# **PWN-11: Портал в Край**

В задании предоставляется консольное приложение с исходным кодом на языке C, скомпилированное под архитектуру linux amd64. При запуске приложения пользователю выводится меню с несколькими опциями.

## **Флаг 1:**

Из исходного кода можно было заключить, что в меню есть скрытая опция, которую можно активировать введя символ - ` (обратный апостроф), после чего пользователю предлагается выполнить арифметическое сложение двух случайных шестнадцатеричных чисел, если введенные пользователем результат окажется верным, на экран выведется первый флаг.

Изображение выглядит как текст, диаграмма, снимок экрана, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.

## **Флаг 2:**

В опции меню [T] Teleport to coordinates допущено 2 ошибки:

* В первый аргумент функции printf передается пользовательский ввод – стока 101
* Переполнение буфера на стеке из-за использования небезопасной функции gets – строка 99

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Программа скомпилирована с защитными техниками: PIE (исполняемый модуль с позиционно-независимым кодом) и canary (стековая канарейка). Используя первую уязвимость – форматной строки, можно получить утечку значения канарейки, а также адреса из основной секции бинарного файла. Что позволит вычислить адрес функци win, которая запускает интерпретатор sh, в случае если ей передан правильный пароль, и записать его на стеке на место адреса возврата, не повредив значение канарейки, используя вторую уязвимость – переполнение буфера.

Для того, чтобы обойти проверку пароля в этой функции, достаточно взять небольшое смещение от начала функции win, чтобы пропустить инструкции отвечающие за проверку пароля.

Скрипт с решением на языке python с использованием библиотеки pwntools:

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.