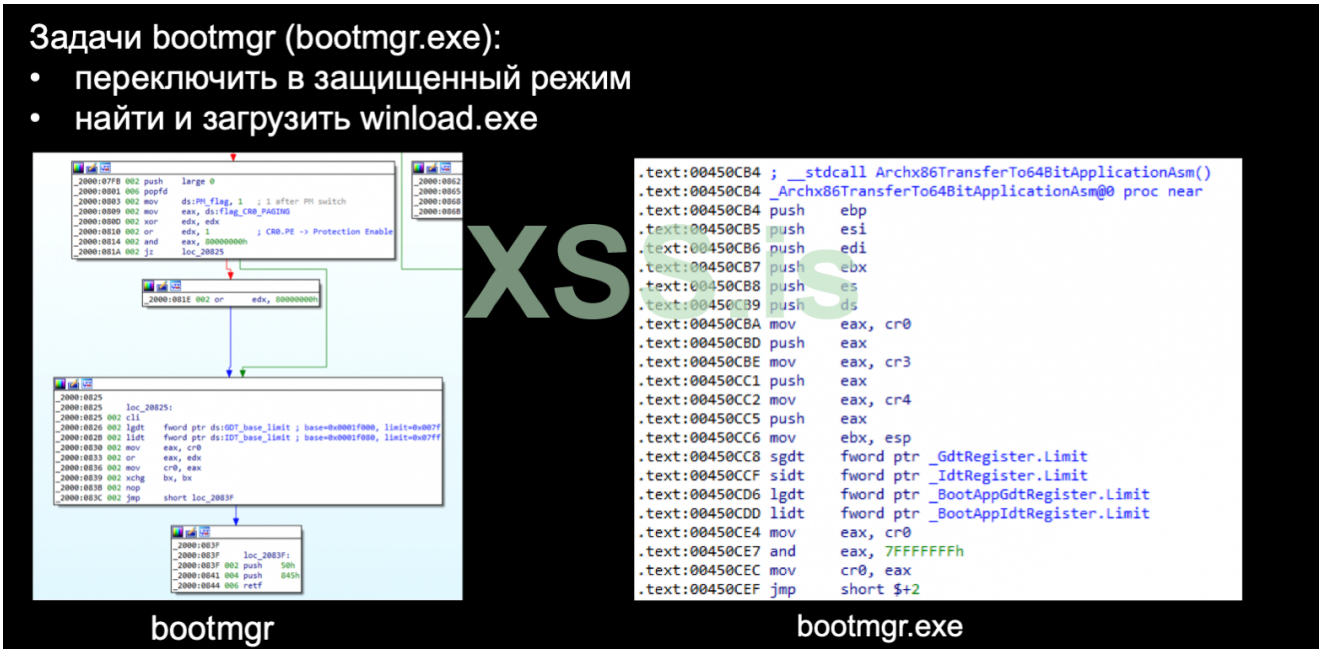


После того как VBR получила управление, ее главной задачей является загрузка кода Initial Program Loader. Этот код может занимать до 15 секторов и располагается сразу же после VBR. Его размер позволяет распарсить файловую систему и найти дальнейшие компоненты, участвующие в загрузке системы

Дальнейшая работа ложится на модули, которые расположены внутри файловой системы.

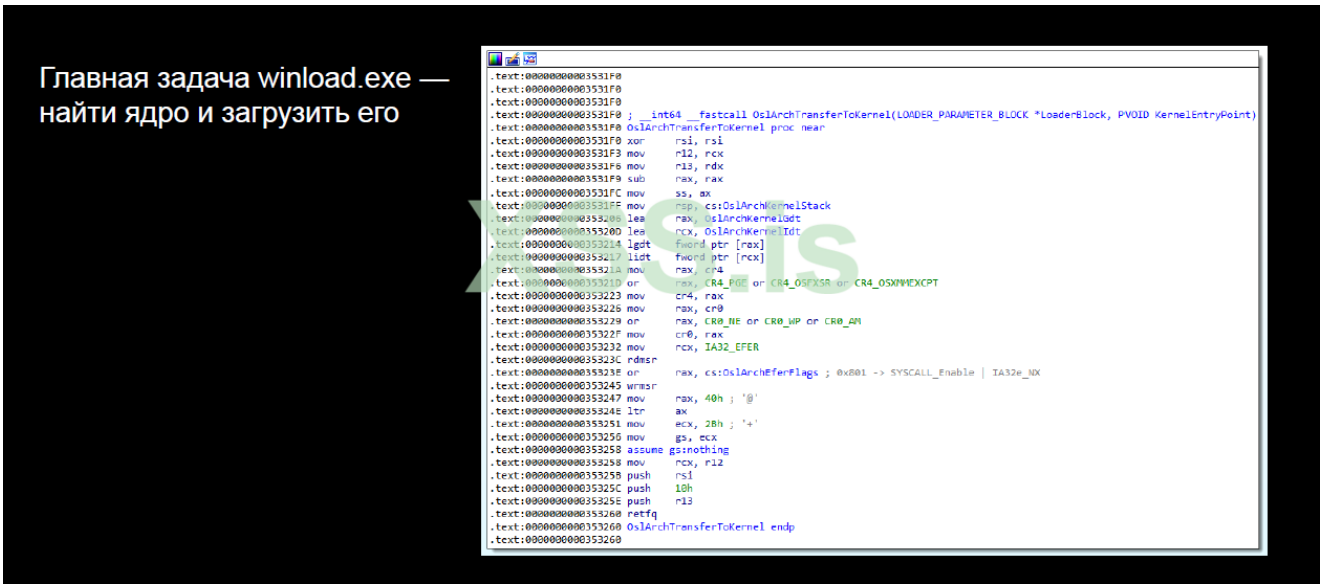


Главной целью менеджера загрузки bootmgr является переключение процессора в защищенный режим. Этот менеджер работает на стыке двух режимов — реального и защищенного.



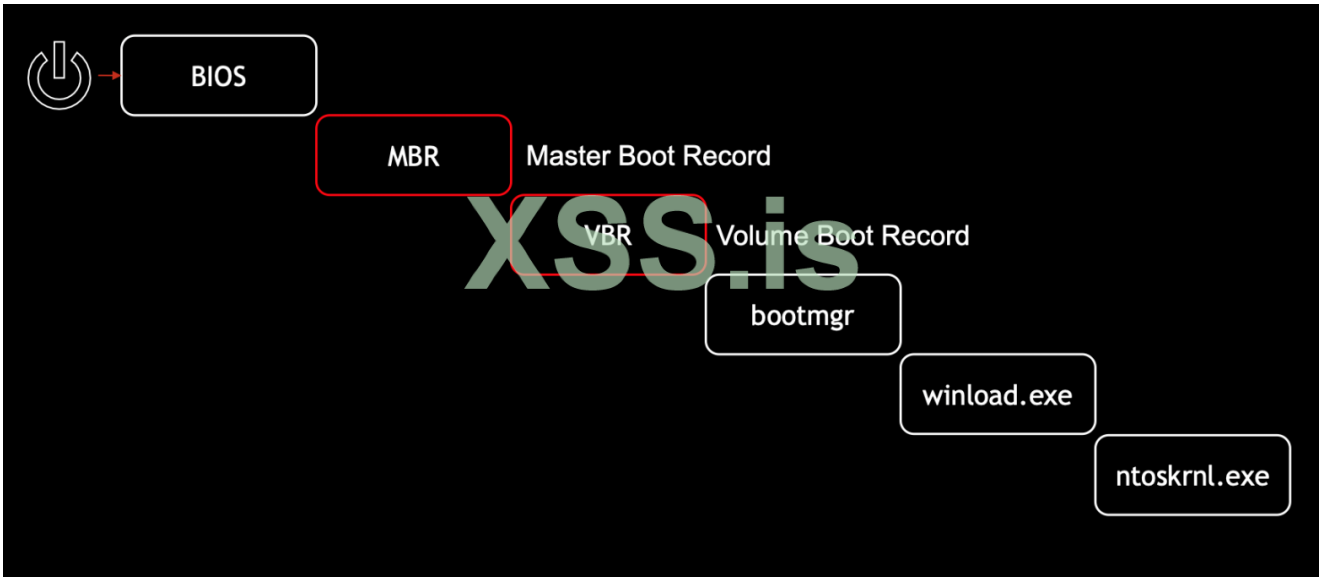
Далее переходим к Winload.exe, задача которого (если глобально) — найти ядро и передать ему управление. После этого вступает в работу непосредственно операционка

— запускает процессы, службы, драйверы и готовится к взаимодействию с пользователем.



Как работает буткит

Буткиту надо вступить в работу как можно раньше, пока система находится в наиболее уязвимом состоянии. Логичным этапом тут является заражение BIOS, что весьма непросто — злоумышленнику надо найти уязвимость или способ, который позволит это сделать. Могут возникнуть и сложности с тем, чтобы переписать и не «окирпичить» компьютер в целом. Чтобы упростить задачу, атакующий смещается в сторону MBR и VBR. В этом случае для заражения не требуется специфических знаний, доступа к железу и т.д. Злоумышленник, обладающий базовым знанием WinAPI, может переписать загрузочный код в нулевом секторе диска или раздела, поменять на вредоносный или же просто на мусор, чтобы препятствовать загрузке. Что касается следующего этапа, то для заражения bootmgr злоумышленникам требуется повысить привилегии в системе и преодолеть другие проблемы, чтобы не поломать при этом загрузку.

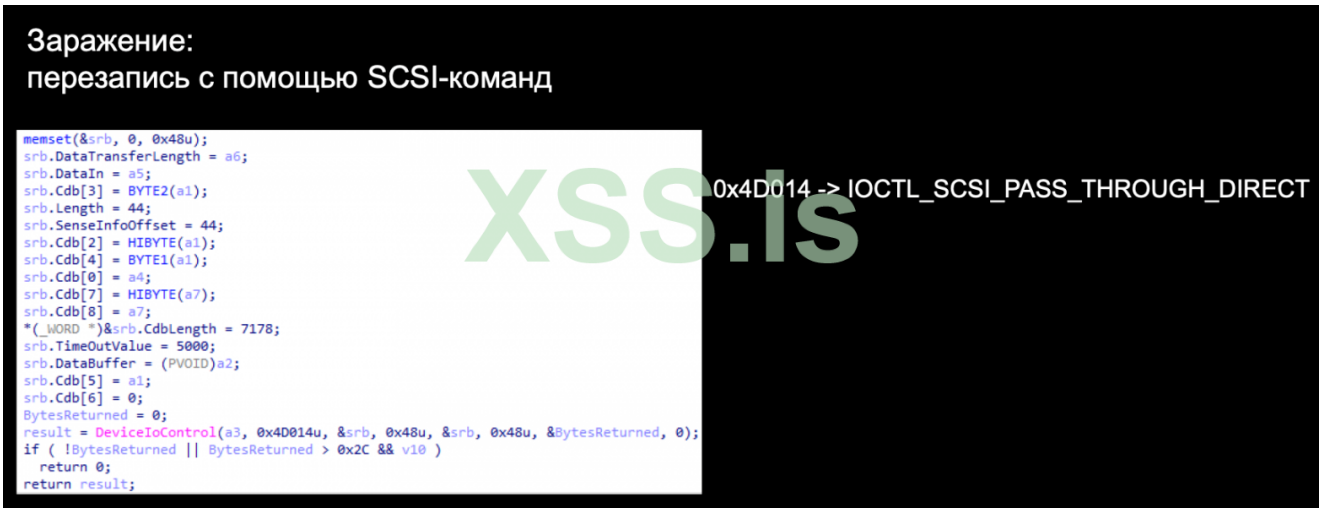


Важно отметить, что инфекторы могут сохранять оригинальный загрузочный код в скрытой области диска. Это может быть сделано для того, чтобы после выполнения вредоносных действий использовать этот оригинальный код для продолжения загрузки системы. На примере ниже инфектор заражает MBR, а для сохранения дополнительных компонентов и прочего создает скрытый раздел в неразмеченной области. Такой раздел не будет отражен ни в таблицах MBR, ни в таблицах NTFS.

Кроме того, злоумышленник может поменять значение адреса реальной VBR на адрес вредоносного кода, который разместил в этой скрытой области, чтобы VBR передавала управление вредоносному коду, не трогая VBR.

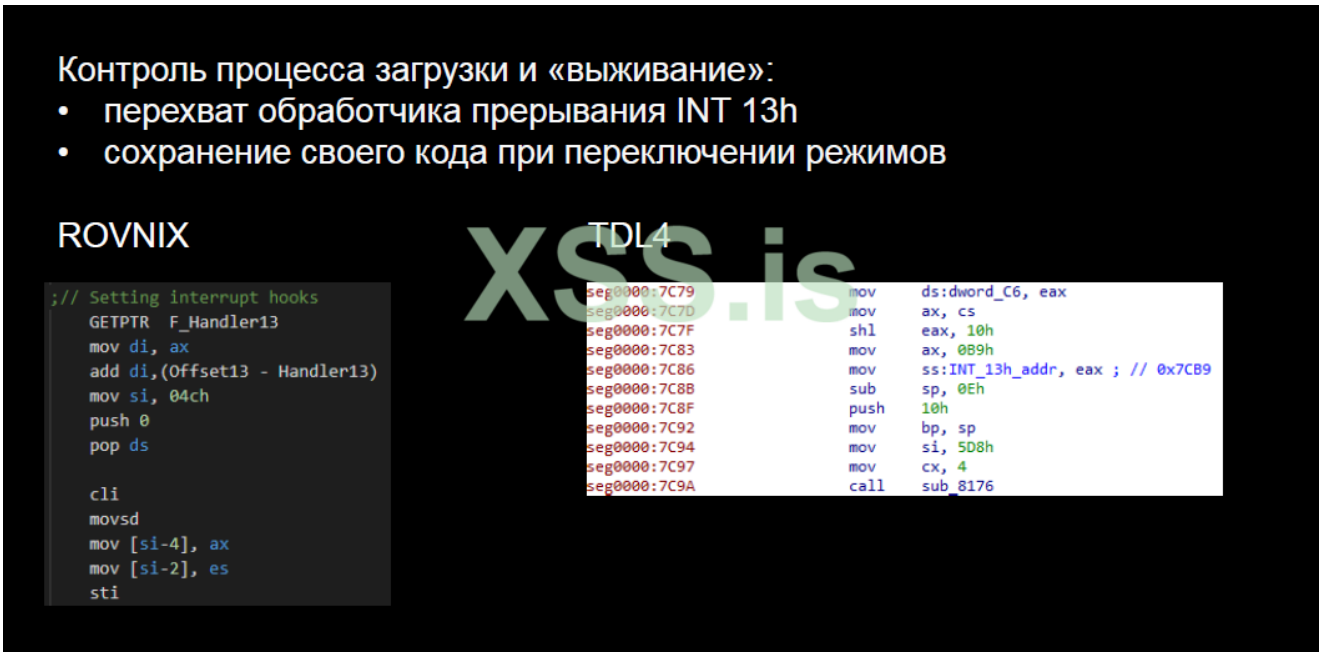
Чтобы записать данные в диск напрямую, используется простейший набор функций CreateFile и WriteFile.

Есть и другой способ записи на диск — с использованием SCSI-команд и функции DeviceIoControl. Это способ позволяет отправить команды напрямую минипорт-драйверу диска, минуя стандартные фильтры. Это может повысить скрытность работы инфектора. После заражения MBR/VBR и перезагрузки системы BIOS считает вредоносную MBR/VBR и запустит.



Буткиту необходимо отслеживать тот этап загрузки, который происходит в данный момент. Главным механизмом является использование перехвата обработчика какого-либо прерывания и замены его на свою вредоносную функцию, сохраняя адрес оригинальной функции, чтобы не поломать загрузку. Самым важным является 0x13-е прерывание — перехватывая его, буткит может проверять, что считывается с диска в

данный момент и позволяет подменить данные, считываемые с диска, например, набор параметров, которые управляют загрузкой системы, включая цифровую проверку подписи драйверов.



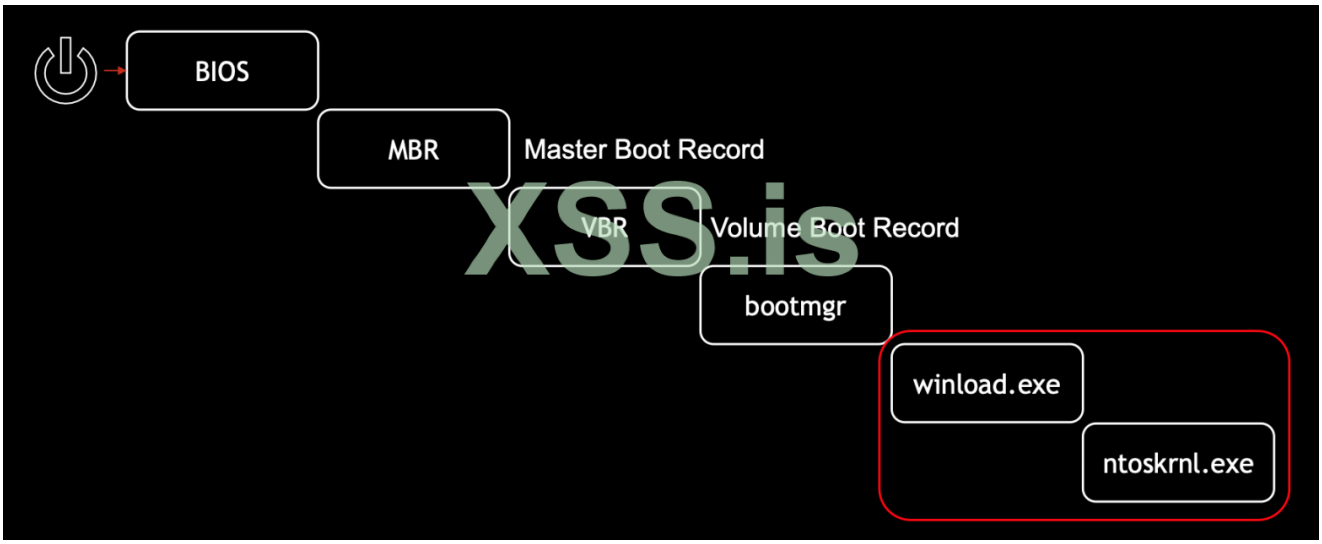
Чтобы сохранить данные при переключении в защищенный режим работы процессора, некоторые буткиты, такие как Rovnix, применяют копирование части своего вредоносного кода в неиспользованную область таблицы IDT.

Кроме того, чтобы передать управление на код, спрятанный в таблице, этот буткит использует механизм отладки с помощью специальных регистров, ставя точку

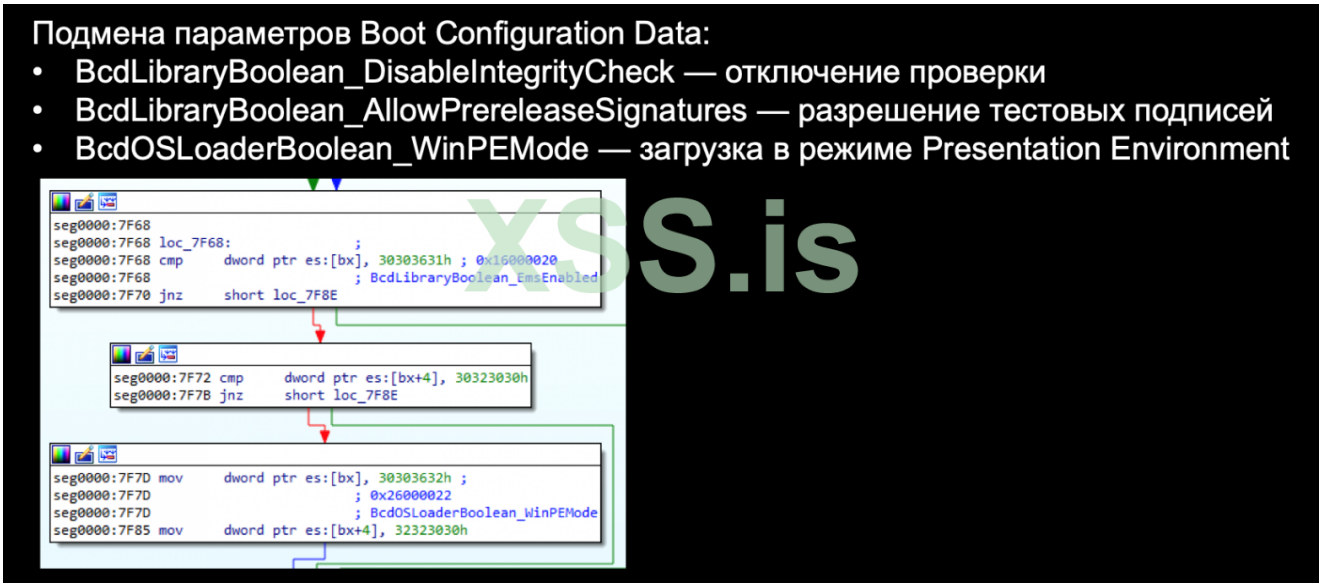
останова на адрес точки входа Winload.exe. И как только управление передается на Winload.exe, сработает ранее перехваченный обработчик первого — дебажного — прерывания и выполняется вредоносный код.

```
;// NT6 uses code segment selector 20h
    mov cx, 20h
@@:
    mov [bx+2], cx      ;// selector
    mov [bx+6], ax      ;// segment hi-bits

;// Setting execution break on OSLoader entry (NT6 only)
;// Ntldr always locates OSLoader.exe at 400000h, the entry point of OSLoader is always 401000h.
    mov eax, 401000h
    mov dr2, eax

;// Enabling read/write breakpoint for R0 and execute for R2
    mov eax, 00030622h
    mov dr7, eax
    ret
HookIdt endp
HookIdt_end::
```

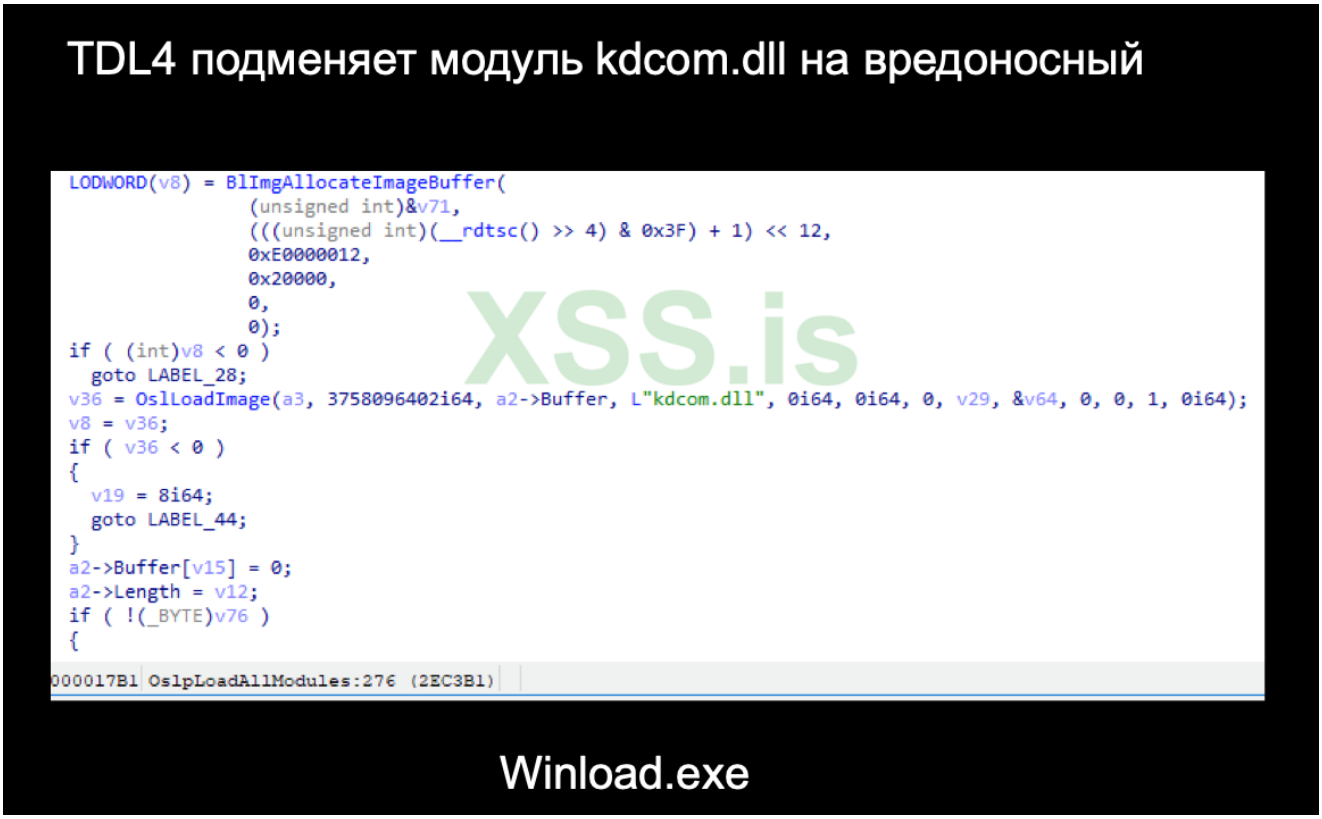
Когда этап с загрузочными секторами пройден, буткит нацеливается на работу с загрузчиком системы и ядром.



В частности, имея контроль над процессом чтения данных перехваченного обработчика прерываний, буткит может подменять значения загрузочных параметров BCD. Например, буткит может вынудить загрузиться систему в уязвимом режиме.

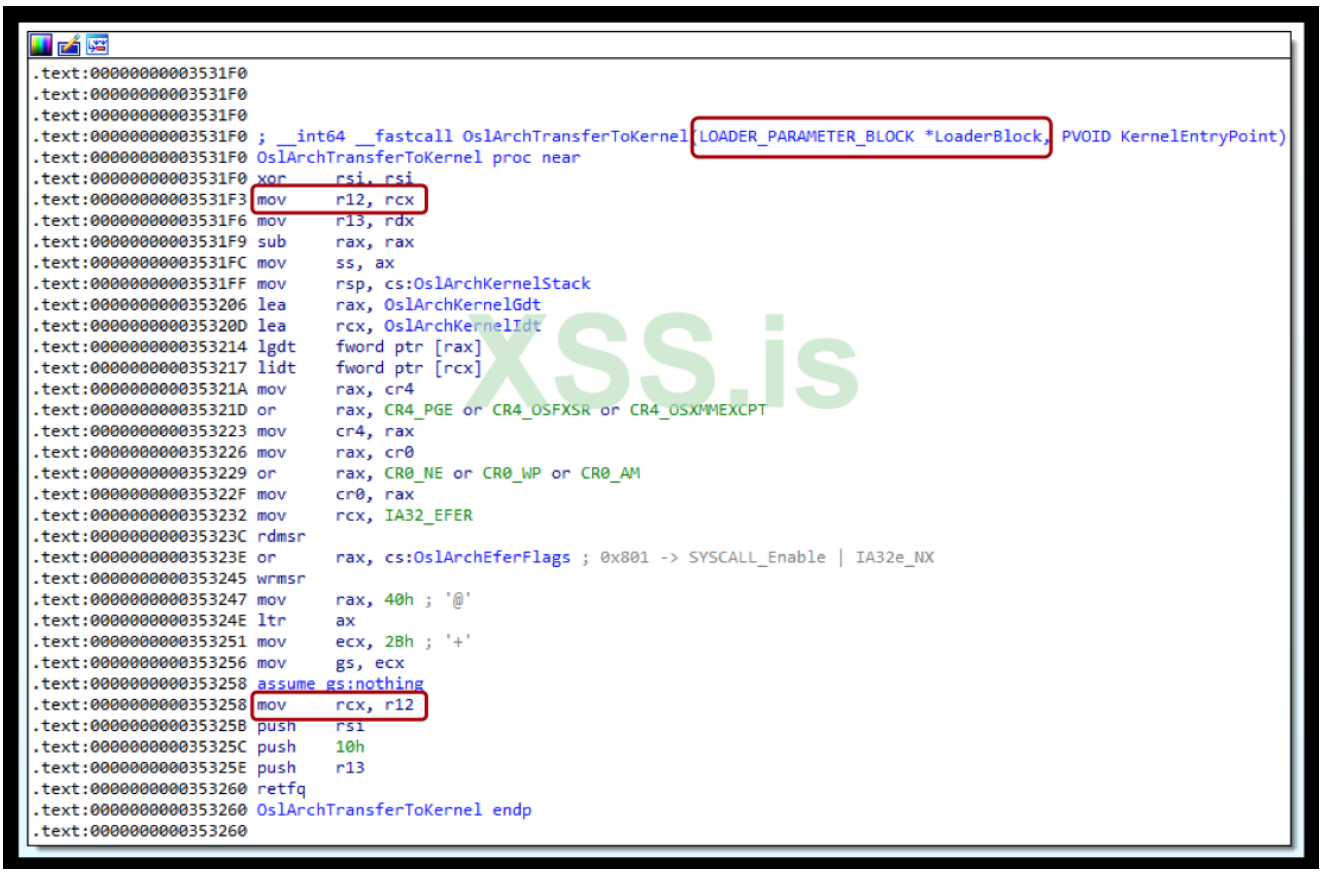


В частности, вредонос TDL4 проверяет, какой параметр считывается: если это EmsEnabled, то буткит подменяет его на параметр WinPEMode и загружает систему в режиме Pre-Installation Environment, в котором проверка подписей отключена.



Rovnix использует более интересный способ. Этот буткит модифицирует список загрузочных драйверов, вставляет свой драйвер, и ядро, когда оно будет проходить по этому списку в процессе инициализации, вызовет точку входа вредоносного драйвера.

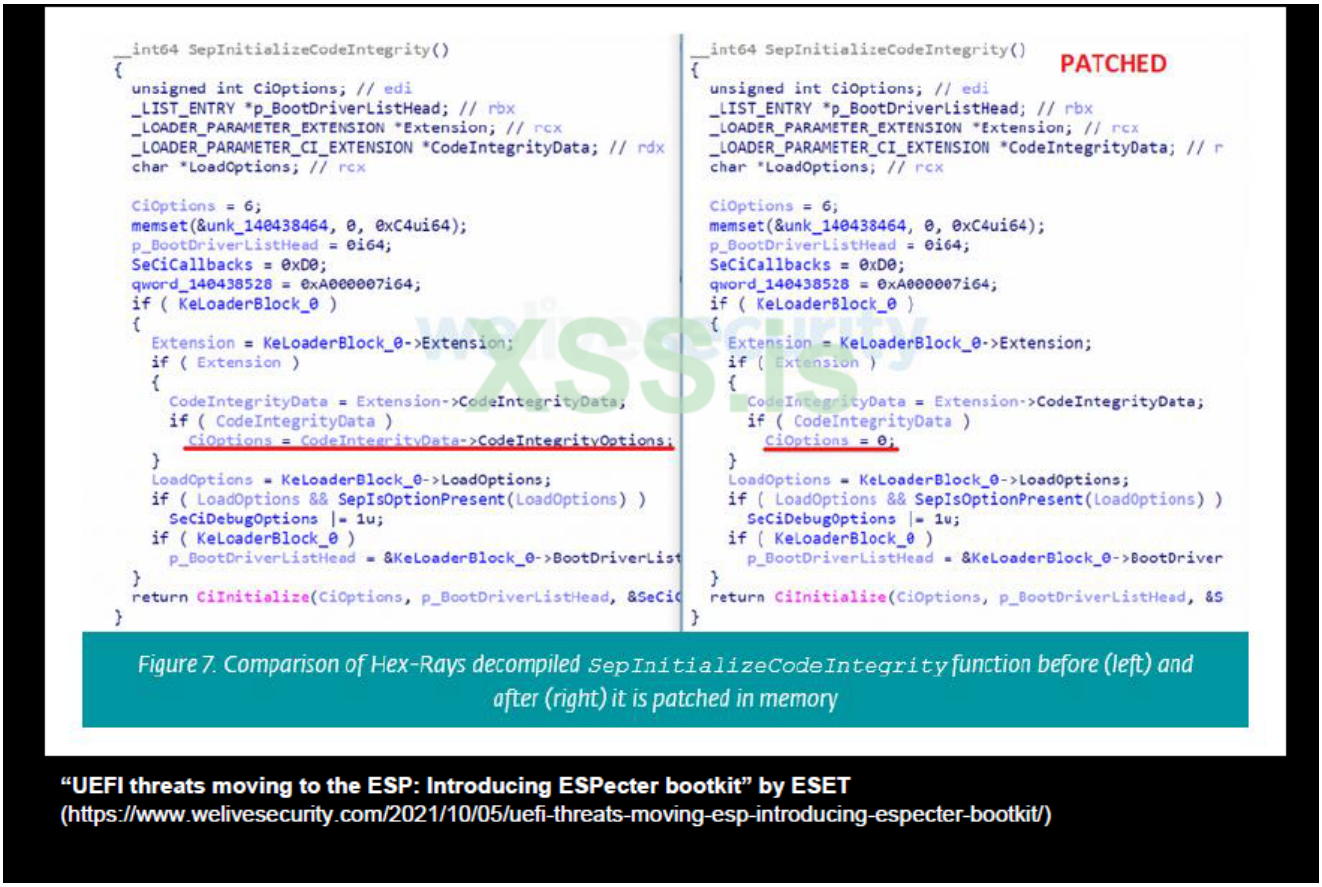
Ниже мы видим, что в первом параметре функции OslArchTransferToKernel содержится указатель на структуру LoaderBlock, и он же передается в качестве единственного параметра точки входа ядра.



Если говорить о более свежих зловредах, то буткит ESPecter, описанный коллегами из ESETниже, перехватывает функцию OslArchTransferToKernel и патчит проверку

цифровых подписей в ядре, оставляя систему в уязвимом состоянии.

Заметим, что этими приемами злоумышленники не ограничиваются: им ничего не мешает, имея возможность влиять на загрузку, выдумать любой механизм, который позволит достичь своей цели.



В заключении материала повторим основные угрозы:

- ядро ОС не всегда может быть уверено в себе;
- код, который выполняется в процессе загрузки, в тех же зараженных MBR и VBR, может получить доступ к любым данным, расположенным на диске без каких-либо ограничений;
- несмотря на то, что Legacy-BIOS уходит в небытие, до сих пор миллионы машин работают на Windows 7 и XP, при этом зачастую они функционируют на критически важных узлах.

На этом все.

В этом материале мы постарались рассказать об основных угрозах, связанных с буткитами, нацеленными на Legacy-BIOS, обрисовать наиболее популярные техники атак и объяснить, почему такие зловреды по-прежнему используются.

Авторы:

- *Антон Белоусов, старший специалист отдела обнаружения вредоносного ПО экспертного центра безопасности Positive Technologies (PT Expert Security Center)*
- *Алексей Вишняков, руководитель отдела обнаружения вредоносного ПО экспертного центра безопасности Positive Technologies (PT Expert Security Center)*

Автор статьи *ptsecurity*