УДК 004.056.53
DOI 10.21685/2587-7704-2017-2-2-4

Open Access | RESEARCH ARTICLE

Подходы к реализации модуля доверенной загрузки для вычислительной платформы с технологией UEFI

А. П. Кашубина
Пензенский государственный университет, Россия, 440026 г. Пенза, ул. Красная, 40

А. П. Иванов
Пензенский государственный университет, Россия, 440026 г. Пенза, ул. Красная, 40

Аннотация. Рассматривается обеспечение доверенной загрузки в вычислительных системах с расширяемым интерфейсом встроенного программного обеспечения (UEFI). Отмечаются недостатки используемых в настоящее время подходов к реализации модулей доверенной загрузки при их эксплуатации в вычислительных системах с UEFI. Предлагаются подходы к реализации модуля доверенной загрузки для вычислительной платформы, поддерживающей UEFI.

Ключевые слова: информационная безопасность, защита от несанкционированного доступа, доверенная загрузка, встроенное программное обеспечение, BIOS, UEFI, модуль доверенной загрузки.

Approaches to implementation of the Trusted Boot Module for a computing platform with UEFI technology

A. P. Kashubina
Penza State University, 40 Krasnaya Street, 440026, Penza, Russia

A. P. Ivanov
Penza State University, 40 Krasnaya Street, 440026, Penza, Russia

Abstract. Provision of a trusted boot in computing systems with Unified Extensible Firmware Interface (UEFI) is discussed. Shortcomings of the used approaches to the implementation of trusted boot modules during their operation in computer systems with UEFI are noted. Approaches to implementation of the Trusted Boot Module for a computing platform that supports UEFI are proposed.

Key words: information security, unauthorized access protection, trusted boot, firmware, BIOS, UEFI, trusted boot module.

В работе рассмотрено обеспечение доверенной загрузки в вычислительных системах с расширяемым интерфейсом встроенного программного обеспечения (UEFI [1]). Обеспечение доверенной загрузки средств вычислительной техники должно применяться согласно Приказам ФСТЭК № 17 и 21 для 1 и 2 класса защищенности информационной системы [2, 3]. Главная задача средств обеспечения доверенной загрузки – это запуск различных операционных систем (ОС) только с заранее определенных постоянных носителей (например, только с жесткого диска, установленного внутри системного блока) после успешного завершения специальных процедур: проверки целостности аппаратных и программных средств и аутентификации пользователя.

Как доказано в [4], аппаратные модули доверенной загрузки (МДЗ) имеют значительные преимущества перед программными средствами. Исходя из этого, для обеспечения более высокого уровня защищенности требуется применять именно аппаратно-программные средства доверенной загрузки. Модуль доверенной загрузки представляет собой комплекс аппаратно-программных средств (плата, аппаратные средства идентификации и аутентификации, программное обеспечение для поддерживаемых ОС), устанавливаемый на рабочее место вычислительной системы. Модули доверенной загрузки обеспечивают выполнение следующих основных функций. В первую очередь, это идентификация и аутентификация пользователей до загрузки ОС с помощью персональных электронных идентификаторов (USB-ключи, смарт-карты, идентификаторы iButton и др.), а также контроль целостности программного и аппаратного обеспечения компьютера до загрузки ОС. Затем – блокировка несанкционированной загрузки с внешних съемных носителей, функционирование сторожевого таймера, позволяющего блокировать работу компьютера при условии, что после его включения и по истечении определенного времени управление не было передано плате МДЗ, и контроль работоспособности основных компонентов МДЗ (энергонезависимой памяти, идентификаторов, датчика случайных чисел и др.). И наконец, регистрация действий пользователей и совместная работа с внешними приложениями (передача параметров авторизации в ОС, интеграция с программными средствами защиты информации и др.) [5].

На протяжении многих лет и до настоящего времени средства доверенной загрузки разрабатывались и совершенствовались в условиях использования совместно с «традиционными» BIOS (basic input/output system) [6]. Это обстоятельство определяло их архитектуру и принципы функционирования. Начиная с 2009 г. вычислительные системы оснащаются UEFI вместо устаревшего BIOS. Цель разработки интерфейса UEFI – опираясь на современное оборудование, обеспечить возможность прямого взаимодействия предоперационной среды с аппаратными компонентами с помощью быстрых блоковых операций ввода-вывода без использования традиционных аппаратных прерываний. UEFI не имеет ограничений, присутствующих в BIOS, и тем самым не затормаживает развитие современных вычислительных систем. UEFI – это гибко программируемый интерфейс, псевдооперационная система, способная выходить в Интернет. Стандарт UEFI включает в себя возможность написания универсальных драйверов устройств и запуска специальных программ, написанных на высокоуровневом языке программирования. К числу таких приложений относится, например, оболочка UEFI (UEFI shell). Следует подчеркнуть, что пользователю доступно функциональное меню настройки UEFI (BIOS Setup). На текущий момент материнские платы большинства производителей разработаны по стандарту UEFI.

Рассмотрим цепочку загрузки UEFI [1]. После нажатия кнопки включения источник питания выполняет самотестирование. После тестирования питания материнская плата снимает сигнал сброса с соответствующего входа процессора. Процессор начинает работу в реальном режиме и приступает к выполнению инструкций по адресу FFFF:0000, считываемых из флэш-памяти. В ней хранится последовательность команд, осуществляющая переход на исполняемый код UEFI.

На рис. 1 приведен процесс загрузки UEFI от момента передачи управления в фазу начальной инициализации (SEC). Как видно на рисунке, процесс загрузки UEFI состоит из начальной инициализации, предварительной инициализации, основной инициализации, выбора устройства загрузки, промежуточной загрузки и работы ОС.

Результатом выполнения фазы начальной инициализации является обновленный микрокод процессора, переход в защищенный режим и передача управления в фазу предварительной инициализации.

В фазе предварительной инициализации (PEI) выполняется передача данных через основные независимые контейнеры (HOB) в фазу основной инициализации. Данные, передаваемые в контейнерах, содержат информацию о системной памяти, наборе команд процессора и информацию об обнаруженных микропрограммах устройств.

На этапе основной инициализации (Driver Execution Environment, DXE) выполняется обработка полученного списка контейнеров, запуск сервисов загрузки (Boot Services), среды выполнения (Runtime Services) и основной инициализации (DXE Services). На данном этапе драйверы проводят окончательную инициализацию аппаратуры и предоставляют абстракцию для системных служб и загрузочных устройств. После завершения работы всех драйверов управление передается менеджеру загрузки (BDS).

В фазе промежуточной загрузки (TSL) управление передается начальному загрузчику ОС. На данном этапе загрузчик ОС в зависимости от файла конфигурации загружает ядро ОС, располо-

женное в файловой системе, и передает на него управление. После этого код UEFI выгружается, кроме сервисов среды выполнения (Runtime Services).

Современные МДЗ часто реализуются на базе платы расширения PCI. Был рассмотрен сертифицированный аппаратно-программный модуль доверенной загрузки «Соболь» [7] и проведен эксперимент с материнской платой ASUS H81M-C, показывающий, что «Соболь» можно обойти стандартными средствами на материнских платах, поддерживающих UEFI. Это связано с работой модуля «Соболь» в legacy режиме (загрузка ПЭВМ с использованием сервисов и прерываний BIOS).

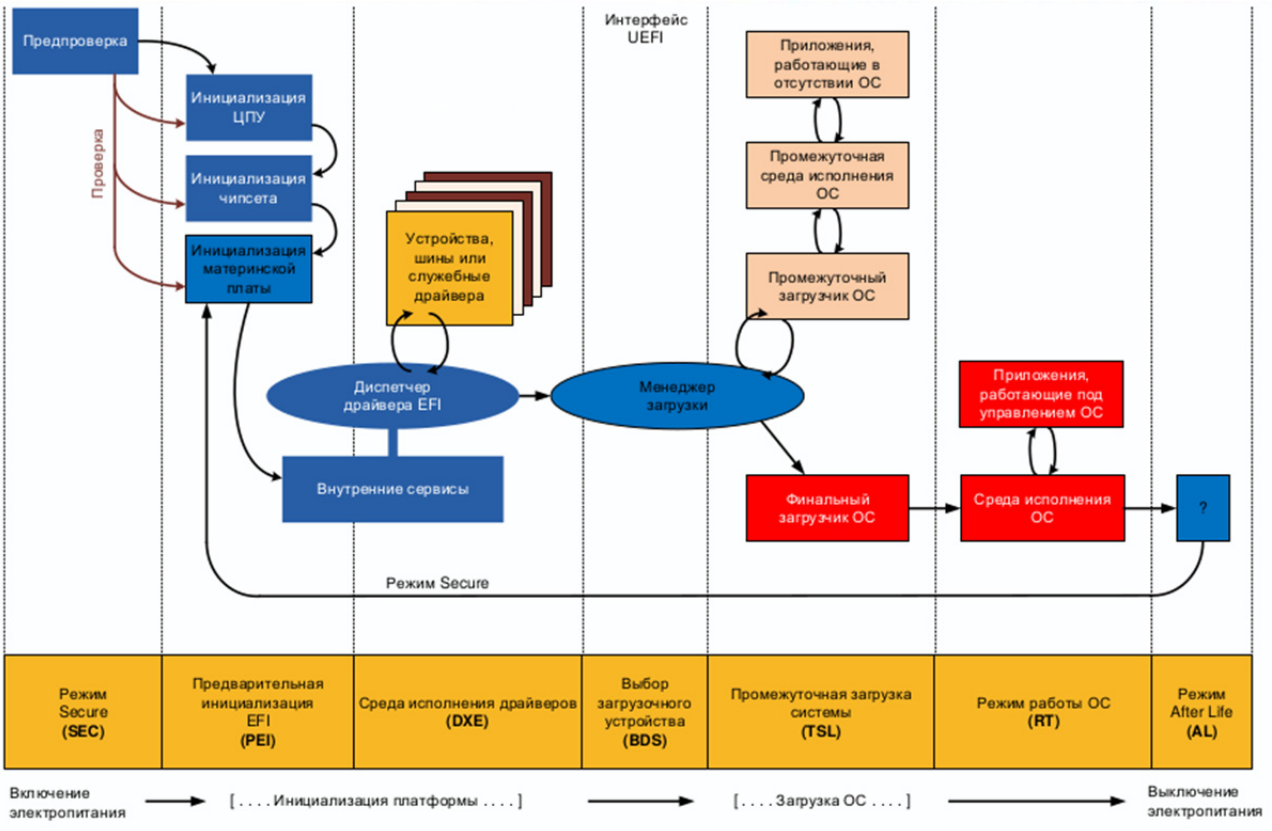

Рис. 1. Последовательность инициализации UEFI firmware и загрузки ОС

В UEFI-системах реализован механизм legacy загрузки для обеспечения совместимости, например, для того чтобы в новом компьютере (с UEFI) заработала старая сетевая плата. Другой пример – запуск старой ОС Windows XP на компьютере с новым UEFI BIOS. В UEFI BIOS могут отличаться режимы инициализации плат расширения и режимы загрузки с носителей информации. Каждое из этих устройств может участвовать в процессе загрузки компьютера в legacy и/или UEFI режимах. Как правило, режим работы каждого из устройств можно настроить.

Возможно, что в скором времени производители программного и аппаратного обеспечения откажутся от режима совместимости, и возможность работы с legacy-устройствами будет исключена.

Для запуска устаревших устройств используется модуль эмуляции (Compatibility Support Module, CSM). Этот модуль реализует сервисы BIOS, работая под управлением платформы UEFI. CSM позволяет UEFI загружать устаревшие ОС и использовать legacy Option ROM (встроенное программное обеспечения плат расширения, написанное с использованием прерываний BIOS). Как видно из рис. 2, CSM является самостоятельным модулем и в зависимости от выбранного устройства и режима загрузки может не инициализироваться [8]. В связи с этим загрузка в UEFI-режиме со стороны МДЗ, перехватывающих прерывания BIOS, никак не контролируется. Поддержка UEFI загрузки добавлена в ОС семейства Windows начиная с Windows 7 x64 и Windows Server 2008. Поэтому если материнская плата имеет спецификацию UEFI, то МДЗ должен работать в этом режиме.

Рассмотрим специфичные для UEFI функции получения управления модулем доверенной загрузки и передачи управления доверенной ОС.

Подходящей фазой для получения управления модулем доверенной загрузки является так называемая фаза загрузки (Boot Phase), во время которой функционируют загрузочные сервисы (Boot Services), так как все оборудование уже проинициализировано, но выбор устройства загрузки еще не

выполнялся. При этом ПО плат расширения (в том числе МДЗ) было проинициализировано на предыдущем этапе (DXE), что позволило ему установить свои обработчики для перехвата управления.

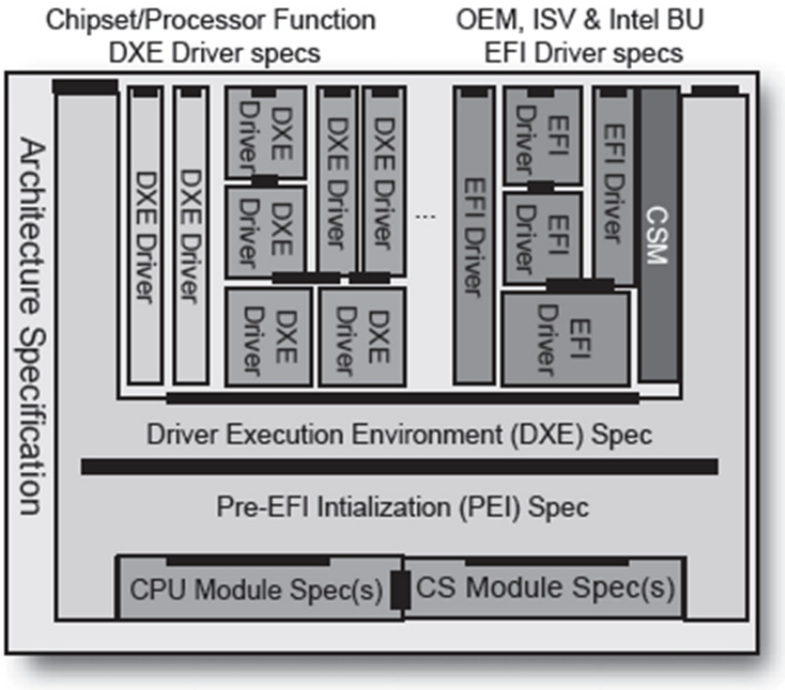

Рис. 2. Системные компоненты UEFI

Перехватывать управление можно по наступлению события *ReadyToBoot* – системное событие, оповещающее о готовности системы к загрузке ОС. Однако при проведении экспериментов с перехватом управления по наступлению события *ReadyToBoot*, в частности на материнской плате ASUS H81M-C выяснилось, что в этот момент ПЭВМ еще не полностью инициализирована. Перехват МДЗ управления на этом этапе приводил к некорректной работе системы. Поэтому был предложен и проверен вариант подменой стандартной функции *LoadImage*, которая выполняет чтение загрузочного образа. Вызов *LoadImage* выполняется на более поздней стадии загрузки, таким образом МДЗ удается успешно выполнять свои функции.

Для доверенной загрузки может быть применено два подхода. Первый заключается в контроле процесса загрузки, выполняемого UEFI BIOS. Перед возвратом управления МДЗ в UEFI BIOS выполняется подмена следующих функций UEFI BIOS: *LoadImage*, *ReadBlocks* и *WriteBlocks* для каждого загрузочного носителя – выполняется отслеживание чтения, записи и запуска загрузочного кода. Также устанавливается перехватчик на системное событие *ExitBootServices*. После возврата управления в UEFI BIOS новый обработчик функции *LoadImage* будет блокировать все попытки загрузки с внешних носителей. При выполнении загрузки с доверенного носителя загрузчик ОС инициирует вызов *ExitBootServices*. Обработчик этого события в МДЗ должен восстановить все подмененные функции в первоначальное состояние.

Во втором подходе управление UEFI BIOS от МДЗ не передается. МДЗ выполняет поиск доверенного носителя. Напрямую с доверенного носителя вычитывается загрузочный образ, начинается загрузка основной операционной системы путем передачи управления считанному образу.

Определение доверенных носителей может осуществляться через сигнатуру диска. В случае, если диск имеет разметку MBR, выполняется чтение его первого сектора и запоминается двойное машинное слово из содержимого этого сектора, расположенное по фиксированному смещению (так называемая сигнатура диска MBR). В случае, если диск имеет разметку GPT, аналогичным образом выполняется чтение его второго сектора и запоминается уникальный идентификатор из содержимого этого сектора, расположенный по фиксированному смещению (DiskGUID). Уникальность сигнатур обеспечивается средствами ОС, например, ОС Windows, которая не разрешает использование нескольких носителей с одинаковыми сигнатурами. При обнаружении совпадений ОС исправляет сигнатуры дисков.

Для усиления защиты можно подменить стандартные функции работы с устройствами ввода *ReadKeyStroke* и *WaitForKey*, тем самым заблокировав доступ к встроенному меню настроек BIOS на материнских платах, поддерживающих спецификацию UEFI.

Таким образом, в настоящее время существует необходимость разработки модуля доверенной загрузки для вычислительных систем, оснащенных UEFI BIOS. В статье показано, что модуль доверенной загрузки может быть реализован с использованием перехвата специфичных для UEFI функций.

Библиографический список

1. Unified Extensible Firmware Interface Specification Version 2.6. Unified Extensible Firmware Interface Forum. – 2016. – URL: http://www.uefi.org/sites/default/files/resources/UEFI%
2. Об утверждении Требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах : приказ Федеральной службы по техническому и экспортному контролю (ФСТЭК России) № 17 от 11 февраля 2013 г. // Российская газета. 2013. – Федеральный выпуск № 6112 (136).
3. Об утверждении Состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных : приказ Федеральной службы по техническому и экспортному контролю (ФСТЭК России) № 21 от 18 февраля 2013 г. // Российская газета. 2013. – Федеральный выпуск № 6083 (107).
4. Кашубина, А. П. Анализ методов обеспечения доверенной загрузки в средствах вычислительной техники / А. П. Кашубина // Актуальные направления развития систем охраны специальной связи и информации для нужд органов государственной власти Российской Федерации : материалы и докл. X Всерос. межведомственной науч. конф. (Орёл, 7–8 февраля 2017 г.) : в 11 ч. Ч. 10. – Орёл : Академия ФСО России, 2017. – С. 149–150.
5. Кашубина, А. П. Обзор методов обеспечения криптографической целостности в различных программно-аппаратных средах / А. П. Кашубина // Актуальные направления научных исследований XXI века: теория и практика : сб. работ Междунар. науч.-практ. конф. «Молодежный форум: технические и математические науки». – Воронеж : Изд-во ФГБОУ ВО «ВГЛТУ», 2015. – С. 284–288.
6. Plug and Play BIOS Specification Version 1.0A. Intel Corporation. – 1994. – URL: http://download.intel.com/support/motherboards/desktop/sb/pnpbiosspecificationv10a.pdf
7. Программно-аппаратный комплекс Соболь Версия 3.0. Руководство администратора. ООО «Код Безопасности». – 2016. – URL: https://www.securitycode.ru/upload/documentation/sobol/Sobol_FSB_Admin_Guide.pdf.
8. Zimmer, V. Beyond BIOS: Implementing the Unified Extensible Firmware Interface with Intel's Framework / Vincent Zimmer, Michael Rothman, Robert Hale. – USA : Intel Press, 2006.

Кашубина, А. П.

Подходы к реализации модуля доверенной загрузки для вычислительной платформы с технологией UEFI / А. П. Кашубина, А. П. Иванов // Инжиниринг и технологии. – 2017. – Vol. 2(2). – DOI 10.21685/2587-7704-2017-2-2-4