Открываем программу в гидре, смотрим определенные строки

140011000	Username:	"Username: "	ds
14001100d	Password:	"Password: "	ds
14001101b	Good	"Good\n"	ds

Идем в xref-функцию. Видим явную инициализацию буфера

```
_Buf = user;
user[0] = '\0';
user[1] = '\0';
user[2] = '\0';
user[3] = '\0';
user[4] = '\0';
user[5] = '\0';
user[6] = '\0';
user[7] = '\0';
user[8] = '\0';
user[9] = '\0';
user[10] = '\0';
user[0xb] = '\0';
user[0xc] = '\0';
user[0xd] = '\0';
user[0xe] = '\0';
user[0xf] = '\0';
user[0x10] = '\0';
user[0x11] = '\0';
user[0x12] = '\0';
user[0x13] = '\0';
user[0x14] = '\0';
```

И потом в него считываются введенный логин

```
do {
  printf("Username: ");
  _File = (*p_Var1)(0);
  fgets(_Buf,100,(FILE *)_File);
  length = strcspn(_Buf,"\n");
  user[length] = '\0';
} while (user[0] == '\0');
```

Дальше ввод пароля типа unsigned char

```
printf("Password: ");
scanf("%u");
length = strlen(_Buf);
sum = 0;
buf_end = _Buf + length;
for (; _Buf != buf_end; _Buf = _Buf + 1) {
  sum = sum + *_Buf;
}
```

И самое интересно – цикл for. Перед ним мы получаем длину непустой части буфера, конец буфера и для каждого символа в буфере, так как это указатель на символ, то

при его разыменовании должен браться код этого символа. И код каждого символа суммируется в переменную sum

А дальше самое интересное

```
if (sum == 0) {
  printf("Good\n");
}
else {
  printf("Bad\n");
}
```

Сумма при вводе валидного логина не может быть равной нулю, так как буфер не пуст. Значит, проверка сама по себе некорректна. Значит, идём патчить условие перехода

Запоминаем адрес инструкции

```
14000f1fa 74 1a          JZ          LAB_14000f216
```

Идем в дебаггер и меняем 74 на 75 или JZ на JNZ

```
00007FF68376F1F   75 1A                jne ezreverse.7FF68376F216
00007FF68376F1F   48:8D0D 1E1E0000     lea rcx,qword ptr ds:[7FF683771021 ]
```

Сохраняем патч, проверяем

```
Username: user
Password: thisisnotapassword
Good
```

АПДЕЙТ

Ghidra ошиблась, а я не заметил

Почему-то C-подобный код заменил мне инструкцию scanf и условие проверки

Однако в на уровне ассемблерных инструкций сравнение имеет другой вид

```
14000f1f6 39 54 24 2c     CMP         dword ptr [RSP + pw],sum
```

То есть сумма сравнивается не с нулем, а с введенным паролем. И пароль должен совпадать с суммой аски-кодов логина

Проверим на непропатченной программе

```
\ezreverse.exe
Username: user
Password: 447
Good
```

Резюмирую:

Не цепляйся только за гидру. Если не уверен, посмотри ассемблерные инструкции – могла быть ошибка