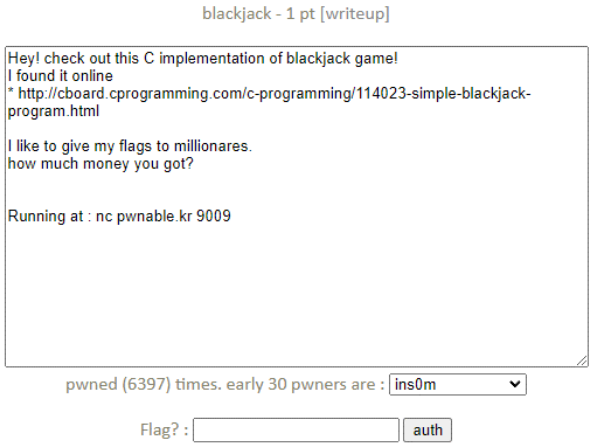
Pwnable.kr
- blackjack -

http://cboard.cprogramming.com/c-programming/114023-simple-blackjack-program.html
Running at : nc pwnable.kr 9009

0. 문제 살펴보기



블랙잭 게임을 하는 듯 보이는 문제이다.
백만장자가 되면 flag를 알려준다고 한다.

1. 소스 코드 확인

위에 있는 url 주소로 들어가면 블랙잭 게임의 소스 코드를 확인할 수 있다.
소스 코드는 꽤 길기 때문에 중요 부분만 캡처해보면 다음과 같을 것이다.

play()

```
void play() //Plays game
{

     int p=0; // holds value of player_total
     int i=1; // counter for asking user to hold or stay (aka game turns)
     char choice3;

     cash = cash;
     cash_test();
     printf("\nCash: $%d\n",cash); //Prints amount of cash user has
     randcard(); //Generates random card
     player_total = p + 1; //Computes player total
     p = player_total;
     printf("\nYour Total is %d\n", p); //Prints player total
     dealer(); //Computes and prints dealer total
     betting(); //Prompts user to enter bet amount

     while(i<=21) //While loop used to keep asking user to hit or stay at most twenty-one times
                  //  because there is a chance user can generate twenty-one consecutive 1's
     {
         if(p==21) //If user total is 21, win
         {
             printf("\nUnbelievable! You Win!\n");
             won = won+1;
             cash = cash+bet;
             printf("\nYou have %d Wins and %d Losses. Awesome!\n", won, loss);
             dealer_total=0;
             askover();
         }

         if(p>21) //If player total is over 21, loss
         {
             printf("\nWoah Buddy, You Went WAY over.\n");
             loss = loss+1;
             cash = cash - bet;
             printf("\nYou have %d Wins and %d Losses. Awesome!\n", won, loss);
             dealer_total=0;
             askover();
         }

         if(p<=21) //If player total is less than 21, ask to hit or stay
         {
             printf("\n\nWould You Like to Hit or Stay?");

             scanf("%c", &choice3);
             while((choice3!='H') && (choice3!='h') && (choice3!='S') && (choice3!='s')) // If invalid choice entered
             {
                 printf("\n");
                 printf("Please Enter H to Hit or S to Stay.\n");
                 scanf("%c",&choice3);
             }


             if((choice3=='H') || (choice3=='h')) // If Hit, continues
             {
                 randcard();
                 player_total = p + 1;
                 p = player_total;
                 printf("\nYour Total is %d\n", p);
                 dealer();
```

```
                if(dealer_total==21) //Is dealer total is 21, loss
                {
                    printf("\nDealer Has the Better Hand. You Lose.\n");
                    loss = loss+1;
                    cash = cash - bet;
                    printf("\nYou have %d Wins and %d Losses. Awesome!\n", won, loss);
                    dealer_total=0;
                    askover();
                }

                if(dealer_total>21) //If dealer total is over 21, win
                {
                    printf("\nDealer Has Went Over!. You Win!\n");
                    won = won+1;
                    cash = cash+bet;
                    printf("\nYou have %d Wins and %d Losses. Awesome!\n", won, loss);
                    dealer_total=0;
                    askover();
                }
            }
            if((choice3=='S') || (choice3=='s')) // If Stay, does not continue
            {
               printf("\nYou Have Chosen to Stay at %d. Wise Decision!\n", player_total);
               stay();
            }
         }
            i++; //While player total and dealer total are less than 21, re-do while loop
    } // End While Loop
} // End Function
```

나의 숫자 합이 21이면 승리하고, 21 초과면 패배하며, 21 이하라면 Hit 할 것인지, Stay 할 것인지를 정한다.
만약, Hit을 하였을 때, 딜러의 패가 더 좋다면 패배하지만, 나의 패가 더 좋다면 승리한다.
나의 패는 랜덤 함수에 의해 정해진다.

dealer()

```
void dealer() //Function to play for dealer AI
{
     int z;

     if(dealer_total<17)
     {
      srand((unsigned) time(NULL) + 1); //Generates random seed for rand() function
      z=rand()%13+1;
      if(z<=10) //If random number generated is 10 or less, keep that value
      {
          d=z;

      }

      if(z>11) //If random number generated is more than 11, change value to 10
      {
          d=10;
      }

      if(z==11) //If random number is 11(Ace), change value to 11 or 1 depending on dealer total
      {
          if(dealer_total<=10)
          {
              d=11;
          }

          else
          {
              d=1;
          }
      }
     dealer_total = dealer_total + d;
     }

     printf("\nThe Dealer Has a Total of %d", dealer_total); //Prints dealer total

} // End Function
```

딜러의 패 역시 랜덤 함수에 의해 생성되는 것을 볼 수 있다.

stay()

```
void stay() //Function for when user selects 'Stay'
{
     dealer(); //If stay selected, dealer continues going
     if(dealer_total>=17)
     {
      if(player_total>=dealer_total) //If player's total is more than dealer's total, win
      {
          printf("\nUnbelievable! You Win!\n");
          won = won+1;
          cash = cash+bet;
          printf("\nYou have %d Wins and %d Losses. Awesome!\n", won, loss);
          dealer_total=0;
          askover();
      }
      if(player_total<dealer_total) //If player's total is less than dealer's total, loss
      {
          printf("\nDealer Has the Better Hand. You Lose.\n");
          loss = loss+1;
          cash = cash - bet;
          printf("\nYou have %d Wins and %d Losses. Awesome!\n", won, loss);
          dealer_total=0;
          askover();
      }
      if(dealer_total>21) //If dealer's total is more than 21, win
      {
          printf("\nUnbelievable! You Win!\n");
          won = won+1;
          cash = cash+bet;
          printf("\nYou have %d Wins and %d Losses. Awesome!\n", won, loss);
          dealer_total=0;
          askover();
      }
     }
     else
     {
          stay();
     }

} // End Function
```

Stay 했을 때, 나의 패 숫자가 딜러보다 더 크거나, 딜러의 패 숫자의 합이 21이 넘을 경우, 나의 승리로 되며 아닐 경우 패배가 된다.

betting()

```
int betting() //Asks user amount to bet
{
 printf("\n\nEnter Bet: $");
 scanf("%d", &bet);

 if (bet > cash) //If player tries to bet more money than player has
 {
        printf("\nYou cannot bet more money than you have.");
        printf("\nEnter Bet: ");
        scanf("%d", &bet);
        return bet;
 }
 else return bet;
} // End Function
```

배팅할 금액을 받고 있으며 배팅 금액을 반환한다.
현재 가진 금액보다 많이 배팅을 할 경우, 배팅 금액을 다시 입력 받으며 그 금액을 반환한다.

askover()

```
void askover() // Function for asking player if they want to play again
{
    char choice1;

     printf("\nWould You Like To Play Again?");
     printf("\nPlease Enter Y for Yes or N for No\n");
     scanf("\n%c",&choice1);

    while((choice1!='Y') && (choice1!='y') && (choice1!='N') && (choice1!='n')) // If invalid choice entered
    {
        printf("\n");
        printf("Incorrect Choice. Please Enter Y for Yes or N for No.\n");
        scanf("%c",&choice1);
    }


    if((choice1 == 'Y') || (choice1 == 'y')) // If yes, continue.
    {
            system("cls");
            play();
    }

    else if((choice1 == 'N') || (choice1 == 'n')) // If no, exit program
    {
        fileresults();
        printf("\nBYE!!!!\n\n");
        system("pause");
        exit(0);
    }
    return;
} // End function
```

게임이 끝났을 때 문장으로 게임을 재시작 할 것인지 떠날 것인지 묻는 함수이다.

2. 취약점 파악

```
int betting() //Asks user amount to bet
{
 printf("\n\nEnter Bet: $");
 scanf("%d", &bet);

 if (bet > cash) //If player tries to bet more money than player has
 {
        printf("\nYou cannot bet more money than you have.");
        printf("\nEnter Bet: ");
        scanf("%d", &bet);
        return bet;
 }
 else return bet;
} // End Function
```

현재 betting 함수를 보면 잘못된 배팅 금액을 입력 받고 재입력을 받은 후에는 별도의 검사 없이 바로 값을 반환하는 것을 볼 수 있다.
만약, 이 점을 이용하여 현재 가진 돈보다 많은 금액을 배팅하고 승리 보상으로 그만큼의 금액을 받게 된다면 바로 백만장자가 될 수 있을 것이다.

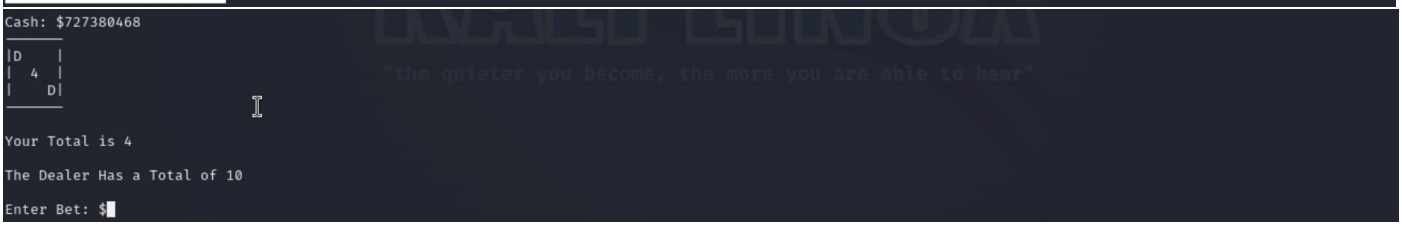
3. 게임 접속

위에 따로 표기해 놓은 netcat을 통해 실행하였다.
위와 같은 처음 화면이 등장하고 Y를 입력하여 진행을 시도하였다.

게임 메뉴 화면이 나오고 1을 눌러 게임을 시작하였다.

게임이 시작되고 배팅 금액을 입력 받고 있다.

일부로 처음에 현재 가진 돈보다 큰 금액을 입력한 뒤, 그 뒤에 재입력 부분에서 많은 돈을 입력해보았더니 성공적으로 다음 단계로 넘어간 것을 확인할 수 있다.
나는 Hit를 통해 카드를 더 넘겨받았다.



그 후, Stay를 하여 딜러만 카드를 뽑게 하였다.
그 결과는 나의 패배로 인하여 현재 Cash에서 내가 배팅한 금액이 마이너스가 될 것이다.
하지만, 우리는 배팅 금액을 재입력할 때, 굉장히 큰 수를 입력하였다.
이 수가 int형의 정수 범위를 벗어나 오버 플로우를 일으켰고 그 후, 나의 Cash는 백만 달러를 넘는 금액이 되었다. 때문에 백만장자를 달성하여 flag를 얻을 수 있게 되었다.
원래는 게임에서 승리하여 Cash를 늘릴 계획이었지만, 이와 같은 방식으로 해결하여 당황스럽다.