УДК 004.052.2

**ИССЛЕДОВАНИЕ УСТОЙЧИВОСТИ UEFI-СОВМЕСТИМЫХ**

**ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ**

**Захаров О.О.1, Дудкин А.С.1**

**Научный руководитель – кандидат технических наук Дудкин А.С.1**

**1Военно-космическая академия им. А.Ф.Можайского**

**Аннотация**

В статье представлено исследование устойчивости UEFI-совместимых программно-аппаратных комплексов: проведен краткий анализ возможных проблем, связанных с энергонезависимой памятью для хранения переменных и прошивки UEFI, последствий неверного или заведомо вредоносного воздействия на NVRAM, выдвинуты предложения и рекомендации по улучшению существующих средств антивирусной защиты и политик безопасности.

**Ключевые слова**

Информационная безопасность, компьютерные атаки, критическая информационная инфраструктура, устойчивость программно-аппаратных комплексов, унифицированный интерфейс расширяемой прошивки (UEFI).

Unified Extensible Firmware Interface (UEFI) – интерфейс между операционной системой и микропрограммами, предназначенный для замены устаревшего BIOS. Несмотря на встроенные технологии безопасности, количество атак на UEFI с каждым годом растет, поскольку данный интерфейс представляет особый интерес для злоумышленников из-за его возможностей – запуск кода до загрузки операционной системы, хранение данных в NVRAM, позволяющее избегать средства защиты информации и другие особенности, которые помогают успешно проводить компьютерные атаки.

Прошивка UEFI располагается на SPI (Serial Peripheral Interface) микросхемах. Кроме прошивки, на них располагается хранилище NVRAM (энергонезависимой памяти), предназначенное для постоянного хранения данных в UEFI-совместимых прошивках. Обычно, для переменных выделяют область памяти из данной микросхемы – в целях сокращения расходов на дополнительные компоненты [1].

Микросхемы построены на технологии NOR Flash, средний показатель записи/стирания и повторной записи которых составляет 100000 циклов – по данным производителей этого хватает на 10 лет активного использования без сбоев [2]. Однако, высокая температура (например, на производствах), а также частая работа с переменными могут уменьшить ее ресурс, что может привести к серьезным последствиям, например, нарушению технологических процессов на производстве.

В связи с особенностями построения SPI микросхем и условий на производствах, авторами были выдвинуты две гипотезы, связанные с устойчивостью UEFI-совместимых программно-аппаратных комплексов:

* заполняя NVRAM, существенно увеличится время загрузки операционной системы;
* частое использование NVRAM может привести к раннему выходу из строя микросхемы (или вызвать нестабильную работу системы).

Для проверки вышеперечисленных гипотез авторами использовался следующий программный алгоритм:

1. Получение привилегии *SeSystemEnvironmentPrivilege* для взаимодействия с NVRAM переменными при помощи вызова WinAPI функции *RtlAdjustPrivilege;*
2. Проверка совместимости компьютера с UEFI при помощи вызова функции *GetFirmwareEnvironmentVariableW* [3]с аргументом {00000000-0000-0000-0000-000000000000};
3. Генерация уникального идентификатора (GUID);
4. Генерация строки со случайными данными длиной 30 символов;
5. Выполнение записи данных в хранилище переменных при помощи функции *SetFirmwareEnvironmentVariableW* [4] до получения кода ошибки о невозможности записи (1470) [5];
6. Проверка работоспособности и скорости загрузки компьютера.

Характеристики стенда для проверки гипотез:

* UEFI-совместимый компьютер;
* Windows 10 (21H2);
* Kaspersky Total Security 21.3.10.391(h).

На основе алгоритма авторами был создан PoC (Proof-of-Concept) для заполнения случайными данными энергонезависимой памяти. При тестировании на стенде были получены следующие результаты:

1. В ходе первого запуска реализации алгоритма система на этапе загрузки выдала предупреждение о практически полном заполнении хранилища переменных, требовала действия пользователя для продолжения работы (рис.1):



Рис.1. Предупреждение о заполнении хранилища переменных

Скорость загрузки компьютера ухудшилась в 2,7 раза. Несмотря на то, что разница между показателями 11,7 секунд, значение будет значительно выше на серверных компьютерах, так как объем хранилища переменных на них гораздо больше, чем на домашних компьютерах (рис.2 и рис.3):

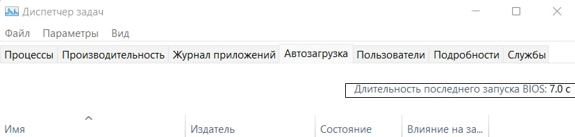


Рис.2. Время запуска компьютера до запуска PoC

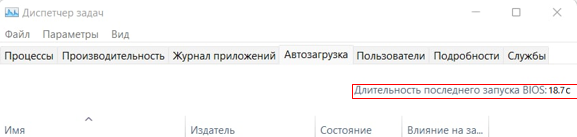


Рис.3. Время запуска компьютера после заполнения данными

1. После второго запуска система затребовала принудительной очистки NVRAM:



Рис.4. Сообщение о требуемой очистки NVRAM

Анализ полученных результатов показал, что заполнение хранилища переменных действительно влияет на скорость загрузки и работы программно-аппаратного комплекса. В ходе тестирования возникали проблемы с функционированием ПАК (черный экран) – требовалась принудительная перезагрузка компьютера. Однако, вывести из строя микросхему SPI не удалось – вторая гипотеза лишь частично подтвердилась. Кроме того, средство антивирусной защиты не обнаружило в проводимых действиях угрозы.

В результате исследования NVRAM переменных в UEFI-совместимых программно-аппаратных комплексов, авторами были выдвинуты следующие предложения по улучшению существующих средств защиты:

* в средствах антивирусной защиты должны быть реализованы перехватчики функций для взаимодействия с переменными в энергонезависимой памяти, где располагает прошивка UEFI;
* необходима проверка цифровой подписи программ, взаимодействующих с NVRAM переменными.

Кроме того, системным администраторам рекомендуется отключить привилегию *SeSystemEnvironmentPrivilege* через групповые политики для тех пользователей, у которых не выполняется ПО, взаимодействующее с NVRAM переменными. При необходимости обновления встроенного программного обеспечения (прошивки) предоставлять временный доступ.

**Литература**

1. Устройство NVRAM в UEFI-совместимых прошивках, часть первая [Электронный ресурс]. Режим доступа: https://habr.com/ru/post/281242/ (дата обращения 24.04.2022)
2. NOR FLASH: A PRACTICAL GUIDE TO ENDURANCE AND DATA RETENTION [Электронный ресурс]. Режим доступа: <https://www.infineon.com/dgdl/Infineon-AN99121_Practical_Guide_to_Endurance_and_Data_Retention-ApplicationNotes-v05_00-EN.pdf?fileId=8ac78c8c7cdc391c017d07425561655b&utm_source=cypress&utm_medium=referral&utm_campaign=202110_globe_en_all_integration-application_note> (дата обращения 24.04.2022)
3. GetFirmwareEnvironmentVariableW function (winbase.h) [Электронный ресурс]. Режим доступа:https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-getfirmwareenvironmentvariablew (дата обращения 24.04.2022)
4. SetFirmwareEnvironmentVariableW function (winbase.h) [Электронный ресурс]. Режим доступа:https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-setfirmwareenvironmentvariablew (дата обращения 24.04.2022)
5. Коды системных ошибок (1300-1699) [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/windows/win32/debug/system-error-codes--1300-1699-> (дата обращения 24.04.2022)