# **PWN-10: 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:**

Участникам был выдан файл run.sh, демонстрирующий как приложение запускается на сервере, из него можно заключить, что флаг передается приложению в через переменные окружения, и при запуске в приложение передается аргумент командной строки, который был замаскирован.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

При инициализации приложения в функции main, используется проверка пароля переданного в первом аргументе командной строки argv[1] при старте приложения, преданное значение шифруется однобайтовым xor ключем 0x15 и сравнивается с константным значением.

В этой функции допущена ошибка, несмотря на то, что пароль состоит из 15 байт, из адреса argv[1]копируется больше байт в стековый буфер. В отладчике необходимо было заметить, что после первого аргумента по адресу argv[1] располагаются переменные окружения. Которые попадут в буфер char password[0x40], и к ним также будет применена операция xor.

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

В опции меню [T] Teleport to coordinates используется функция printf, в первый аргумент которой передается пользовательский ввод, что позволяет использовать уязвимость форматной строки для получения переменных расположенных ниже по стеку, в том числе и буфер password содержащий зашифрованный флаг.

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Также важно обратить внимание, что программа была скомпилирована с флагом -D\_FORTIFY\_SOURCE=2, что не позволяло обращаться к аргументам printf используя порядковые номера. Поэтому для получения содержимого буфера password необходимо было вывести все содержимое стека.

После получения буфера password его необходимо расшифровать обратно.

Скрипт с решением на языке python с использованием библиотеки pwntools:

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.