УДК 004.056.53
DOI: 10.17586/0021-3454-2018-61-1-41-46

АНАЛИЗ БЕЗОПАСНОСТИ УДАЛЕННОГО ДОСТУПА СРЕДСТВАМИ INTEL MANAGEMENT ENGINE

А. А. Оголюк, А. В. Шабалин

Университет ИТМО, 197101, Санкт-Петербург, Россия
E-mail: antoxa940@gmail.com

Рассматриваются скрытые детали реализации подсистем UEFI BIOS и Intel Management Engine современных компьютеров, основанных на базе платформы x86. Представлен анализ безопасности подсистем и возможных последствий их дискредитации. Описываются принципы, на основе которых могут быть реализованы атаки на рассматриваемые подсистемы. Исследуется возможность изменения сложившейся ситуации и предлагаются подходы к повышению безопасности систем, работающих на базе платформы Intel x86.

Ключевые слова: *Intel Management Engine, UEFI BIOS, безопасность, удаленный доступ, платформа x86*

Введение. Intel Management Engine (IME) — важная составляющая платформы x86 и крупный компонент подсистемы UEFI BIOS современных компьютеров. Эта система практически недоступна для пользователя или администратора и содержит защищенный и привилегированный исполняемый код, доступ к которому, однако, может быть получен из обычного окружения операционной системы (ОС). Несмотря на это многие специалисты по информационной безопасности не знают о ее существовании.

История создания IME относится к 2006 г., когда компания Intel представила подсистему AMT, основной задачей которой является обеспечение возможности удаленного администрирования для систем на базе Intel.

В отличие от предыдущего поколения технологий удаленного управления, доступных только в серверах, работающих на чипсетах Intel, эта система работала и на персональных компьютерах. Технически это было реализовано добавлением в чипсет нового микроконтроллера, отвечающего за функционирование системы, что позволило придать AMT дополнительные эффективные функции, а именно [1]:

— интегрированный http(s) сервер;

— внеполосный (out-of-band) доступ к встроенному сетевому адаптеру, в том числе управление всеми входящими и исходящими сетевыми пакетами;

— доступ ко всем устройствам ввода и вывода;

— доступ к энергонезависимой памяти и т.д.

Этот микроконтроллер (и вся AMT-подсистема) начинает работать в момент подключения компьютера к сети питания. Это означает, что подсистема функционирует даже при выключенном компьютере.

Подсистема AMT постоянно развивалась и со временем стала частью подсистемы Intel Management Engine. Так, в 2007 г. компания Intel представила новую версию со следующим дополнительным функционалом:

— прямой доступ к RAM;

— прямой доступ к памяти встроенного видеоадаптера (что позволяет следить за видеопотоком, выводимым на экран);

— поддержка KVM и т.д.

Также многие стандартные возможности BIOS постепенно переносились в IME. Изначально эту подсистему поддерживали только флагманские материнские платы с чипсетами

Intel линейки Q, которые отличались высокой ценой, что негативно сказывалось на их популярности. Впоследствии эта подсистема была выпущена на всех чипсетах Intel, будь то серверный, десктопный или мобильный сегмент. В первых версиях подсистемы использовались микроконтроллеры с архитектурой SPARC и ARC32, однако в современных системах они заменены на x86 контроллеры, что упрощает процесс реверс-инжиниринга кода Management Engine [2].

Современная реализация IME включает следующие компоненты:

— микроконтроллер (содержит интегрированную ROM);

— регион в SPI флеш-памяти, в котором хранится прошивка ME;

— выделенная RAM (около 32 Мб);

— модули UEFI BIOS DXE/SME;

— интерфейс Management Engine;

— сетевой контроллер прямого доступа к адаптеру Ethernet.

Подводя итог вышесказанному, можно отметить, что Intel Management Engine представляет новый уровень исполнения кода (дополнительный к хорошо известным ring 0 – 3 базовой архитектуры i386, SMM и уровню гипервизора). Исполняемый на этом уровне (ME) код полностью скрыт и не может контролироваться с остальных уровней исполнения (в том числе, ОС или BIOS).

Подсистема Management Engine хранится в памяти чипа SPI, которая содержит следующие регионы [3]:

— дескриптор, описывающий все области памяти SPI и атрибуты доступа к ним;

— прошивку UEFI BIOS;

— прошивку Management Engine;

— GbE (прошивку сетевого адаптера Ethernet);

— PDR (дополнительные модули производителя).

От перезаписи кодовый сегмент ME защищает цифровая подпись, которая вместе с открытым ключом RSA Intel записана в начале этого же раздела. Таким образом, любой имеет возможность проверить аутентичность кода системы. Злоумышленник же не может заменить включенный открытый ключ своим, так как подпись и целостность проверяются кодом загрузки (находится в контроллере ROM). Дополнительно код ME защищен от перезаписи установленным атрибутом региона памяти только на чтение.

Учитывая перечисленные особенности подсистемы Intel Management Engine, можно с уверенностью утверждать, что любой компромисс в системе ее безопасности может иметь серьезные последствия. Если злоумышленник сможет перезаписать или внедрить свой код в подсистему IME, то он получит множество возможностей по организации атак, например:

— полностью невидимое для ОС и BIOS вредоносное программное обеспечение (ПО);

— постоянное размещение вредоносного кода;

— восстановление уничтоженного кода (исполняемого в ОС) вредоносным ПО, внедренным в прошивку ME;

— полный доступ к аппаратным ресурсам;

— привилегированное исполнение кода в реальном времени.

Следует заметить, что на текущий момент не известны реализации атак на подсистему Intel Management Engine, но это не означает, что она полностью безопасна. Как и любая другая система, она подвержена уязвимости утечки закрытого ключа. Если такой ключ станет доступен злоумышленнику, то защита от перезаписи и внедрения вредоносного кода в подсистему Intel Management пропадет. Основным способом обхода защиты от записи в область памяти Management Engine может стать аппаратное программирование чипа SPI (например, с использованием программатора CH341A) или использование рассматриваемых далее программных методов.

Векторы атак на подсистему Intel Management Engine. Вне зависимости от того, что не известна реализация атаки на подсистему Intel Management Engine, можно представить подходы к ее организации:

— дамп выделенной памяти ME путем снятия битов блокировки и атрибутов региона памяти;

— атака методом холодной перезагрузки (Cold Boot) — RAM-модули меняются местами для чтения или замены содержимого (требуется медленная память);

— увеличение размера выделенной памяти ME на стадии инициализации, что приведет к необходимости использования дополнительной выделенной памяти; при следующей загрузке расширение будет удалено и появится доступ к ранее записанной области памяти;

— использование самостоятельно написанных Java-апплетов, запрашивающих подсистему ME для их выполнения (одна из функций AMT);

— реверс-инжиниринг приложений Intel Windows (C#, C++, Java) которые взаимодействуют с подсистемой ME.

Другая причина для беспокойства — это широкое использование подсистемы Intel Management Engine в современных системах. Компания Intel заявляет, что все функции удаленного доступа отключены на большинстве компьютеров (за исключением высокопроизводительных и серверных решений). Но исследование прошивки (все основанные на Intel платформы используют код Intel) показывает, что код, контролирующий удаленное управление, присутствует во всех компьютерах, он просто не активен. Это, однако, не может гарантировать, что некий „магический“ пароль или сетевой пакет не активирует этот код, заставляя вашу систему собирать информацию о вас или предоставляя удаленный доступ. Наибольшая же опасность заключается в том, что практически невозможно исправить эту ситуацию. Исходный код прошивки Intel полностью закрыт и никогда не будет доступен для анализа, а единственной альтернативной x86-совместимой платформой является решение, выпускаемое компанией AMD, но оно распространено намного меньше и имеет собственную похожую на IME подсистему. Существуют, конечно, несовместимые платформы (такие, как ARM, SPARC и т.п.), но для широкого их использования на рынке персональных компьютеров и ноутбуков нет очевидных причин. Вариант с модификацией прошивки для удаления нежелательного блока кода AMT тоже не представляется возможным, так как проводятся проверки RSA подписи и целостности.

Векторы атак на UEFI BIOS. UEFI BIOS — наиболее известная система в основанных на Intel компьютерах. EFI (Extensible Firmware Interface) — стандарт, заменивший BIOS (Basic Input Output System) в период 2004—2006 гг. В 2005 г. Intel внесла этот стандарт в Unified Extensible Firmware Interface Forum, основанный несколькими ведущими технологическими компаниями с целью модернизировать процесс загрузки системы. В связи с этим EFI был переименован в UEFI, однако большая часть документации использует оба термина.

Изначально EFI был представлен для платформы IA64, но затем стал стандартом для всех решений на базе x86, а также многих других. Весь код EFI находится в энергонезависимой памяти и является первым запускаемым кодом после запуска системы.

В отличие от BIOS, код EFI (UEFI) работает в 32-разрядном защищенном режиме.

Стадии загрузки UEFI BIOS [4]:

— SEC (Security);

— PEI (Pre EFI);

— DXE (Driver Execution Environment);

— BDS (Boot Device Select);

— TSL (Transient System Load);

— RT (Run Time, исполнение кода ОС);

— AL (After Life, выключение).

Загрузка системы начинается с этапа SEC (этап безопасности). На этом этапе выполняются инициализация временной памяти и проверка целостности встроенного программного обеспечения, а также инициализация фазы PEI, которая служит для целей, аналогичных прежней стадии инициализации BIOS (инициализация RAM, копирование из энергонезависимой памяти

в RAM, выполнение модулей PEI, инициализация интерфейсов и т. д.). После этого UEFI создает структурированное пространство DXE для запуска драйверов и собственных служб. Между DXE-драйверами и службами могут существовать зависимые исполняемые файлы, которые также нуждаются в загрузке. Помимо этого, фаза DXE выполняет аппаратную инициализацию и создает абстрактный интерфейс доступа к аппаратным средствам (для служб). В отличие от MBR, используемого в BIOS, здесь используется системный раздел EFI (FAT32). UEFI пытается найти загрузочный код в разделе EFI и передать ему управление. В случае если найти этот код не удалось, то выводится сообщение об ошибке, иначе загружается операционная система. Системный загрузчик, драйверы и службы операционной системы могут обращаться к прошивке через специально разработанные протоколы. После полной загрузки операционная система может получить доступ к некоторым из этих интерфейсов через службы времени исполнения EFI. DXE-этап включает в себя подмодуль SMM (System Management Mode) [5].

Прошивка UEFI BIOS также является важным элементом системы безопасности компьютера. Злоумышленник, внедривший свой код в сегмент кода UEFI BIOS, получит такие возможности, как постоянное выполнение вредоносного кода; удаление, переустановка ОС, форматирование диска; доступ к аппаратным ресурсам (RAM, CPU, Ethernet и видеоадаптер). Основным объектом атаки в данном случае является флеш-память SPI-чипа. Если в находящемся в ней коде не проводить проверку целостности исполняемых модулей UEFI BIOS, то, используя программатор, внедрить собственный код проще.

Кроме этого, злоумышленника может заинтересовать BIOS Setup в качестве объекта атаки. Реализованные на практике варианты атаки на BIOS Setup сводятся к изменению значения переменной NVRAM “SETUP”. После загрузки в оболочку UEFI злоумышленник может изменить переменную SETUP (с использованием самостоятельно написанного EFI приложения или команды “set var”). Переменная SETUP содержит множество важных системных настроек, в том числе и бит блокировки SPI (который запрещает программную перезапись прошивки). Адрес бита может быть различным в зависимости от конечной платформы и модели компьютера, но может быть получен из образа BIOS, который, как правило, доступен на сайте производителя.

Самый простой способ перезаписи области UEFI BIOS — это использование утилит или дисков восстановления разработчика системы. Такое ПО официально не доступно, но может быть легко найдено в сети Интернет. Помимо этого, программное обеспечение UEFI BIOS доступно на официальных сайтах производителя (Lenovo, HP, ACER и т.д.) в составе так называемых дисков восстановления, предназначенных для возврата компьютера в рабочее состояние после ошибки BIOS.

Наиболее известными инструментами для работы с этой системой являются Intel Management Environment System Tools и утилита Intel Flasher, поскольку они обходят проверку целостности образа прошивки и просто записывают образ в область памяти Management Engine или UEFI BIOS [6].

Таким образом, при использовании инструментов Intel ME становится меньше проблем с записью модифицированной прошивки в память SPI. В случае если выключена блокировка от записи UEFI, модифицированное микропрограммное обеспечение может быть записано на чип SPI программным способом (с помощью Intel ME Tools). Этот процесс может быть автоматизирован. Более того, это „обновление“ может быть выполнено не только из DOS или оболочки UEFI, но и из самой Windows (у Intel есть исполняемые файлы для всех операционных систем, включая Linux). Если область памяти UEFI BIOS защищена от записи, то может быть использован описанный выше вариант атаки на BIOS Setup с изменением переменной NVRAM “SETUP”.

Другие векторы атак. Помимо атак, нацеленных на IME и UEFI BIOS, существуют и другие векторы атак, среди которых можно выделить:

— выполнение кода в SMM (System Management Mode);

— внедрение кода во встроенное ПО устройств PCI (Ethernet, Thunderbolt и т.п.);

— отключение механизма Secure Boot.

SMM — это еще один привилегированный режим исполнения кода на платформе x86 (начиная с i486). SMM-режим активируется прерыванием SMI, которое может быть сгенерировано [7]:

— контроллером USB (в режиме USB Legacy);

— подсистемой IME;

— регистрами GPIO;

— таймером SMI;

— чипсетом SMI (по доступу к порту ввода/вывода);

— ACPI SMI (режим сна и т.п.).

По умолчанию код, выполняемый в режиме SMM, может получить полный доступ к оперативной памяти и доступ ко всем подключенным устройствам. Также в момент выполнения SMM код не доступен из ОС, система может только видеть, что выполняется SMM-режим, и не более. Таким образом, режим исполнения кода SMM является привлекательной целью для злоумышленника. Главное направление атаки в этом случае заключается в попытке генерации программного прерывания SMI и поиске уязвимостей в исполняемом коде SMM. Современные атаки на SMM основаны на изменении кода поддержки SMM, находящегося в RAM, а не в защищенной SRAM. Внедрение в эту память кода и инициирование прерывания SMI теоретически позволяет выполнять вредоносный код в режиме SMM. Предположительно, большинство систем, разработанных до 2015 г., когда компания Intel представила рекомендации по предотвращению атак, уязвимы для атак этого рода.

Secure Boot — это последнее направление атаки, рассматриваемое в рамках статьи. UEFI BIOS Secure Boot-режим предотвращает исполнение неавторизированного загрузочного кода. Целостность и подпись загрузочного кода проверяются с использованием открытых ключей, располагающихся в NVRAM (по умолчанию там хранятся ключи Microsoft, разрешая выполнение только загрузочного кода Microsoft Windows). Теоретически вредоносный загрузочный код может быть подписан закрытым ключом Microsoft (официальный сертификат со сгенерированными ключами можно купить у партнеров Microsoft) и в этом случае он пройдет проверку безопасности при загрузке. Например, известный загрузчик Canonical Ubuntu Linux, созданный другой компанией, однако, работает с Secure Boot и доступен на всех компьютерах.

Заключение. Приходится признать, что не существует простого и универсального способа реализовать на практике атаки на основе описанных выше уязвимостей и векторов атак. Это связано с тем, что платформа Intel и все коды встроенного программного обеспечения закрыты, поэтому их невозможно полностью исследовать даже с использованием технологий обратной инженерии. Однако реализация атаки теоретически возможна, что усугубляется широким распространением IME. В ближайшем будущем эту ситуацию изменить невозможно (альтернативы платформам Intel и AMD x86 не существует), однако можно указать направления для ее решения и даже реализовать некоторые защитные решения без поддержки Intel (которая очень мала). Это могут быть:

— отключение всего SMM-кода;

— отключение всех внешних компонентов прошивки;

— отключение S3 Bootscript (после спящего режима);

— широкое использование монитора транзакций SMI (для обнаружения злонамеренных вызовов SMI);

— включение режима Secure Boot;

— включение пароля BIOS;

— обширная обратная инженерия образцов прошивок для поиска уязвимостей;

— обзор кода (открытых систем на базе UEFI, таких как Tiano-Core).

СПИСОК ЛИТЕРАТУРЫ

1. *Щеглов А. Ю., Щеглов К. А.* Анализ и проектирование защиты информационных систем. Контроль доступа к компьютерным ресурсам. Методы, модели, технические решения. СПб: Профессиональная литература, 2017.
2. *Kumar A.* Active Platform Management Demystified: Unleashing the Power of Intel VPro Technology. N.Y.: Intel Press, 2009.
3. *Ruan X.* Platform Embedded Security Technology Revealed: Safeguarding the Future of Computing with Intel Embedded Security and Management Engine. N.Y.: APress, 2014.
4. *Rothman M., Xing G., Wang Y., Gong J.* Reducing platform boot Time // Intel White Paper. 2011.
5. https://software.intel.com/en-us/amt-sdk
6. http://www.win-raid.com/t596f39-Intel-Management-Engine-Drivers-Firmware-amp-System-Tools.html
7. http://blogs.phoenix.com/phoenix_technologies_bios/uefi/

Сведения об авторах

Александр Александрович Оголюк — канд. техн. наук, доцент; Университет ИТМО, кафедра вычислительной техники; E-mail: xms2007@yandex.ru

Антон Владимирович Шабалин — магистрант; Университет ИТМО, кафедра вычислительной техники; E-mail: antoxa940@gmail.com

Поступила в редакцию
24.04.17 г.

Ссылка для цитирования: *Оголюк А. А., Шабалин А. В.* Анализ безопасности удаленного доступа средствами Intel Management Engine // Изв. вузов. Приборостроение. 2018. Т. 61, № 1. С. 41—46.

ANALYSIS OF REMOTE ACCESS SAFETY USING THE INTEL MANAGEMENT ENGINE

A. A. Ogolyuk, A. V. Shabalin

ITMO University, 197101, St. Petersburg, Russia
E-mail: antoxa940@gmail.com

Several hidden details of the implementation of UEFI BIOS subsystems and Intel Management Engine of modern x86-based computers are discussed. Results of security analysis of the described subsystems and the possible consequences of their discredit ar presented. Basic principles of possible attacks on the subsystems under consideration are described. The possibility of changing the current situation is explored and approaches are proposed to improve the security of the systems based on the Intel x86 platform.

Keywords: Intel Management Engine, UEFI BIOS, security, remote access, x86

Data on authors

Alexander A. Ogolyuk — PhD, Associate Professor; ITMO University, Department of Computation Technologies; E-mail: xms2007@yandex.ru

Anton V. Shabalin — Graduate Student; ITMO University, Department of Computation Technologies; E-mail: antoxa940@gmail.com

For citation: Ogolyuk A. A., Shabalin A. V. Analysis of remote access safety using the Intel Management Engine. *Journal of Instrument Engineering*. 2018. Vol. 61, N 1. P. 41—46 (in Russian).

DOI: 10.17586/0021-3454-2018-61-1-41-46