

Безопасность UEFI через призму механизмов защиты ViPNet SafeBoot

Михаил Лукин
Системный архитектор

БОЛЬШОЙ
МОСКОВСКИЙ
ТЕХНО  ФЕСТ

Безопасность UEFI: проблема

Развитие атак на firmware

- UEFI - набор стандартов по унификации механизмов инициализации платформ
- UEFI firmware – стандарт де-факто для платформ архитектуры x64
- Стандартизация firmware снижает порог входа в разработку компонентов уровня firmware
- Механизмы безопасности UEFI firmware играют ключевую роль в безопасности всей системы



UNIFIED EXTENSIBLE
FIRMWARE INTERFACE

Следствием является
развитие атак на
UEFI firmware

От экстравагантности к обыденности

- Ряд стандартных решений безопасности (UEFI, IHV, IBV, ODM, OEM, OS): Secure Boot, SPI Flash Protection, Intel Boot Guard, HP Sure Start, ...
- Несмотря на это - множество уязвимостей, malware
- Исследования по безопасности UEFI firmware: Kaspersky, ESET, PT, BINARLY, Eclypsium, ...



Причины внимания атакующих к уязвимостям UEFI firmware

- Ранний старт компонентов malware для обхода/отключения механизмов защиты ОС
 - Windows: PatchGuard, DSE
 - Linux: kernel lockdown, IMA
- Доступ к привилегированным режимам выполнения кода высшего приоритета
- Возможность максимального закрепления в системе
- Бóльшая невидимость от средств защиты уровня ОС



Основные вектора атаки на ранние механизмы защиты

- Перезапись firmware: встраивание malware, DoS
- Чтение firmware и конфигурации firmware
- Запись в NVRAM: регистрация старта внешних компонентов UEFI, ключи Secure Boot, проблемные переменные, переполнение
- Эксплуатация неактуальной конфигурации Secure Boot

- Эксплуатация механизмов обновления firmware без верификации
- Эксплуатация механизмов firmware по доступу к файлам на диске
- Модификация OpROM подключенных PCI-устройств
- Подмена/модификация загрузчика ОС
- Подмена/модификация произвольных файлов ОС

Проблемы стандартных средств защиты UEFI firmware

- Разделение на независимые решения (программные, аппаратные)
- Сложность настройки/согласования независимых решений
- Отсутствие комплексной защиты firmware и данных firmware в NVRAM
- Уязвимости Secure Boot и его расширений
- Функциональные ограничения Secure Boot и его расширений
- Отсутствие доверия к алгоритмам КЦ
- Незащищенные механизмы обновления firmware



Рассмотрим ситуацию с уязвимостями и malware на конкретных примерах

Основные уязвимости (CVE) #1

Категория: манипуляции с внешней конфигурацией

CVE-2020-10713
(BootHole)

переполнение стека
при обработке
grub.cfg в grub2

обход Secure Boot

CVE-2022-2601
CVE-2022-3775

переполнение кучи
при обработке
шрифтов в grub2

обход Secure Boot

CVE-2023-5058
CVE-2023-39539
CVE-2023-40238
(LogoFail)

переполнение кучи или
стека при обработке
изображений различных
форматов из файлов или
firmware

обход Secure Boot

Основные уязвимости (CVE) #2

Категория: манипуляции с подписью загрузчиков ОС

CVE-2022-34301
CVE-2022-34302
CVE-2022-34303

подпись дефолтным
ключом Secure Boot
уязвимых загрузчиков
ОС

обход Secure Boot

CVE-2024-7344

подпись дефолтным
ключом Secure Boot
уязвимых загрузчиков
ОС (с преднамеренным
обходом Secure Boot)

обход Secure Boot

CVE-2024-8105
(PKFail)

использование тестовых
ключей Secure Boot на
реальных платформах

обход Secure Boot*

Основные уязвимости (CVE) #3

Категории: манипуляции с NVRAM и протоколами загрузки

CVE-2025-3052

запуск подписанной
дефолтным ключом
Secure Boot уязвимой
утилиты UEFI
и манипуляции
с переменными NVRAM

обход Secure Boot

**CVE-2025-4275
(Hydrophobia)**

манипуляции
с переменными NVRAM
в сценарии
обновления firmware

обход Secure Boot

CVE-2023-40547

удаленное выполнение
кода в сетевой части
загрузчика shim для
получения загрузчика
следующей стадии (HTTP)

обход Secure Boot

Основные malware #1

Категория: буткиты уровня ФС

**ESpecter
(2021)**

модификация
bootmgfw.efi,
отключение
PatchGuard и DSE

старт кода уровня
ядра ОС

**Glupteba
(2023)
EfiGuard
(2019-2023)**

модификация
bootmgfw.efi,
перехват сервисов
таблиц BS/RT,
отключение
PatchGuard и DSE

старт кода уровня
ядра ОС

**BlackLotus
(2022/2023)
CVE-2022-21894**

откат на уязвимый
bootmgfw.efi,
изменение MOKList,
старт подмененного
grubx64.efi

старт кода уровня
ядра ОС

**Bootkitty
(2024)**

эксплуатация LogoFAIL*,
изменение MOKList,
старт подмененного
grubx64.efi

старт кода уровня
ядра ОС

Основные malware #2

Категория: импланты уровня firmware

**VectorEDK
(2015)
MosaicRegressor
(2020)**

внедрение DXE-модулей,
создание файлов на
системном разделе
Windows

старт кода уровня ОС

**LoJax
(2018)
CVE-2017-3197
CVE-2017-3198**

внедрение DXE-модуля,
создание файлов на
системном разделе
Windows

старт кода уровня ОС

**MoonBounce
(2022)**

патч DxeCore, перехват
сервисов BS/RT,
отключение Windows
PatchGuard и DSE

старт кода уровня
ядра ОС

**CosmicStrand
(2022/2023)**

патч CsmDxe, перехват
сервисов BS/RT,
перехват функций ядра
Windows

старт кода уровня
ядра ОС

Сомнительное ПО уровня UEFI

Computrace / LoJack
Lenovo Service Engine
ASUS Armoury Crate

...

использование ACPI WPBT для записи
исполняемых файлов в системный
раздел Windows для последующего
автостарта

**использование данных механизмов
в сложных атаках через подмену
исходных файлов на компоненты
malware**

Жизненный цикл уязвимости

Сценарий 1

- Нахождение уязвимости и подтверждение ее через PoC

Сценарий 2

Исследователь / white hat

- Сигнализация вендору (INHV/IBV/ODM/OEM/OS) об уязвимости
- Разбор уязвимости вендором
- Разработка и тестирование исправления уязвимости вендором
- Выпуск обновления с исправлением уязвимости
- Оглашение информации о закрытии уязвимости (вендором) и описание самой уязвимости (исследователем)
- Распространение обновления на целевые системы

Атакующий исследователь / black hat

- Написание malware с эксплуатацией найденной уязвимости
- Распространение malware на целевые системы

Как с этим бороться (с т.з. конечных “пользователей”)?

Стандартные средства защиты

- Максимально оперативная установка обновлений
- Надежда на то, что все стандартные средства защиты доступны и настроены корректно (в связке)
- Надежда на то, что окно для сценария 1 будет минимальным и сценарий 2 не будет задействован

Сторонние средства защиты

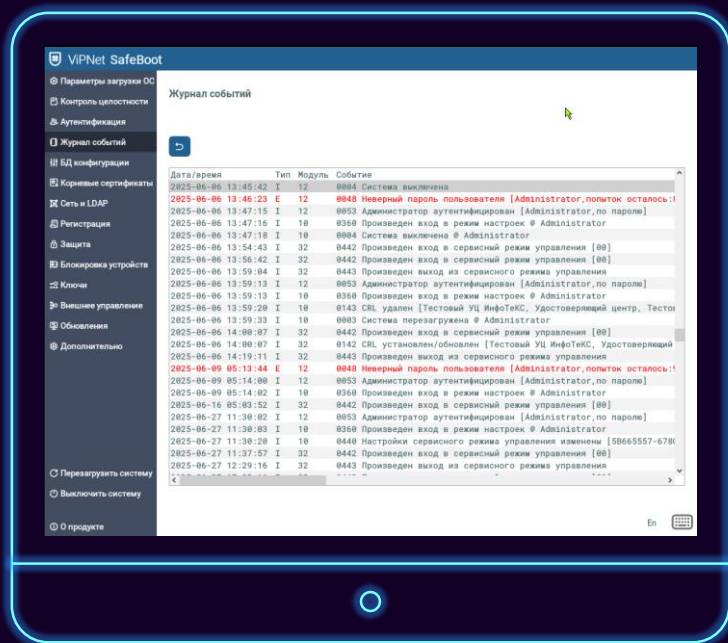
- Комплексное, централизованное и единообразное управление механизмами защиты
- Проактивное закрытие многих классов уязвимостей и векторов атак
- Встроенные базовые средства по разбору и решению инцидентов безопасности

Безопасность UEFI: решение

МДЗ ViPNet SafeBoot



- Программное решение уровня UEFI firmware
- Унифицированная поддержка множества платформ и ОС
- Гибкие механизмы встраивания в firmware множества платформ (x64, aarch64)
- Независимые и автономные механизмы КЦ и защиты
- Функциональные механизмы управления
- Сертифицированное решение по требованиям ФСТЭК России и ФСБ России



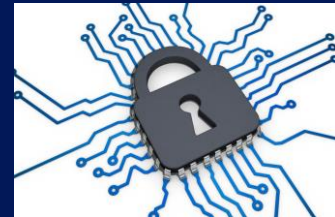
Механизмы загрузки доверенной ОС

- Передача управления на заданный загрузчик ОС
- Предварительный КЦ программной среды:
 - произвольных файлов: исполняемых (драйверов, сервисов, приложений, библиотек, ...), конфигурационных, скриптов
 - каталогов файлов
 - реестра Windows (на уровне ключей/значений)
 - КЦ секторов загрузочной области диска
- Механизмы КЦ основаны на независимой схеме на базе алгоритмов семейства ГОСТ
- Изменение настроек доступно только аутентифицированному администратору МДЗ



Основные механизмы защиты

- Защита SPI Flash от записи и чтения
- Фильтрация SW SMI
- Эмуляция NVRAM
- Блокировка встроенных механизмов обновления firmware
- Блокировка входа в режимы конфигурирования firmware
- Блокировка исполнения PCI OpROM
- Фильтрация передачи управления на сторонний код
- Виртуализация системных таблиц UEFI
- Блокировка ACPI WPBT
- Блокировка записи на диск и чтения с диска
- Маскирование системных модулей firmware



Как это соотносится
с противодействием
конкретным уязвимостям
и malware?

Основные уязвимости (CVE) #1

Категория: манипуляции с внешней конфигурацией

CVE-2020-10713
(BootHole)



переполнение стека
при обработке
grub.cfg в grub2

~~обход Secure Boot~~

КЦ файлов

CVE-2022-2601
CVE-2022-3775



переполнение кучи
при обработке
шрифтов в grub2

~~обход Secure Boot~~

КЦ файлов

CVE-2023-5058
CVE-2023-39539
CVE-2023-40238
(LogoFail)



переполнение кучи или
стека при обработке
изображений различных
форматов из файлов или
firmware

~~обход Secure Boot~~

блокировка чтения
с диска, защита SPI
Flash, фильтрация SW
SMI, КЦ файлов

Основные уязвимости (CVE) #2

Категория: манипуляции с подписью загрузчиков ОС

CVE-2022-34301
CVE-2022-34302
CVE-2022-34303



~~подпись дефолтным
ключом Secure Boot
уязвимых загрузчиков
ОС~~

~~обход Secure Boot~~

КЦ файлов

CVE-2024-7344



~~подпись дефолтным
ключом Secure Boot
уязвимых загрузчиков
ОС (с преднамеренным
обходом Secure Boot)~~

~~обход Secure Boot~~

КЦ файлов

CVE-2024-8105
(PKFail)



~~использование тестовых
ключей Secure Boot на
реальных платформах~~

~~обход Secure Boot*~~

КЦ файлов

Основные уязвимости (CVE) #3

Категории: манипуляции с NVRAM и протоколами загрузки

CVE-2025-3052



запуск подписанной
дефолтным ключом
Secure Boot уязвимой
утилиты UEFI
и манипуляции
с переменными NVRAM

обход Secure Boot

КЦ файлов, защита SPI
Flash, фильтрация SW
SMI, эмуляция NVRAM

CVE-2025-4275
(Hydrophobia)



манипуляции
с переменными NVRAM
в сценарии
обновления firmware

обход Secure Boot

защита SPI Flash,
фильтрация SW SMI,
эмуляция NVRAM

CVE-2023-40547



удаленное выполнение
кода в сетевой части
загрузчика shim для
получения загрузчика
следующей стадии (HTTP)

обход Secure Boot

КЦ файлов*

Основные malware #1

Категория: буткиты уровня ФС

ESpecter
(2021)



модификация
bootmgfw.efi,
отключение PatchGuard
и DSE

старт кода уровня
ядра ОС

КЦ файлов

Glupteba
(2023)
EfiGuard
(2019-2023)



модификация
bootmgfw.efi,
перехват сервисов
таблиц BS/RT,
отключение
PatchGuard и DSE

старт кода уровня
ядра ОС

КЦ файлов

BlackLotus
(2022/2023)
CVE-2022-21894



откат на уязвимый
bootmgfw.efi, изменение
MOKList, старт
подмененного
grubx64.efi

старт кода уровня
ядра ОС

КЦ файлов

Bootkitty
(2024)



эксплуатация LogoFAIL*,
изменение MOKList,
старт подмененного
grubx64.efi

старт кода уровня
ядра ОС

КЦ файлов, защита SPI
Flash, фильтрация SW
SMI, эмуляция NVRAM

Основные malware #2

Категория: импланты уровня firmware

VectorEDK
(2015)
MosaicRegressor
(2020)



внедрение DXE-модулей,
создание файлов на
системном разделе
Windows

старт кода уровня ОС

защита SPI Flash,
фильтрация SW SMI,
эмуляция NVRAM

LoJax
(2018)
CVE-2017-3197
CVE-2017-3198



внедрение DXE-модуля,
создание файлов на
системном разделе
Windows

старт кода уровня ОС

защита SPI Flash,
фильтрация SW SMI,
эмуляция NVRAM

MoonBounce
(2022)



патч DxeCore, перехват
сервисов BS/RT,
отключение Windows
PatchGuard и DSE

старт кода уровня
ядра ОС

защита SPI Flash,
фильтрация SW SMI,
эмуляция NVRAM

CosmicStrand
(2022/2023)



патч CsmDxe, перехват
сервисов BS/RT,
перехват функций ядра
Windows

старт кода уровня
ядра ОС

защита SPI Flash,
фильтрация SW SMI,
эмуляция NVRAM

Сомнительное ПО уровня UEFI

Computrace / LoJack
Lenovo Service Engine
ASUS Armoury Crate

...



~~использование ACPI WPBT для записи
исполняемых файлов в системный
раздел Windows для последующего
автостарта~~

~~использование данных механизмов
в сложных атаках через подмену
исходных файлов на компоненты
malware~~

блокировка ACPI WPBT,
блокировка записи на диск

Каким же образом
происходит встраивание
имплантов в firmware?

Встраивание в firmware #1: практический пример

Аппаратная платформа:

- Ноутбук от зарубежного OEM первой категории на базе Intel CPU (двухлетней давности)

Активированные механизмы защиты:

- Intel Boot Guard: аппаратная часть с фиксацией конфигурации в fuses/FPFs (Production Mode)
- Intel Boot Guard: программная часть
- Регистры PRx: инициализированы с блокировкой
- Регистр BIOS_CNTL: инициализирован с блокировкой



Казалось бы –
все идеально?

Встраивание в firmware #2: практический пример

Проблема: механизмы защиты настроены некорректно:

- Intel Boot Guard в программной части не закрывает все исполняемые firmware volume
- Регистры PRx не закрывают все исполняемые firmware volume
- Защиту на базе регистра BIOS_CNTL можно отключить через NVRAM

Результат:

- Ноутбук **не защищен** от изменения firmware даже программным способом - например, средствами flashrom или fpt с уровня ОС



Встраивание в firmware #3: противодействие

Использование ViPNet SafeBoot позволяет “из коробки” закрыть эту проблему собственными механизмами защиты:

- Защита SPI Flash: полное закрытие BIOS region (включая NVRAM)
- Фильтрация SW SMI: блокировка использования ПО обновления firmware на базе SMM-протоколов
- Аутентификация пользователя в МДЗ: блокировка возможности изменения конфигурации защиты
- Блокировка входа в режимы конфигурирования firmware



Обеспечивается
совместимость
с Intel Boot Guard



Бонус: блокируется
функциональность
активированного
Computrace



**Подведем итог:
стандартных средств
защиты недостаточно**

Наш девиз: постоянное движение вперед!

- Анализ актуальных исследований по безопасности firmware и систематизация полученной информации
- Проведение собственных исследований существующих механизмов защиты и определение их достаточности
- Оперативная поддержка новых механизмов защиты в очередных релизах
- Взаимодействие с несколькими регуляторами, испытательными лабораториями, ИСП РАН



ТЕХНОinfotecs Фест

Спасибо за внимание!
Готов ответить на ваши вопросы



NBJ НАЦИОНАЛЬНЫЙ
БАНКОВСКИЙ ЖУРНАЛ



РУТОКЕН
КОММЕРЧЕСКАЯ ПРАКТИКА



ПЕРСПЕКТИВНЫЙ
МОНИТОРИНГ

АВАНГАРД

ARinteg

АТБ
ЭЛЕКТРОНИКА

infotecs
{/cademy}

m≈rlion



Подписывайтесь
на наши соцсети,
там много интересного

