

2020 시스템 해킹 교육 1회

2018. 07. 07.





INDEX



- | | |
|------|------------------|
| 001/ | 하계 방학 교육 소개 |
| 002/ | 시스템 해킹 이란? |
| 003/ | 리눅스 시스템 |
| 004/ | Integer Overflow |
| 005/ | 과제 설명 |

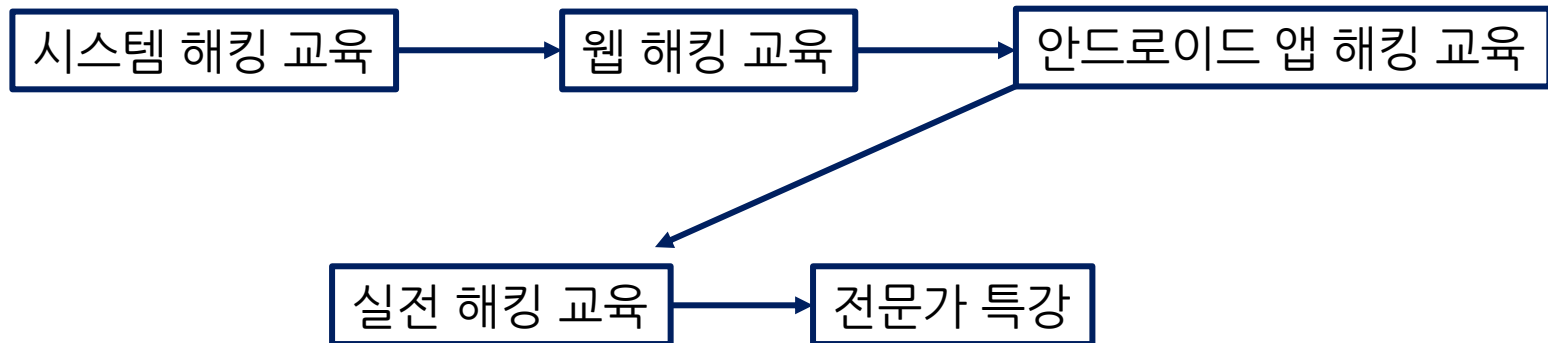


하계 방학 교육 소개



예년에는 시스템 해킹, 웹 해킹, GitHub 교육으로 구성하였으나
올해는 교육의 구성이 조금 바뀌었음

온/오프라인 동시 진행중



하계 방학 교육 소개



시스템 해킹 교육 진행 계획

-> 온 오프라인 동시 진행

-> 총 6 회차를 3주에 걸쳐 교육


-> 리눅스 시스템을 exploit 하는 원리에 대해 주로 학습



시스템 해킹이란? 시스템(system)?

시스템

위키백과, 우리 모두의 백과사전.

 다른 뜻에 대해서는 [시스템 \(동음이의\)](#) 문서를 참조하십시오.

시스템(**영어**: system)은 각 구성요소들이 상호작용하거나 상호의존하여 복잡하게 얽힌 통일된 하나의 집합체(unified whole)다. 또는 이 용어는 복잡한 사회적 체계의 맥락에서 **구조와 행동을 통제하는 규칙**들의 집합체를 일컫기도 한다.

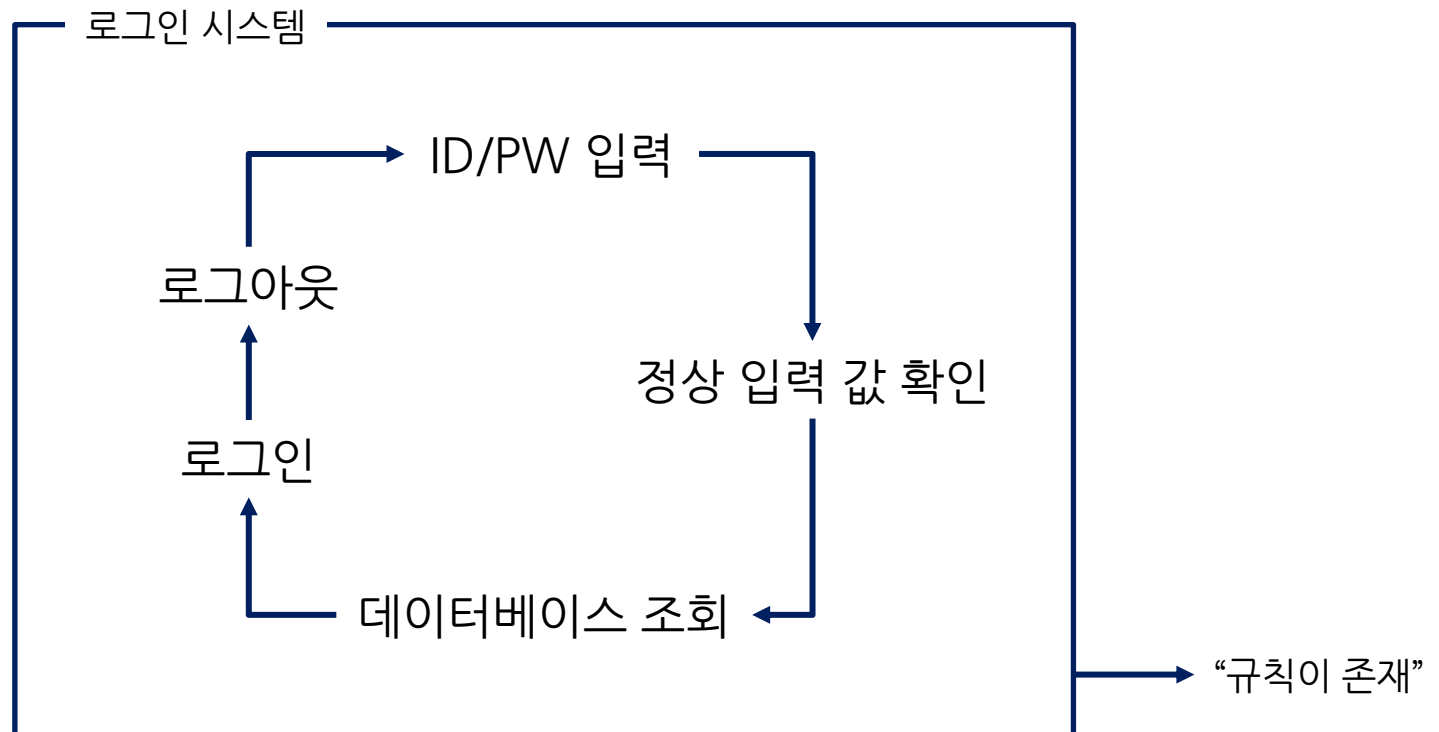
어떠한 구성 요소들이 **규칙**을 가지고 있는 것

시스템 해킹이란?

시스템(system)?



우리의 경우 컴퓨터 시스템(운영체제, SW/HW)





시스템 해킹이란? 해킹이란?

해킹

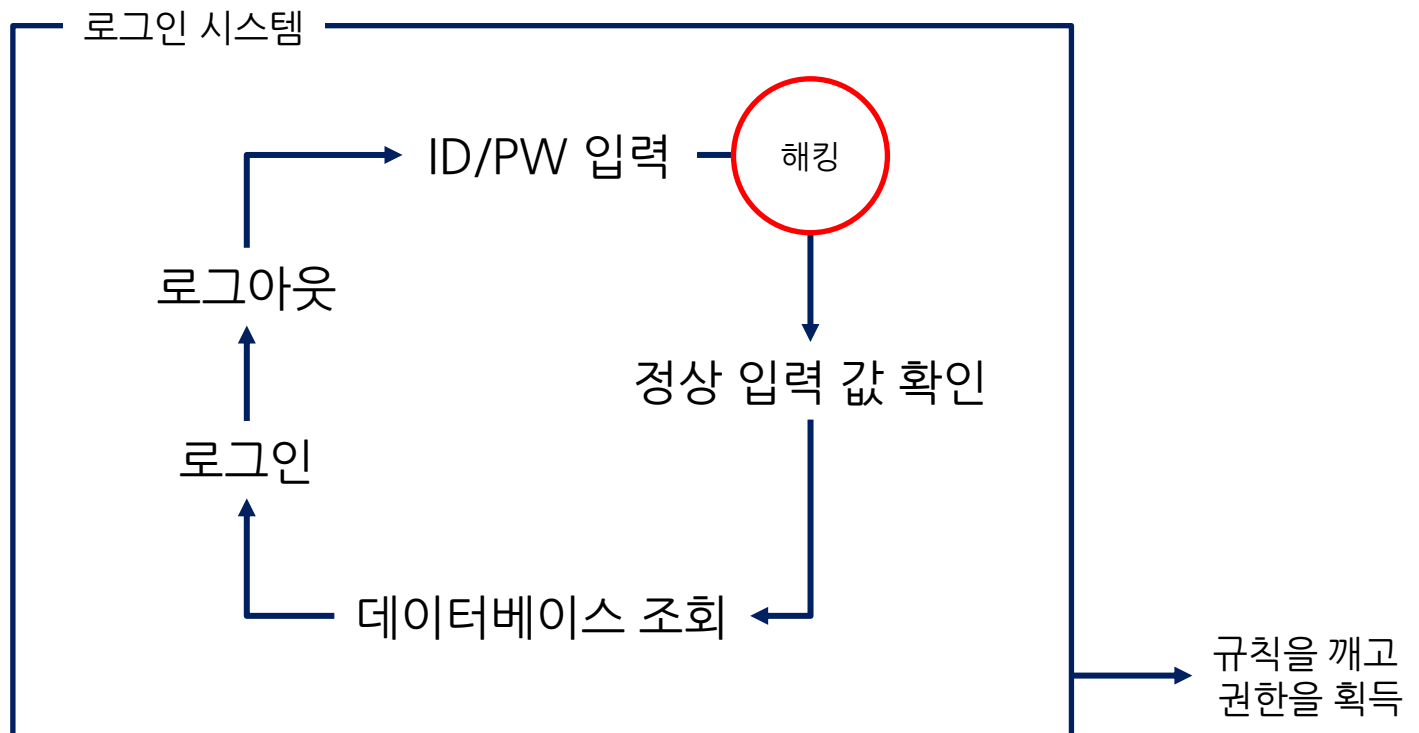
위키백과, 우리 모두의 백과사전.

해킹(hacking)은 타인의 컴퓨터 시스템에 무단 침입해 데이터에 접속할 수 있는 권한을 얻는 것이다. 전자 회로나 컴퓨터의 하드웨어, 소프트웨어, 네트워크, 웹사이트 등 각종 정보 체계가 본래의 설계자나 관리자, 운영자가 의도하지 않은 동작을 일으키도록 하거나 체계 내에서 주어진 권한 이상으로 정보를 열람, 복제, 변경 가능하게 하는 행위를 광범위하게 이르는 말로도 쓰인다.

시스템 해킹이란? 해킹이란?



시스템 해킹의 관점에서 보면..



시스템 해킹이란? 해킹이란?



⇒ 소프트웨어는 대부분 사람이 만든다

⇒ 사람이 만들다 보니 **치명적인 실수가** 있을 수 있다.

⇒ **개발자가 의도하지 않은 부분(취약점)** 을 공격해 악의적인 행동을 하는 것

시스템 해킹이란? 해킹이란?



[취약점 예시]

나는 사용자 입력 한다.

나는 유저를 생성 한다.

나는 시스템을 종료 한다.



시스템 해킹이란? 해킹이란?

[취약점 예시]

나는 사용자 입력 한다.

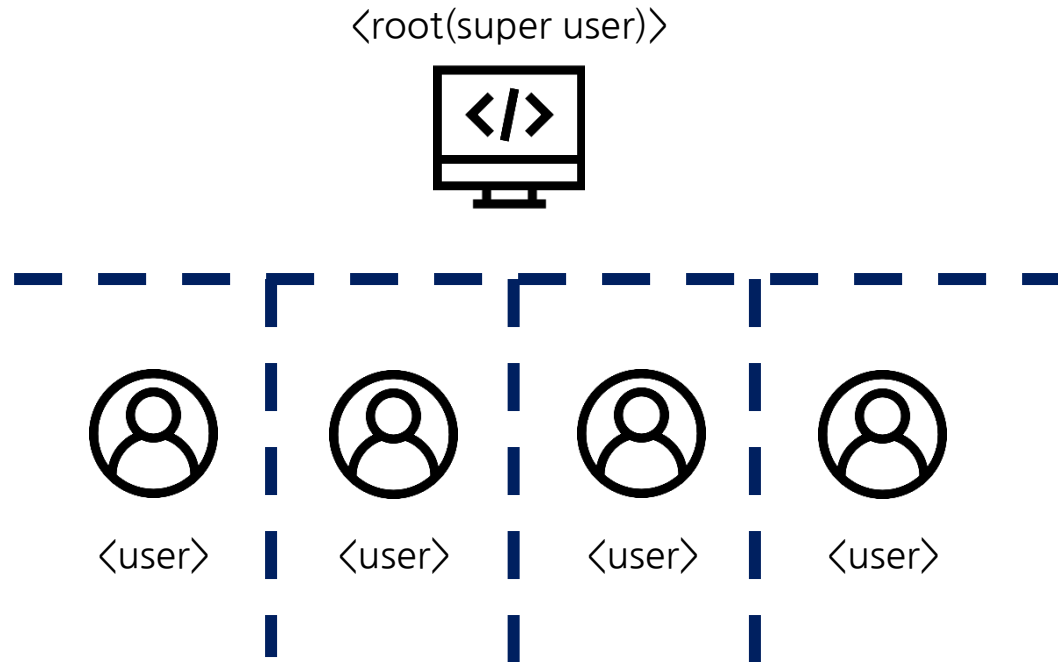
나는 파일 목록을 출력하고; 시스템을 전부 삭제 한다.



리눅스

#운영체제 #다중사용자 #오픈소스 #해커의 운영체제 #해킹입문

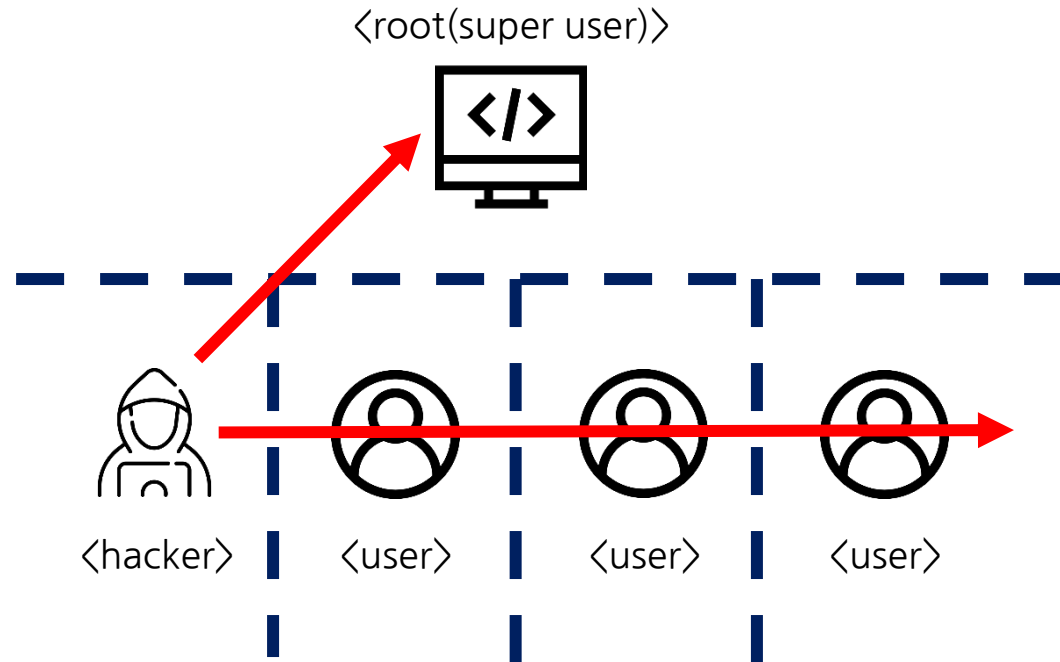
리눅스 시스템 리눅스란?



다중 사용자 시스템의 특징

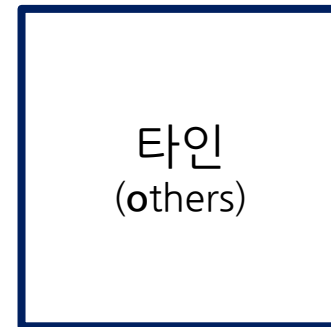
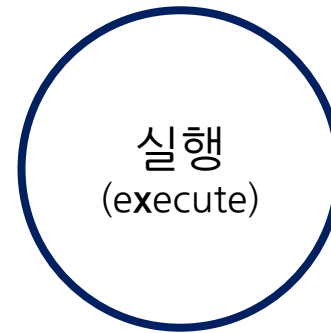
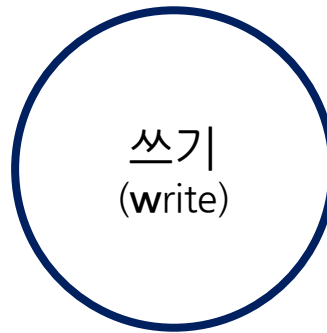
-> 서로 간의 영역을 **권한**을 이용해 철저히 분리

리눅스 시스템 리눅스란?



리눅스 시스템 해킹 == 본인이 가진 **권한**을 넘어서는 행위

리눅스 시스템 리눅스 권한





```
minibeef@argos-edu:~/cedu/week1$ ls -al
total 24
drwxrwxr-x 2 minibeef minibeef 4096 May  5 11:52 .
drwxrwxr-x 8 minibeef minibeef 4096 Jul  2 10:54 ..
-rwxrwxr-x 1 minibeef minibeef 8296 May  5 11:52 test
-rw-rw-r-- 1 minibeef minibeef   65 May  5 11:48 test.c
```

-rwxrwxr-x

이렇게 생긴 것이 해당 파일의 권한

리눅스 시스템 리눅스 권한



	사용자	그룹	다른 사람
-	<div>rwx</div>	<div>rwx</div>	<div>r-x</div>

r : 읽을 수 있다.

w : 수정할 수 있다.

x : 실행시킬 수 있다.



chmod(**change mode**) : 파일 권한 변경

```
minibee@argos-edu:~/cedu/week1$ chmod -x test
minibee@argos-edu:~/cedu/week1$ ls -al
total 24
drwxrwxr-x 2 minibee minibee 4096 May  5 11:52 .
drwxrwxr-x 8 minibee minibee 4096 Jul  2 10:54 ..
-rw-rw-r-- 1 minibee minibee 8296 May  5 11:52 test
-rw-rw-r-- 1 minibee minibee   65 May  5 11:48 test.c
```

```
minibee@argos-edu:~/cedu/week1$ ./test
-bash: ./test: Permission denied
```

“-” 한 후 rwx : 권한을 뺏겠다.

“+” 한 후 rwx : 권한을 주겠다.



chmod(**change mode**) : 파일 권한 변경

```
minibeef@argos-edu:~/cedu/week1$ ls -al
total 24
drwxrwxr-x 2 minibeef minibeef 4096 May  5 11:52 .
drwxrwxr-x 8 minibeef minibeef 4096 Jul  2 10:54 ..
-rwxrwxrwx 1 minibeef minibeef 8296 May  5 11:52 test
-rw-rw-r-- 1 minibeef minibeef  65 May  5 11:48 test.c
```

```
minibeef@argos-edu:~/cedu/week1$ chmod u-x test
minibeef@argos-edu:~/cedu/week1$ ls -al
total 24
drwxrwxr-x 2 minibeef minibeef 4096 May  5 11:52 .
drwxrwxr-x 8 minibeef minibeef 4096 Jul  2 10:54 ..
-rwxrwxrwx 1 minibeef minibeef 8296 May  5 11:52 test
-rw-rw-r-- 1 minibeef minibeef  65 May  5 11:48 test.c
minibeef@argos-edu:~/cedu/week1$
```

빼을 or 부여할 권한 앞에 u(user), g(group), o(others)를 넣으면 해당 인원의 권한만 수정가능



(실습1) 간단하게 권한 체험해보기

[vi에 대한 사용법 설명 필요]

1. vi test를 통해 텍스트 파일 생성
2. 아무 문장이나 써서 저장
3. cat test를 통해 파일 읽기(r)
4. chmod -r test
5. cat test하고 달라진 점 확인

첫 취약점!

Integer Overflow



변수는 자료형마다 담을 수 있는 크기가 정해져 있다.

char : 1 byte
int : 4 byte
double : 8 byte

Integer Overflow는 이러한 자료의 유한함에서 발생한다.

첫 취약점!

Integer Overflow



Integer Overflow

정수

흘러 넘침

2,147,483,647

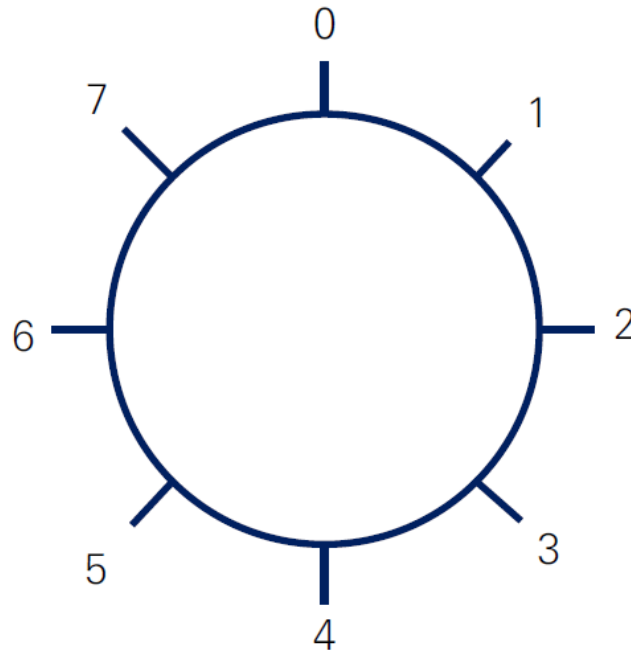
정수를 담을 수 있는 크기를 넘어서면 음수나 아주 작은 수로 바뀌는 버그

첫 취약점!

Integer Overflow



아래와 같이 0~7을 나타내는 수 체계가 있다고 하자

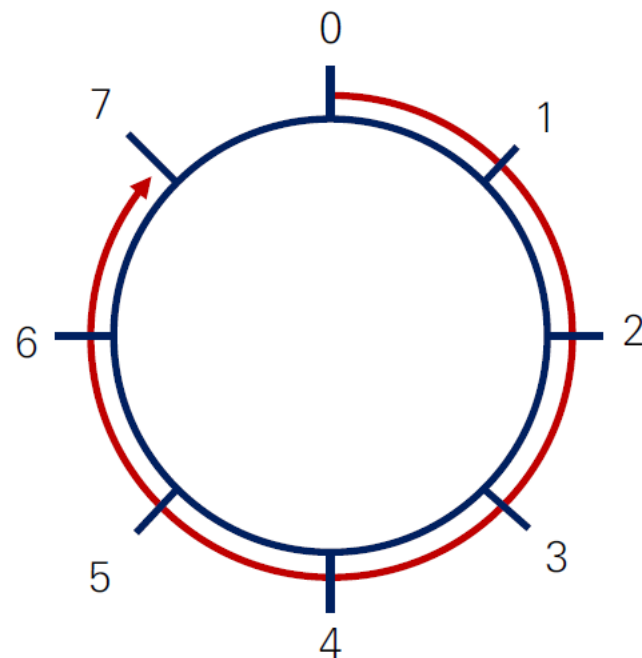
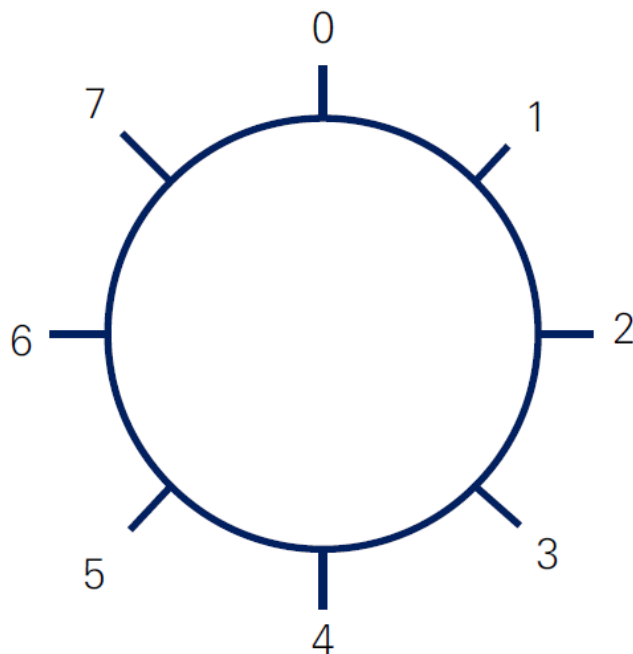


첫 취약점!

Integer Overflow



0에서 부터 7만큼 이동하면 7이다.

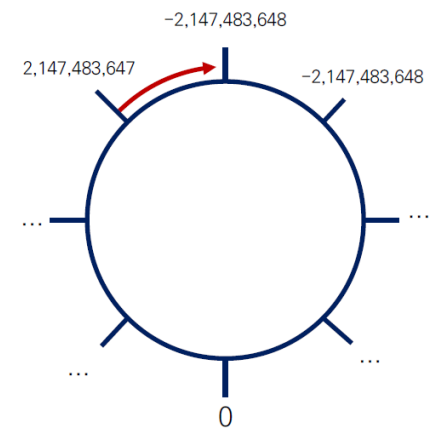
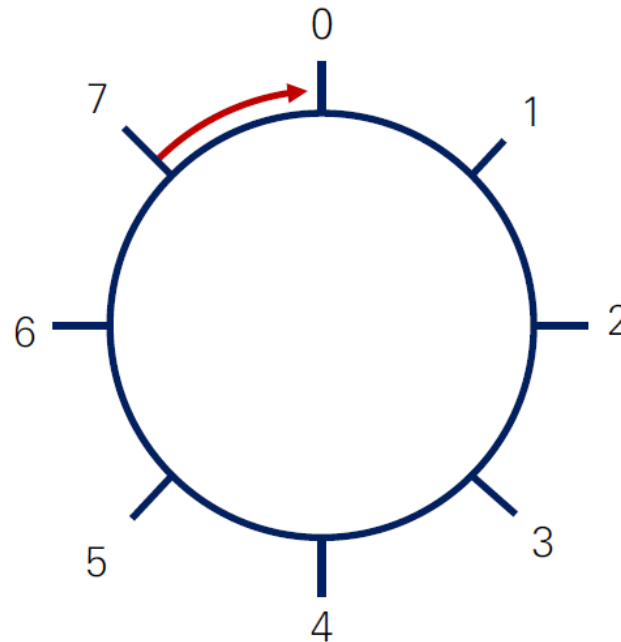


첫 취약점!

Integer Overflow



7(최대치)로부터 1만큼 이동하면 최솟값인 0으로 바뀐다.



첫 취약점!

(실습2) Integer Overflow



```
1 ARGOS × +  
#include <stdio.h>  
  
int main()  
{  
    int a = 2147483647;  
    printf("int max : %d\n", a);  
  
    a++; // a = a + 1  
    printf("int max + 1 : %d\n", a);  
}
```

```
minibee@cargos-edu:~/cedu/week2$ ./prac3  
int max : 2147483647  
int max + 1 : -2147483648
```

첫 취약점!

(실습2) Integer Overflow



1. vi를 이용해 파일 생성
2. 소스코드 작성
3. :wq 저장하고 나가기
4. gcc -o [출력 파일] [소스 파일]
5. ./[출력 파일]

첫 취약점!

(실습2) Integer Overflow



[추가 지식 - 컴파일]



언어가 다른 사람과 대화할 때 번역기가 필요 하듯이

인간은 이진수를 사용하는 컴퓨터와 대화하기 위해 컴파일러가 필요하다.



첫 과제!

(과제 1) My First Hacking

```
#include <stdio.h>
int main()
{
    unsigned int money = 0;
    int salary = 0;

    printf("\nhello, Your Money is %d$\n\n", money);

    printf("===== PAYDAY =====\n");
    printf("BOSS : How much do you want to get paid?\n");
    printf("Me : ");
    scanf("%d", &salary);

    if(salary > 100) {
        printf("BOSS : You're fired!\n");
    } else {
        printf("BOSS : Sure, Good choice\n");
        money += (unsigned int) salary;
    }

    printf("\nYour Money is %u$\n", money);
    if(money > 10000) {
        printf("You Win!\n");
    } else {
        printf("You Lose!\n");
    }
}
```

기다리고 기다리던 월급날이 되었다! 사장님께 돈을 얼마나 달라고 할까?

1. 처음 돈은 0\$ 이다.
2. 입력을 통해 내가 원하는 만큼 월급을 제시할 수 있다.
3. 제시한 금액이 100\$를 넘어가면 해고 당하고, 돈은 하나도 못 받는다.
4. 하지만 내 잔고에는 10000\$ 이상이 있어야 승리한다.

* Type Casting : 변수의 자료형을 바꾸는 것

* unsigned int : 0~4,294,967,295

첫 과제!

(과제 1) My First Hacking



```
minibee@argos-edu:~/sysedu$ ./hw1  
  
hello, Your Money is 0$  
  
===== PAYDAY =====  
BOSS : How much do you want to get paid?  
Me : XXXXXXXXXX  
BOSS : Sure, Good choice  
  
Your Money XXXXXXXXXX  
You Win!  
minibee@argos-edu:~/sysedu$ █
```

Q & A

Thank You for Listening

