3) Reverse

Viene dato un esercizio *BankAcc*, simile si presume all'esercizio omonimo di pagina 106. Viene infatti dato un solito file binario senza estensione *BankAcc.*

Il file binario consiste in un inserimento di uno username, che non viene in realtà trovato.
Usando il comando *strings*, si nota subito che in chiaro, l'utente da inserire è *"UniPD_Student"*.
Si nota dalla lista delle funzioni qualcosa di interessante:

- Una funzione *decrypt* che letteralmente copia una serie di byte e viene chiamato dall'esterno eseguendo un confronto sui propri registri; in particolare usa *canary* che, come sappiamo, fa emergere un nuovo indirizzo ad ogni esecuzione
- Il controllo da parte di *security_check* che chiama la funzione *ptrace* che conosciamo; potenzialmente patchabile e sovrascrivibile con delle nop. Provando a mettere una password molto lunga, viene trovato uno stack_smashing.
- Una funzione *create_otp*, con una chiamata alla funzione *time* e alla funzione *srand*; anche questa, per esperienza pregressa, utile da considerare e nel caso patchare
- Una funzione *checkPassword* il cui scopo è offuscare la password, eseguendo una serie di copie e paragoni in memoria. Seguendo attentamente l'ordine di dichiarazione delle variabili, si nota che queste conducono alla stringa "*gP01o3!v*"

Dando un esempio di come si vedono su Radare2 le variabili:

```
0x004010e0]> pdf @ sym.check_password
          ; CALL XREF from main @ 0x4014e0(x)
247: sym.check_password ();
          ; var int64_t canary @ rbp-0x8
          ; var int64_t var_19h @ rbp-0x19
          ; var int64_t var_1ah @ rbp-0x1a
          ; var int64_t var_1bh @ rbp-0x1b
          ; var int64_t var_1ch @ rbp-0x1c
          ; var int64_t var_1dh @ rbp-0x1d
          ; var int64_t var_1eh @ rbp-0x1e
          ; var int64_t var_1fh @ rbp-0x1f
          ; var char *s @ rbp-0x20
```

Ragionando sull'ordine di comparsa e di chiamata nella funzione, visibile da IDA possiamo comporre la stringa "*P10v3go!*", convertendo ASCII a testo.

```
  printf("\nInsert password:
  __isoc99_scanf("%s", s);
  if ( strlen(s) != 8 )
    return 0LL;
  if ( s[5] != 103 )
    return 0LL;
  if ( s[0] != 80 )
    return 0LL;
  if ( s[2] != 48 )
    return 0LL;
  if ( s[1] != 49 )
    return 0LL;
  if ( s[6] != 111 )
    return 0LL;
  if ( s[4] != 51 )
    return 0LL;
  if ( s[7] == 33 )
    return s[3] == 118;
  return 0LL;
}
```

UniPD_Student
P10v3go!

In seguito, viene richiesto una otp, esso è creato randomizzato con il tempo e viene effettuato il mod 9999, in modo da fare un sanitize del random, questo non è rompibile, dobbiamo cercare da altre parti.

```
  time_t timer; // [rsp+10h] [rbp-10h] BYREF
  unsigned __int64 v3; // [rsp+18h] [rbp-8h]

  v3 = __readfsqword(0x28u);
  v0 = time(&timer);
  srand(v0);
  return (unsigned int)(rand() % 9999);
}
```

Notiamo inoltre dal decompiler di IDA che la password viene vista come intera:

```
{
  puts("Username not found, exiting...");
  exit(0);
}
if ( !(unsigned int)check_password(s1) )
{
  puts("Incorrect password, exiting...");
  exit(0);
}
puts("\n============================\nWelcome Back, UniPD_Student!\n==========
OTP = create_OTP("\n============================\nWelcome Back, UniPD_Student
printf(
  "For security reason, we sent you a random 4 digit OTP PIN via SMS!\n"
  "Please insert the OTP 4 digit PIN to authenticate: ");
__isoc99_scanf("%d", &v4);
if ( OTP != v4 )
{
  printf("Invalid OTP PIN! The right one was %04i \n", OTP);
  exit(0);
}
puts("PIN Correct! Here your bank account:");
strcpy(v7, "dge~cmLg``\"UdeUsBt\\J");
decrypt(v7, 20LL);
printf("%s \n", v7);
return 0;
```

Inserendo i dati precedenti e saltando all'indirizzo della funzione *check_otp*, nel mio caso "0x0000000000401510" poi continuando con *c* l'esecuzione, si arriva alla flag:

```
============================
Login to see your Bank Account
============================
Insert Username: UniPD_Student

Insert password: P10v3go!

============================
Welcome Back, UniPD_Student!
============================
```

```
Continuing.
For security reason, we sent you a random 4 digit OTP PIN via SMS!
Please insert the OTP 4 digit PIN to authenticate: 1234
PIN Correct! Here your bank account:
SPRITZ{P00r_45_DuCk}
[Inferior 1 (process 4108) exited normally]
```