УДК 004.451+004.424.46
К. П. СТОРЧАК, доктор техн. наук, доцент;
Н. К. ШАТОХІНА, канд. техн. наук, доцент;
В. О. ХОМЕНЧУК, аспірант,
Державний університет телекомунікацій, Київ

РОЗШИРЕННЯ ФУНКЦІЙ UEFI ЗА ДОПОМОГОЮ МОДИФІКАЦІЇ ВИХІДНОГО КОДУ

Розглянуто можливість розширення стандартного функціоналу за допомогою редагування бінарного вихідного коду UEFI. Як приклад запропоновано додавання функціоналу Nvme для застарілої материнської плати. Проаналізовано метод інсталяції бінарного вихідного коду за межами UEFI.

Ключові слова: UEFI; розширення функціоналу; модифікація бінарного коду.

Вступ

Базова система введення-виведення (BIOS) давно перестала вважатися найпростішим прошиванням, головною метою якого є ініціалізація і тестування на низькому рівні апаратних компонентів комп'ютера для подальшого передавання керування завантажувачу операційної системи (ОС). Із моменту введення UEFI набуває рис спрощеної сучасної операційної системи зі своїми фазами завантаження і механізмами забезпечення безпеки, зокрема реалізації різних криптографічних функцій.

Стаття має ознайомчий характер. Не рекомендовано повторювати описані в статі дії, аби не нашкодити вашому апаратному забезпеченню.

Основна частина

UEFI (*Unified Extensible Firmware Interface* — інтерфейс розширюваної «прошивки») — інтерфейс між операційною системою і мікропрограмами, які керують низькорівневими функціями комп'ютерного обладнання [1].

Основне призначення UEFI: коректно ініціалізувати обладнання під час увімкнення системи і передати керування завантажувачу операційної системи. UEFI призначено для заміни BIOS — інтерфейсу, який традиційно використовується всіма IBM PC-сумісними персональними комп'ютерами. Першу специфікацію UEFI (тоді ще просто «EFI») було розроблено компанією Intel, пізніше від першої назви відмовилися і остання версія стандарту носить назву *Unified Extensible Firmware Interface* (UEFI). Нині розробкою UEFI опікується Unified EFI Forum [1].

UEFI також дає можливість розширювати прошивку платформи, завантажуючи виконувані образи — драйвери або програми. Зазначені образи являють собою клас файлів, визначених специфікацією UEFI, які містять виконуваний код. Завантажені образи отримують доступ до сервісів, що також визначено специфікацією UEFI (boot services, runtime services). Виконувані образи можуть бути завантажені в пам'ять як вбудованим менеджером завантаження, так і іншими виконуваними образами [1].

Сьогодні, коли науково-технічний прогрес набуває дедалі більшого розвитку, з'являється багато нових технологій. Для більшості систем інтеграція нових вирішень передбачає оновлення програмних та апаратних компонентів. Старі материнські плати, що вже не підтримуються виробником, можуть отримати нове життя, тобто можна або просто додати функціонал, що більш витратно, або додати додаткові компоненти в бінарний вихідний код.

UEFI розміщується в мікросхемі ROM-пам'яті (*Read Only Memory*), що забезпечує постійну доступність UEFI незалежно від працездатності зовнішніх відносно материнської плати компонентів (наприклад, завантажувальних дисків).

Кожний із вихідних кодів можна подати у вигляді компонентів, які ієрархічно завантажується під час запуску системи. Кожний компонент має свою адресу та розмірність. Тому у разі додавання компонентів слід врахувати, що зміниться адреса наступних компонентів. Для простого ін'єктування компонентів існує додаток MMTool (рис. 1). Він дає можливість не лише додавати компоненти, а і витягувати їх з інших файлів.

Ін'єкція у вихідний бінарний код дасть змогу:

- використовувати режим роботи відеокарт NVIDIA SLI;
- застосовувати SSD диски з використанням Nvme в режимі UEFI boot;
- редагувати таблиці SLIC для активації ОС без доступу до мережі;
- змінювати параметри контролю системи внутрішнього моніторингу для забезпечення максимізації параметрів роботи процесора та інших комплектуючих;
- додавати підтримання апаратної віртуалізації.

Одним із прикладів додавання функцій — це додати підтримання апаратної віртуалізації комп'ютером, тобто необхідно щоб її підтримував центральний процесор комп'ютера і материнська плата. Взагалі, від материнської плати не потрібно

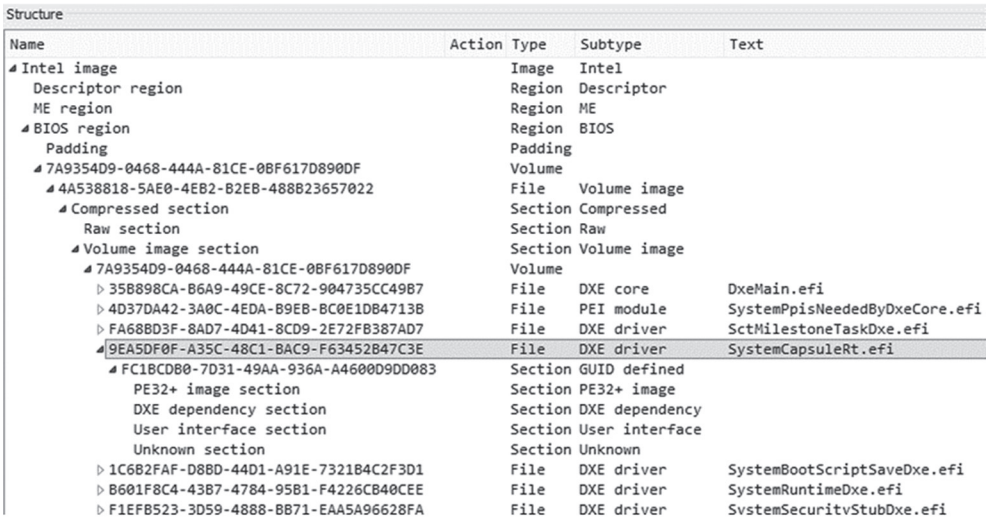

Рис. 1. Дерево компонентів, відображені в програмі UEFItool

ніяких особливих операцій із підтримання апаратної віртуалізації, за винятком того, що BIOS/UEFI материнської плати має просто ввімкнути це підтримання. Проте багато виробників материнських плат із різних причин штучно вимикають у BIOS/UEFI підтримання апаратної віртуалізації.

Для прикладу розглянемо додавання функції Nvme, Nvme boot для материнської плати Asus P8Z77-V LE та донора Asus Z97-PRO (firmware version 2702) (рис. 2). Саме починаючи з версії firmware version 2205, Asus Z97-PRO розпочала підтримання цих технологій.

Nvme — це специфікація доступу до твердотільних дисків (SSD), які приєднуються через шину PCI Express (PCIe). Хоча інтерфейс Advanced Host Controller Interface (AHCI) має перевагу сумісності програмного забезпечення, він не забезпечує оптимальної продуктивності [2].

Nvme було розроблено з нуля, використовуючи низьку затримку та паралелізм PCI Express SSD, а також виконуючи паралелізм сучасних процесорів, платформ та додатків. На високому рівні основні переваги Nvme над AHCI пов'язані з його здатністю використовувати паралелізм у хост-апаратному та програмному забезпеченні, що визначається різницею в глибині черг команд [2].

Першим кроком буде завантаження останніх версій firmware та експорту необхідних компонентів. Прошивка має в собі компоненти Nvme, NvmeSmm, NVMEINT13.

Під час імпортування компонентів у прошивку значення Vol. Index має збігатися зі значенням CSMCORE у новій прошивці. Компонент буде добавлено в кінець компонентів відповідного індексу.

Останнім кроком залишається завантажити прошивку в материнську плату. Майже кожна

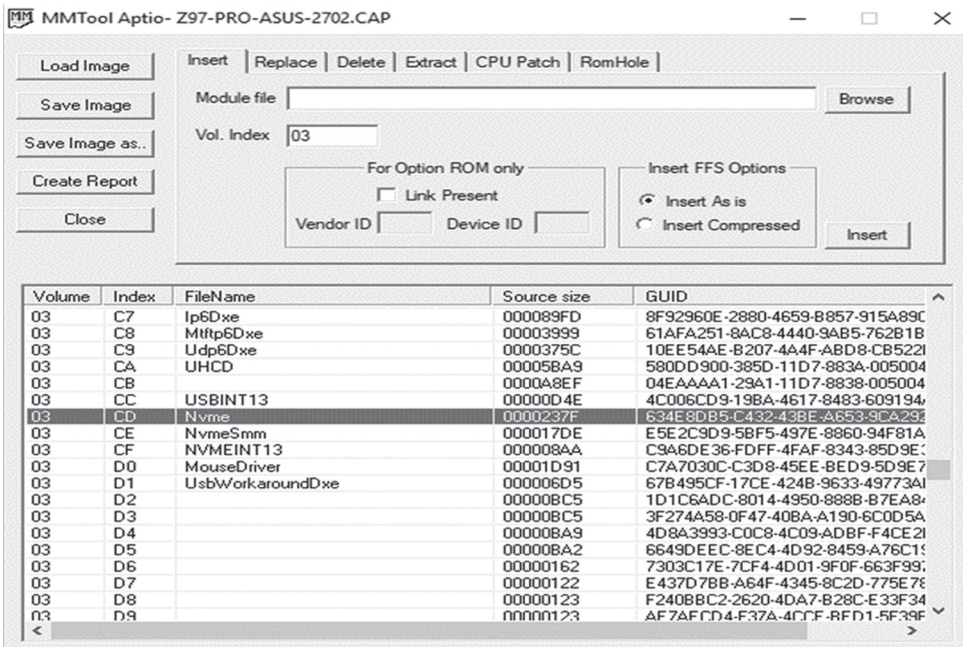

Рис. 2. Імпортування компонентів із прошивки донора

материнська плата має в собі один або кілька способів завантаження: завантаження відразу з операційної системи, завантаження з інтерфейсу UEFI, завантаження з USB пристрою тощо. Але всі ці способи захищено «захистим від дурня», що контролюється контрольною сумою (щоб випадково не завантажити невідповідну прошивку). У нашому разі контрольна сума буде відрізняться від оригінальної. Залишається тільки завантажити дані напряму в мікросхему пам'яті, де безпосередньо знаходиться прошивка. Для цього існує велика кількість програматорів.

Програматор CH341A застосовується для програмування мікросхем UEFI/BIOS комп'ютерів, ноутбуків, відеокарт, мультимедійних плеєрів, пам'яті телевізорів, ЖК-дисплеїв, маршрутизаторів, супутникових ресиверів та інших пристроїв (рис 3).

CH341A підтримує більшість режимів роботи, що пропонують виробники обладнання, а саме:

1. конвертор USB <-> USART. Застосувань йому маса, можна відновлювати мікросхеми після невдалого прошивання модемів та роутерів, відновлювати зіпсовані жорсткі диски, під'єднуватися до налагоджувальних інтерфейсів різного устаткування;

2. SPI-програматор для прошивання і відновлення UEFI/BIOS більш-менш нових ПК (виробництва 2008 року і новіших, масовий перехід на SPI-чіпи). Із упровадженням технології SecureBoot (і супутніх їй) прошивання модифікованого UEFI/BIOS перетворилося зі звичайної операції в процес, сповнений різними блокуваннями програмування UEFI/BIOS (перевірками версій, контрольних сум тощо). Зовнішній SPI-програматор вирішує ці проблеми повністю (рис. 4);

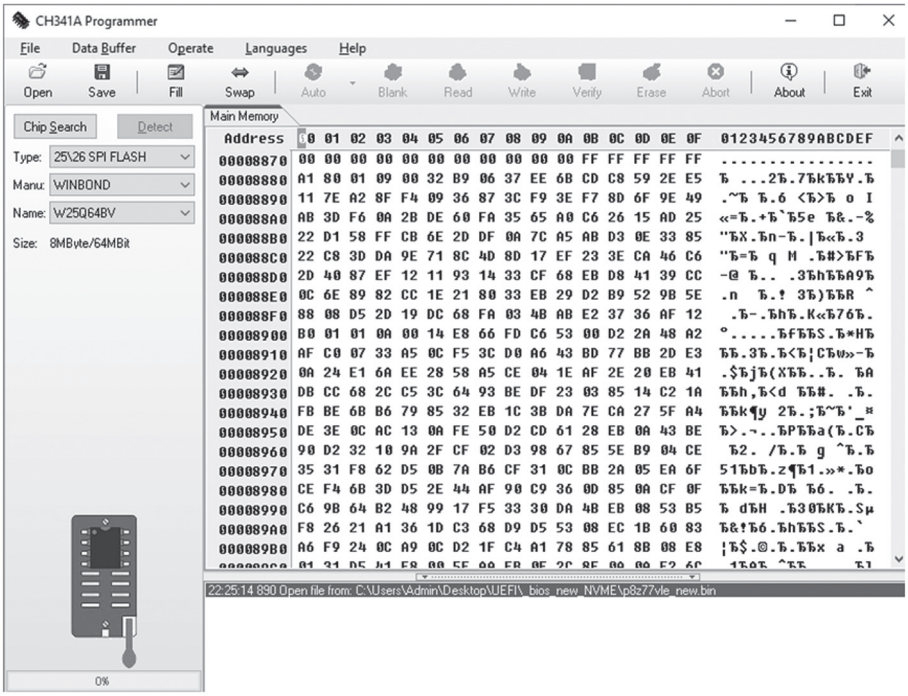

Рис. 3. Інтерфейс програми CH314A Programmer для інсталяції бінарного вихідного коду

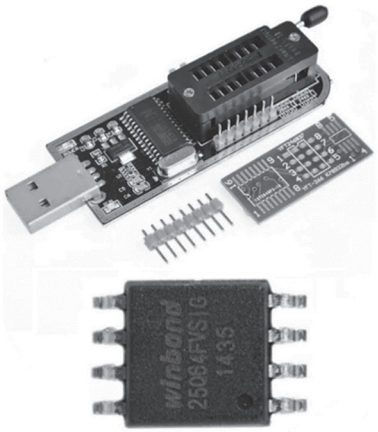

Рис. 4. Програматор мікросхем CH341A та мікросхема пам'яті

3. JTAG-наладник для різних мікроконтролерів;

4. I2C bus master, який можна застосовувати для контролю за VID процесором або відеокартою, і за нормальних умов I2C сумісний із SMBus і PCBus, які використовують низькошвидкісні периферії ПК.

Висновки

У статті було розглянуто можливість розширення функціоналу UEFI, розглянуто програмне забезпечення, що дасть змогу виконати необхідні дії над бінарним кодом. Описано алгоритм модифікації бінарного коду на прикладі додання функціоналу Nvme в режимі UEFI boot. Нові функції

зможуть продовжити актуальність певних моделей материнських плат, але існує ризик невдалої конфігурації, що може спричини погіршення роботи основних функцій.

Список використаної літератури

1. ***Unified*** *Extensible Firmware Interface [Електронний ресурс wikipedia.org]. URL: https://uk.wikipedia.org/wiki/UEFI (дата звернення 20.08.2019)*

2. ***NVMe*** *support for all Systems with an UEFI BIOS [Електронний ресурс win-raid.com]. URL: https://www.win-raid.com/t871f50-Guide-How-to-get-full-NVMe-support-for-all-Systems-with-an-AMI-UEFI-BIOS.html (дата звернення 20.08.2019)*

Рецензент: доктор техн наук, доцент **А. П. Бондарчук,** Державний університет телекомунікацій, Київ.

К. П. Сторчак, Н. К. Шатохина, В. О. Хоменчук

РАСШИРЕНИЕ ФУНКЦИЙ UEFI С ПОМОЩЬЮ МОДИФИКАЦИИ ИСХОДНОГО КОДА

Рассмотрена возможность расширения стандартного функционала с помощью изменения бинарного исходного кода UEFI. В качестве примера предложено добавление функционала Nvme для устаревшей материнской платы. Проанализирован метод инсталляция бинарного исходного кода за пределами UEFI.

Ключевые слова: UEFI; расширение функционала; модификация бинарного кода.

K. P. Storchak, N. K. Shatokhina, V. O. Khomenchuk

EXTENSION OF UEFI FUNCTIONS BY MODIFYING THE SOURCE CODE

The article discusses the possibility of extending standard functionality by editing UEFI binary source code. Adding Nvme functionality for an outdated motherboard is considered as an example. The method of installing binary source code outside of UEFI is considered.

The UEFI is housed in a Red Only Memory chip, which ensures that UEFI is permanently availabe regardless of the performance of the components external to the motherboard.

With the introduction of SecureBoot technology (and its related), the UEFI/BIOS modified firmware has evolved from a routine operation into a process filled with various UEFI/BIOS programming blocking (version checks, checksums, and more).

The article describes one of the possible methods of circumventing these restrictions.

Keywords: UEFI; expansion of functionality; binary code modification.