Реверс-инжиниринг читов для Counter-Strike 2 и обход пейволла

Теоретическая часть

Реверс-инжиниринг (обратный инжиниринг) — это процесс исследования готового изделия или программы для получения информации о его конструкции, принципах работы и технологии производства без доступа к исходной документации, чтобы воссоздать его, создать аналог, улучшить или понять его функции.

IDA Free - это бесплатная версия мощного интерактивного дизассемблера Interactive Disassembler, разработанного компанией Hex-Rays. Это инструмент для реверс-инжиниринга, который позволяет анализировать исполняемые файлы без доступа к их исходному коду.

Основные возможности IDA Free:

- Дизассемблирование x86/x64 кода
- Графовый и текстовый режимы просмотра
- Поиск строк и функций
- Базовый анализ потока управления
- Экспорт ассемблерного листинга

Патчинг (модификация бинарного кода) — процесс модификации исполняемого файла путём изменения его бинарного кода. В контексте реверс-инжиниринга патчинг используется для:

- Обхода систем защиты и лицензирования
- Исправления ошибок в программах
- Модификации функциональности
- Удаления или изменения нежелательных элементов

Практическая часть

1) Анализ exe

Был получен EXE-файл читов с пэйволлом (защитой по ключу). С помощью IDA Free проведён анализ для поиска функций аутентификации.

Была найдена функция - Overlay::DrawAuth(void). При анализе её графа была обнаружена логика проверки аутентификации. В коде идентифицирован блок, в котором параметр _refptr_g_IsAuthenticated устанавливается в истинное значение при успешной проверке:

```
mov     rdx, cs:_refptr_g_IsAuthenticated
mov     eax, 1
xchg    al, [rdx]
mov     ecx, 1
mov     cs:_ZL10isMoveMode, 0 ; isMoveMode
call    _ZN7Overlay19SetInputPassThroughEb ; Overlay::SetInputPassThrough(bool)
jmp     loc_140003B03
```

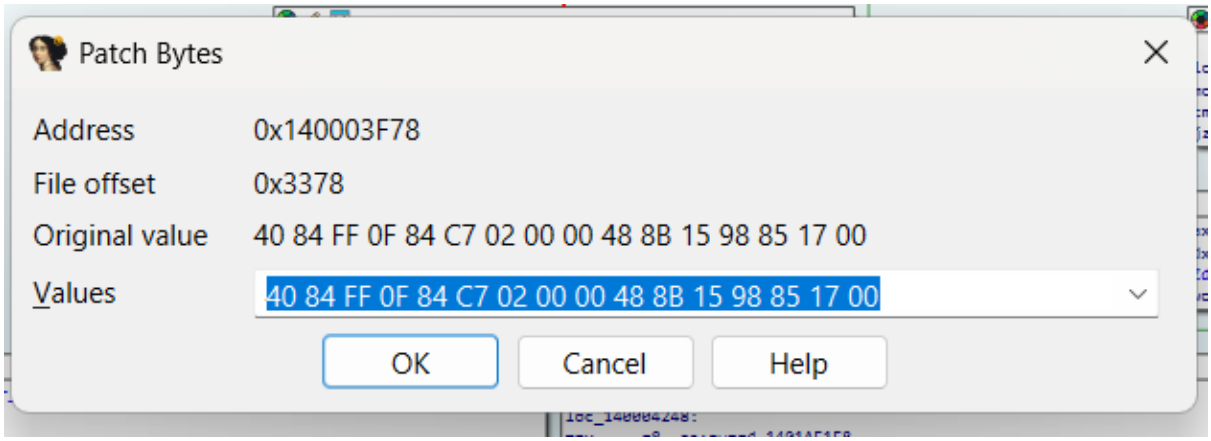
После этого был обнаружен критический участок кода, который при успехе выполняет блок с установкой флага аутентификации, при неудаче — выводит сообщение "Banned or Invalid."

2) Разработка и применение патча

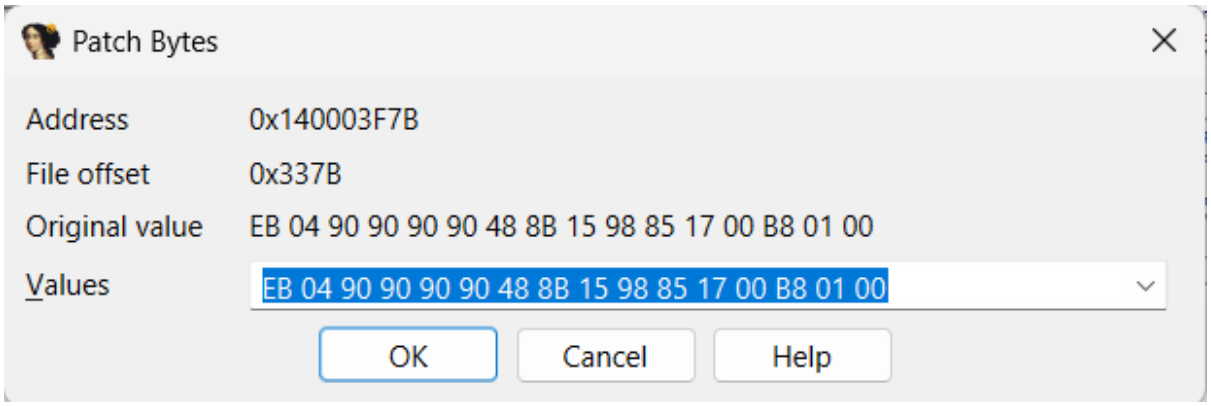
Мы заметили, что в функции есть переход - jz loc_140004248. Это условный переход, который при dil = 1 подтверждает аутентификацию, а при dil = 0 - выводит ошибку. Было принято решение заменить его на безусловный jmp, что обеспечивает постоянное выполнение пути успешной аутентификации независимо от проверки ключа.

Для реализации этого мы выполнили патч, то есть поменяли байты у адреса loc_140003F78. Мы выделили этот блок и применили

следующие переходы в меню у IDA: Edit -> Patch program -> Change byte:



Поменяли значение на:



После этого применили наши изменения: Edit -> Patch program -> Apply patches и протестировали полученный экзешник. Теперь при вводе любого ключа мы получаем доступ ко всем читам: