Day 07

Secure coding. Exploitation

Agenda

1. Quiz && Homework
2. Các lỗi nguy hiểm thường gặp khi lập trình
3. Thực hành khai thác lỗi phần mềm

Command injection

File: **command.c**

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    char s[200] = "echo ";
    puts("Nhap vao 1 so, toi se in ra so do. Hehe");
    fgets(s + 5, 100, stdin);
    system(s);
    return 0;
}
```

Buffer overflow

File: **bof.c**

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {
    char s[20];
    int N = 5;
    for (int i = 0; i < N; i++) {
    puts("Nhap vao 1 string");
    gets(s);
    printf("i = %d N = %d\n", i, N);
    }
}
```

Integer overflow/underflow

File: **iouf.c**

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

int main() {
    uint8_t a = 100;
    int8_t b = 200;
    if (a > b) {
        puts("yes");
    } else {
        puts("no");
    }
    return 0;
}
```

Integer overflow/underflow

File: **iof.c**

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

int main() {
    int8_t a = 125;
    for (int i = 0; i < 10; i++) {
    a++;
    printf("a = %d\n", a);
    }
    uint8_t b = -1;
    printf("b = %u\n", b);
}
```

File: **uaf.c**

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct dog {
    void (*bark)();
} dog;

typedef struct cat {
    void (*meow)();
} cat;

void bark() {
  puts("go go go...");
}

void meow() {
  puts("bruh.");
}

```

```c
void good() {
   dog *d = malloc(sizeof(dog));
   cat *c = malloc(sizeof(cat));
   d->bark = bark;
   c->meow = meow;
   d->bark();
   c->meow();
   free(c);
   free(d);
}
```

```
→ hsgs_admin ./uaf
go go go...
bruh.
→ hsgs_admin
```

```c
void weird() {
   cat *c = malloc(sizeof(cat));
   free(c);
   /* use only dog stuff here, no cat ever */
   c->meow();
}
```

Use-After-Free

```
29  void good() {
30      dog *d = malloc(sizeof(dog));
31      cat *c = malloc(sizeof(cat));
32      d->bark = bark;
33      c->meow = meow;
34      d->bark();
35      c->meow();
36      free(c);
37      free(d);
38  }
```

```
→ hsgs_admin ./uaf
go go go...
bruh.
→ hsgs_admin
```

Use-After-Free

```
22  void weird() {
23      cat *c = malloc(sizeof(cat));
24      free(c);
25      /* use only dog stuff here, no cat ever */
26      c->meow();
27  }
```

Agenda

1. Quiz && Homework
2. Các lỗi nguy hiểm thường gặp khi lập trình
3. Thực hành khai thác lỗi phần mềm

Non-printable input???

Một số chương trình trên thực tế cần input không gõ từ bàn phím được.

=> Viết chương trình gửi input cho chương trình khác

=> pwntools!!!

```
from pwn import *
p = process("./prog")
p.recv()
p.sendline("\x13\x01\x02\x03")
```

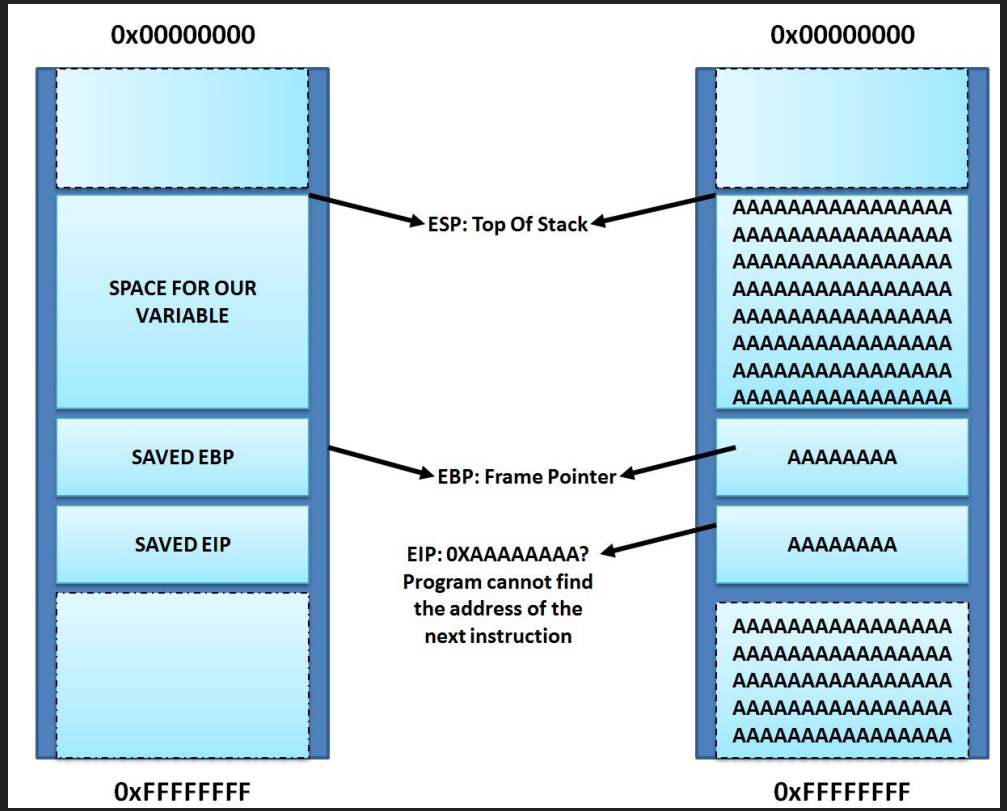
Serious Buffer Overflow

Review: Khi kết thúc 1 hàm, hàm đó sẽ nhảy về địa chỉ return address.

Chuyện gì xảy ra khi return address thay đổi trước khi hàm đó kết thúc?

Nếu người dùng chương trình có thể điều khiển return address đó thì sao?

Serious Buffer Overflow



Format String Error

Resources

https://softwareengineering.stackexchange.com/questions/175253/why-does-an-unsigned-int-compared-with-a-signed-character-turn-out-with-an-unexp

https://stackoverflow.com/questions/5416414/signed-unsigned-comparisons