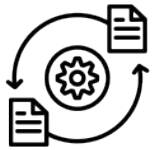
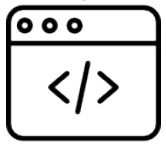
ОСНОВНЫЕ СРЕДСТВА ЗАЩИТЫ ПО ОТ ВЗЛОМА

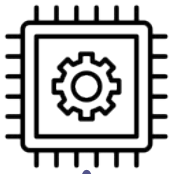


Организационные средства защиты:
1. Политика предоставления доступа к ПО.
2. Техническая поддержка клиентов.
3. Персонализация лицензий.

Правовые средства защиты:
1. Патентование технических решений.
2. Договоры и лицензионные соглашения.
3. Борьба с сайтами, распр. взломанное ПО



Программные средства защиты:
1. Шифрование программного кода и файлов.
2. Активация и управление лицензиями.
3. Противодействие реверсингу ПО.



Программно-аппаратные средства защиты:
1. Генерация HWID.
2. Использование USB-ключей.
2. Физическое разделение программного обеспечения и оборудования.

СТАТИЧЕСКАЯ И ДИНАМИЧЕСКАЯ ИДЕНТИФИКАЦИЯ

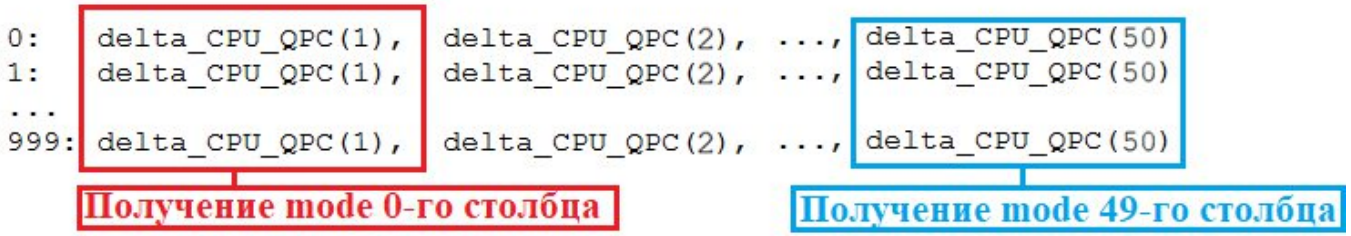
Под статической идентификацией понимается получение HWID ПК на основе различных аппаратных или программных идентификаторов.
Так же данный подход именуется атрибутивным
(получение различных атрибутов /свойств установленного
аппаратного или программного обеспечения).

Динамическая идентификация аппаратной части ПК - измерение нескольких уникальных параметров, совокупность которых может однозначно идентифицировать множество
ПК с одинаковыми аппаратным и программным обеспечением.
Такую генерацию HWID сложнее обнаружить и обойти,
т.к. отсутствует явное использование какой-либо статической характеристики.

ДИНАМИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ВЕРСИЯ №1

Была произведена попытка реализации динамической идентификации аппаратной части ПК **на основе матрицы скорости выполнения кода.**

Формирование матрицы скорости выполнения CryptGenRandom()



Расчёт mode матрицы

Основные шаги реализации:

1. Выбрать функцию для измерения
2. Выбрать аппаратные таймеры
3. Сгенерировать матрицы, рассчитать mode каждой
4. Сравнить mode двух матриц
5. Матрицы равны при совпадении более 50%

ДИНАМИЧЕСКАЯ ИДЕНТИФИКАЦИЯ
УТОЧНЕНИЕ ДЕТАЛЕЙ

После неудачной первой версии была предпринята попытка связаться с авторами научных статей в данной сфере чтобы выяснить ряд вопросов. Получены следующие ответы:

1. Использовать стандартные Windows API.
2. Не работать с GPU.
3. Использовать два таймера с различной частотой.
4. Чем точнее будут оба таймера, тем лучше.
5. Данные о характеристиках ПК с тестами утеряны.
6. Матрица под каждый таймер, а затем вычитание по модулю.

Автор статьи не помнит ключевых деталей, отсутствующих в статье.

В ОС Windows доступно несколько различных таймеров для использования:

1. RTC с частотой 32.768 КГц.
2. PIT с частотой 1.193180 МГц.
3. ACPI с частотой 3.58 МГц.
4. HPET с частотой от 10 до 24 МГц.
5. LAPIC без фиксированной частоты.

ДИНАМИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ВЕРСИЯ №2

С учётом ответов авторов научных статей было принято решение изучить и протестировать способ измерения скорости выполнения кода на основе тиков процессора как наиболее точный вариант измерений – для первой матрицы.
WinAPI QueryPerfomanceCounter для второй матрицы.

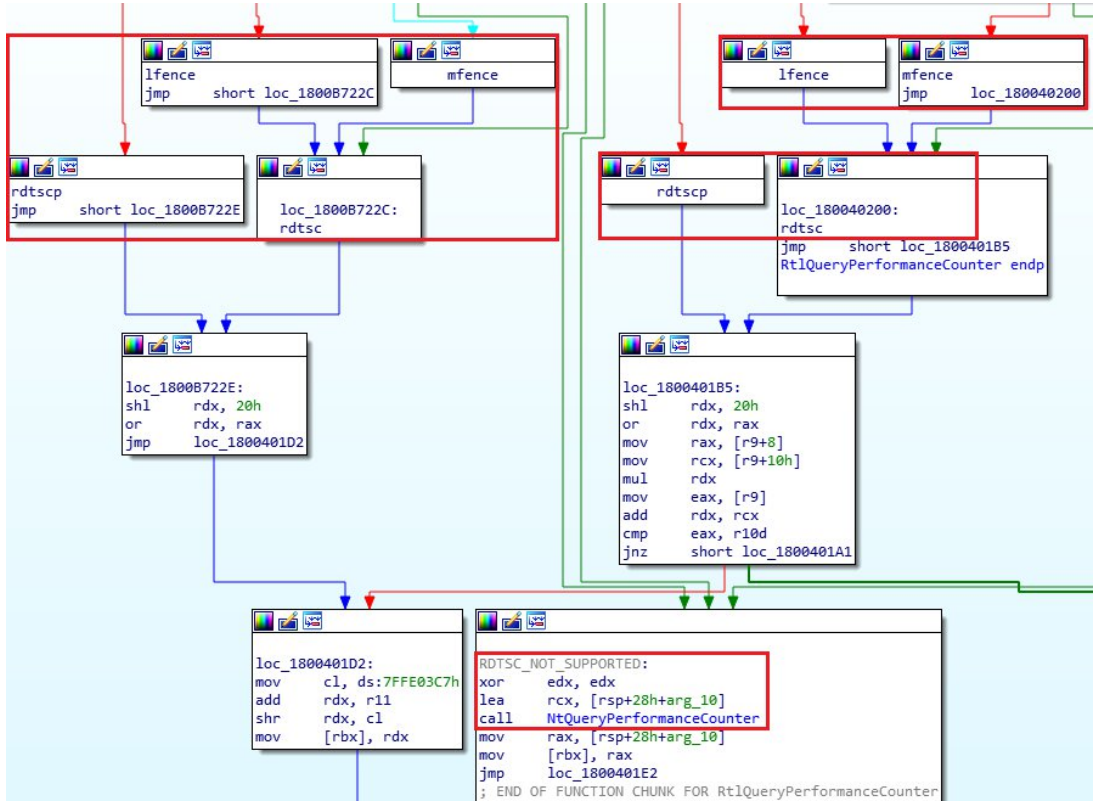
```
LFENCE()
StartTicks = RDTSC()

;...измеряемый код...

LFENCE()
EndTicks = RDTSC()

DiffTicks = EndTicks - StartTicks
```

Замер тиков процессора



Asm листинг функции ntdll.RtlQueryPerformanceCounter

ДИНАМИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ВЕРСИЯ №2. РЕЗУЛЬТАТЫ

Тесты на одном ПК показали 100% совпадение дельта-матриц в 10 случаев из 10 без нагрузки на CPU. Однако, матрицы, сгенерированные без нагрузки на CPU и с нагрузкой, считаются двумя мало похожими матрицами (процент идентичности варьируется около 70%).

Сравнение полученной и ожидаемой стабильности дельта-матриц

Рекомендации по дальнейшему исследованию:

1. Тесты на нескольких ПК с одинаковыми аппаратными компонентами.
2. Работать с таймерами HPET и ACPI из ядра.
3. Стабилизировать частоту ОЗУ.
4. Применить STM.
5. Тесты на разных моделях CPU Intel & AMD.

ИТОГИ

1. Не удалось реализовать нативную версию генерации HWID на основе замеров матриц скорости выполнения кода в 2025 году на старом и новом железе.
2. В интернете нет готовой реализации (есть 2 проекта-попытки на 100 строк кода на C и JS, авторы которых не пытались копнуть глубже, а просто взяли первую функцию измерения времени и попытались в лоб решить задачу)
3. Автор статьи (Iskander Sanchez-Rola) говорит что «это было так давно, я уже всё забыл. Исходников нету»
4. ИИ так же решает задачу в лоб (неверно), не отвечает ничего путного.

В итоге, реально ли это? Думаю да, однако в статье явно скрыты ключевые детали для реализации.
Возможно это было сделано намеренно и прямо сейчас работающая реализация крутится у кого-то на проде для детекта ботов, подсчёта уников и тд.
Необходим ещё более глубокий ресерч (у меня на это исследование в формате ВКР ушло несколько месяцев работы после основной работы).