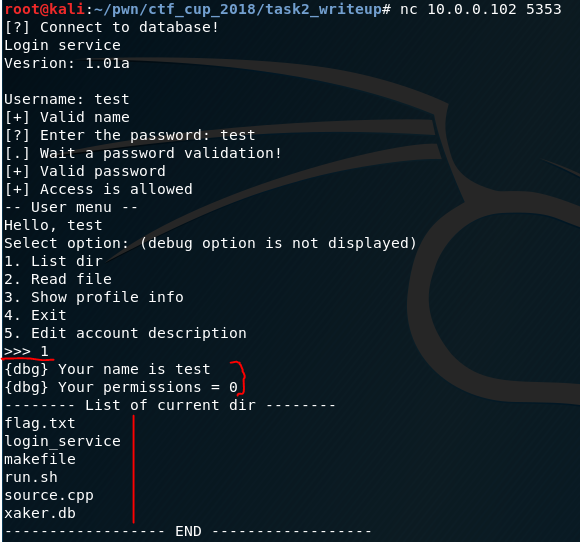
**Решение заданий категории Pwn.**

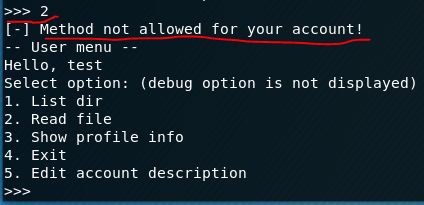
**Задание №2. «Login panel: Permissions»**

После получения командой доступа к аккаунту «test/test», необходимо проверить все доступные команды.

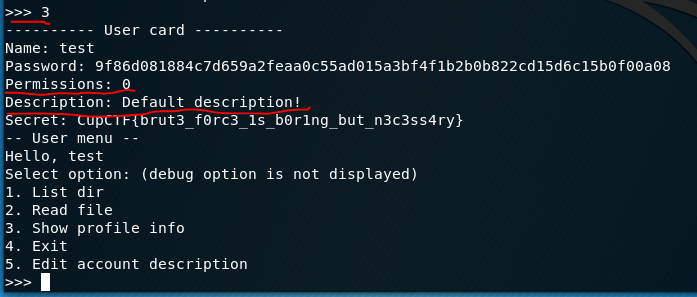


Первая команда – просмотр содержания текущей директории. В ходе её выполнения также отображается оставленная отладочная информация. Например, имя текущего пользователя и значение, которое отвечает за его привилегии.

Вторая команда – чтение файла. Для данного аккаунта команда недоступна.

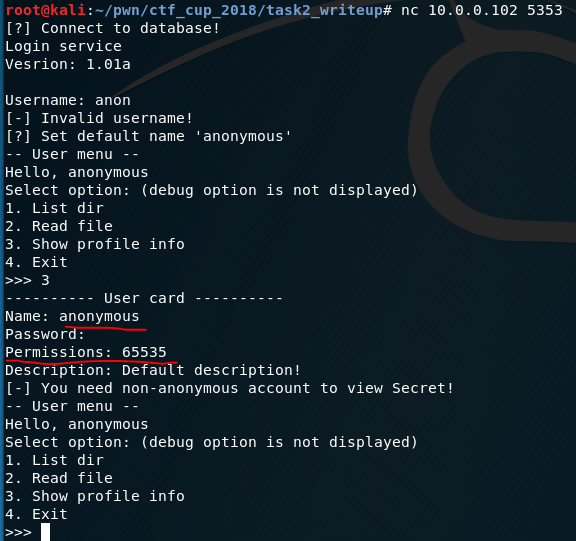


Третья команда – просмотр информации о профиле пользователя. Отображаемая информация содержит имя, хеш пароля, значение привелегий, описание профиля.



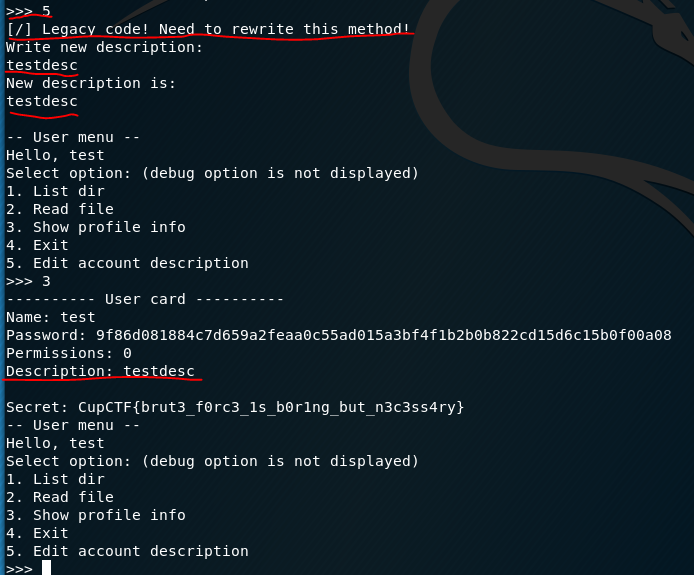
Здесь стоит обратить внимание на такие поля, как описание и привилегии. В отличии от аккаунта «anonymous» в аккаунте «test» существует ещё одна команда. Это команда редактирования информации об аккаунте, которая содержится в поле «Description» и по стандарту в нём находится строка «Default description!».

Поле «Permissions» содержит значение 0 для аккаунта «test» и значение 65535 для аккаунта «anonymous».



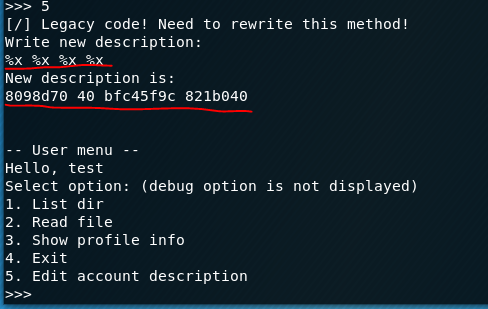
Попробуем использовать новую функцию, которая не была доступна в аккаунте «anonymous» - функция под номером 5.

Сразу после вызова данной функции появляется сообщение о том, что код данной функции наследуемый и нужно переписать данный метод, это оповещение может натолкнуть на мысль о том, что в данной функции меню есть уязвимость.

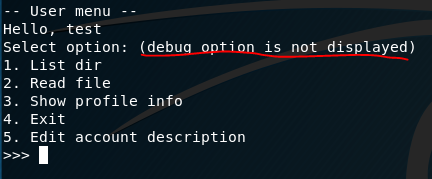


Вводим новое описание, после чего оно выводится на экран и изменяется в аккаунте.

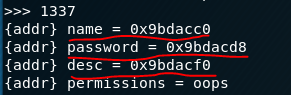
Попробуем передать форматную строку и вывод нового описания будет представлять собой значения со стека – это значит, что код данной функции подвержен уязвимости форматно строки.



Уязвимость найдена, однако без знания внутреннего адресного пространства эксплуатация будет затруднена. На данном этапе необходимо внимательно взглянуть на описание меню пользователя. В строке выбора опции в скобках сказано, что отладочные опции не отображены, это значит, что есть ещё некоторые опции, номера которых не отображены. Попробуем перебрать их вручную или с помощью скрипта.

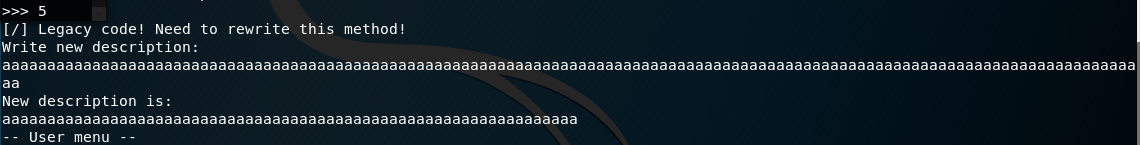


После недолгого перебора можно найти функцию под номером 1337, которая выведет информацию об адресах некоторых полей в информации профиля.



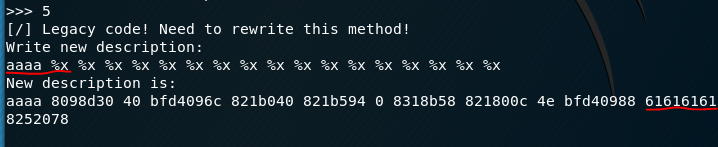
Судя по всему, это некоторые поля структуры или класса, так как они расположены друг за другом в памяти (но это не 100% утверждение).

Теперь у нас появляется адрес, по которому будет записано описание профиля, однако адреса, по которому хранится значение, отвечающее за привилегии – нет. Но, данное значение можно определить путём передачи буферов различной длины при вызове функции номер 5. Стоит отметить, что это утверждение верно, если для хранения описания не используется динамический массив или контейнер.



Попробуем передать 128-байтный буфер и заметим, что записывается только 64 байта, то есть к адресу хранения описания нужно прибавить 64 байта и будет получен адрес, по которому хранится значение привилегий.

После нахождения адреса, необходимо определить каким по счёту на стеке является наш аргумент при использовании уязвимости форматной строки. Это делается довольно просто.



Теперь осталось соединить это всё в финальный эксплоит, пример эксплоита, получающего файл с флагом представлен в файле «exploit\_task2\_get\_flag.py».

Пример работы представлен на изображении ниже. Можно заметить, что имя пользователя осталось «test», но поле привилегий поменяло своё значение, а также появилась новая функция под номером 6.

