2022 A Basic CS skill: ABC Winter School

Security & Cryptography

Team 8

2022 / 02 / 06

ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY

공지사항

날짜	주제	날짜	주제
1/2	Course Overview (TODAY)	2/4	Metaprogramming
1/5	Git	2/6	Security and Cryptography
1/16	Shell 1 (기본 명령어)	**2/13**	Debugging and Profiling
1/19	Shell 2 (스크립트 작성)	2/16	Potpourri
1/23	Text editor(Vim)	2/20	Special topic 1
1/28	Data Wrangling	2/23	Special Topic 2 and Q&A

- 2/7 ~ 2/9 학회 참석으로 인해 수업이 어려울 것 같습니다.
- 2/9 수업 -> 2/13 수업

문제!

100개의 약 중 하나가 독약이다

독약을 먹으면 1시간 후 죽는다

1시간이 주어졌을 때 독약을 찾기 위해 필요한

최소한의 토끼 수는?

(약을 먹이는데 걸리는 시간 무시, 약의 양 무한정)

정답!

CV #

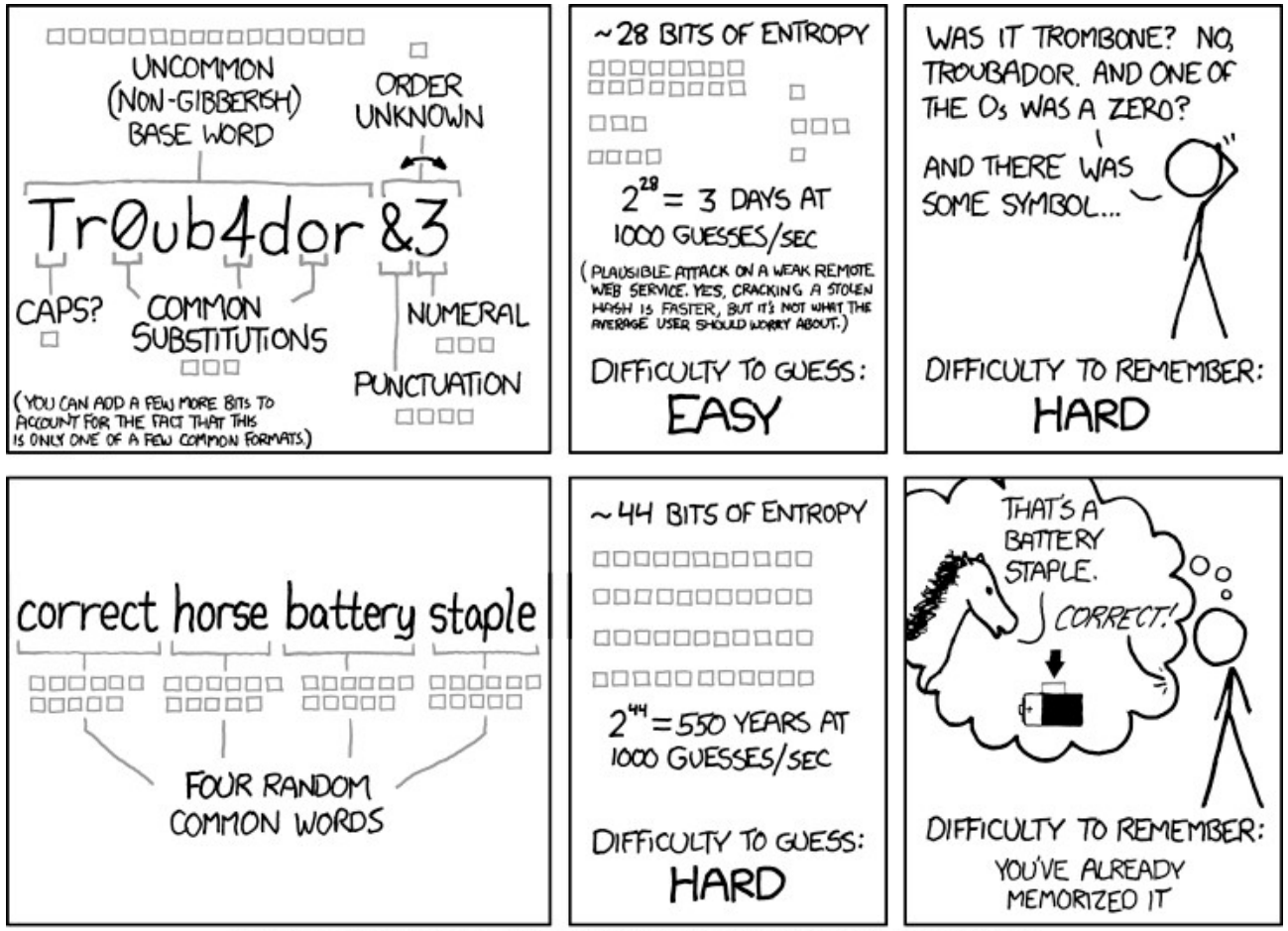
1번약 (이진수 0000001번약)은 1번 토끼 빼고 다 먹입니다.
2번약 (이진수 0000010번약)은 2번 토끼 빼고 다 먹입니다.
...
89번약 (이진수 1011001번약)은 1, 4, 5, 7번 토끼 빼고 다 먹입니다.
...
100번약 (이진수 1100100번약)은 3, 6, 7번 토끼 빼고 다 먹입니다.

만약 89번약이 독약이라면 1, 4, 5, 7번 토끼는 살고 나머지는 죽겠죠.
산 토끼를 1, 죽은 토끼를 0이라고 하면
7번토끼~1번토끼 순서대로 1011001 이 되고, 십진법으로 바꾸면 89가 나오게 됩니다.

Entropy

- 열역학적 엔트로피 -> 에너지의 무질서도
- 여기서는 **정보 이론**에서 말하는 엔트로피를 뜻합니다.
- 열역학에서의 엔트로피와 비슷하게 정보 이론에서의 엔트로피는 정보의 불확실성을 나타냅니다.
- 데이터의 단위로의 정보 엔트로피는 저장 또는 통신에 사용되는 평균 비트 수로 표현 됩니다.
- Ex) 매일 날씨의 정보가 2비트로 표현 될 수 있다면 하루의 날씨를 평균 2비트로 나타낼 수 있다.

Entropy


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS:
EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER:
HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS:
HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER:
YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

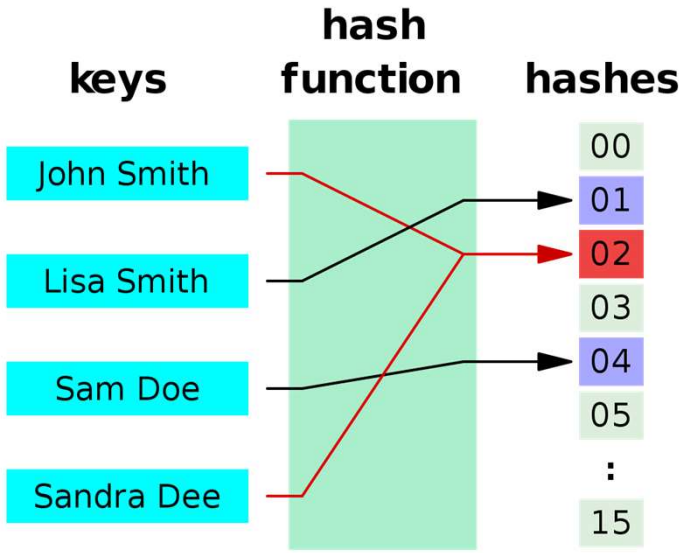
용어

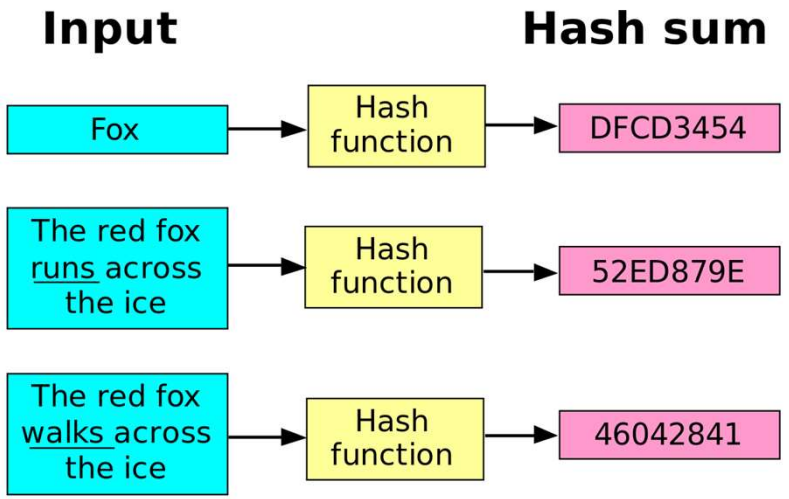
- 키 (key)
- 평문 (plaintext, message)
- 암호문 (ciphertext)
- 암호화 (encryption)
- 복호화 (decryption)

시저 암호 예시

Hash Function

- 임의의 길이를 가진 데이터를 고정된 크기의 데이터로 특정한 규칙을 통해 매핑하는 역할을 합니다.
- 이는 암호화랑은 차이가 있습니다. (복호화가 없기 때문입니다.)
- 대표적으로 Git에서 사용하였던 SHA-1 Hash가 있습니다.



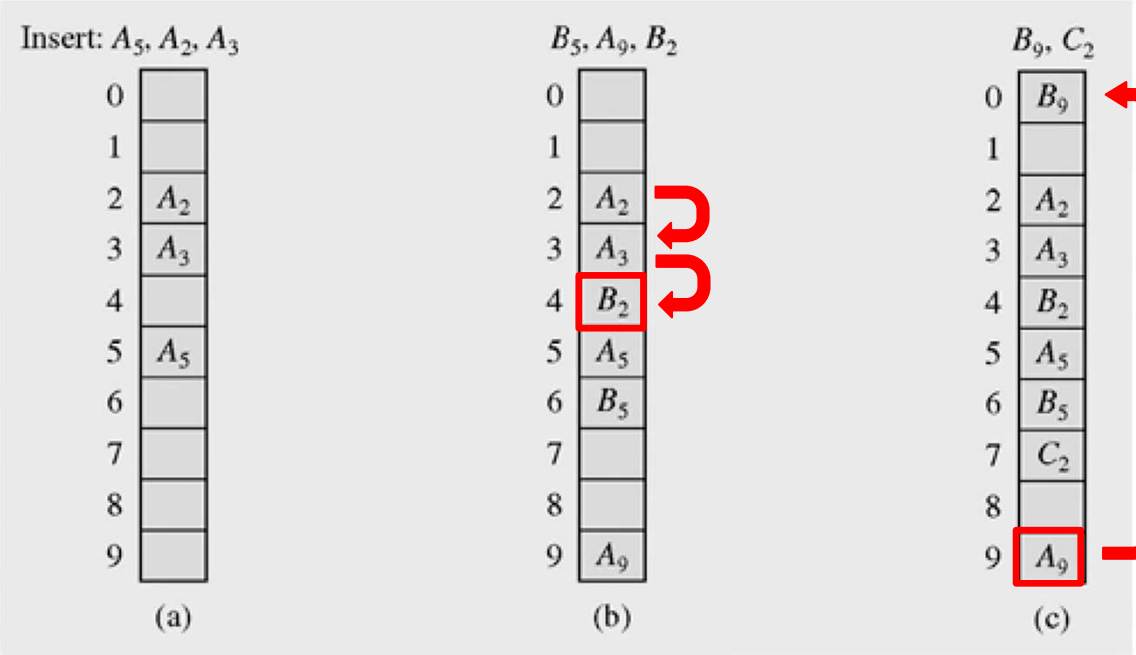


Hash Function

- Deterministic : 같은 input을 hash function에 넣으면 결과는 항상 같습니다.
- Non-invertible : output을 input으로 되돌리는 방법이 없습니다. (단방향)
- Hash collision : 여러 개의 input이 동일한 값으로 매핑 되는 경우를 말합니다.
- Ex) hash_function(input) = input % 10
  - Input은 10의 나머지 값을 hash function의 output으로 매핑 됩니다.
  - 이 경우, 11과 121 input은 같은 1 값으로 매핑이 됩니다.
  - Hash collision은 이러한 상황을 의미합니다.
- Hash collision을 해결하는 방법으로는 크게 open-addressing과 chaining이라는 방법이 일반적으로 많이 사용됩니다.

Open addressing

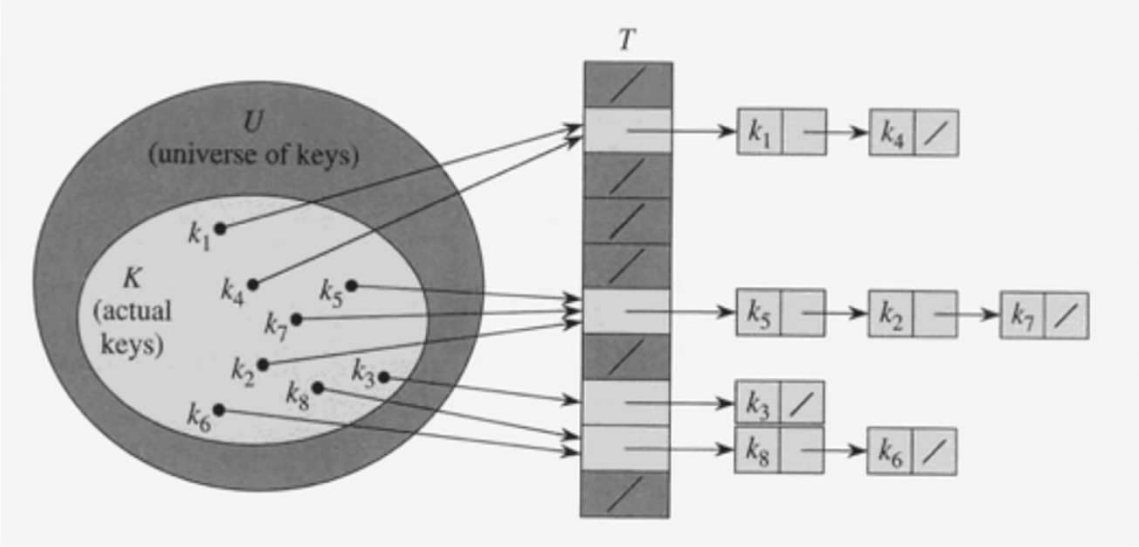
- 충돌이 일어나는 값이 들어온 경우, 정해진 policy에 따라 다른 값에 매핑을 하게 됩니다.
  - Linear probing, Quadratic probing 등등…



Linear probing 예시

Chaining

- 동일 장소로 해싱된 모든 값들을 하나의 linked list로 저장합니다.
- Open addressing에서 발생할 수 있는 충돌이 발생하지 않습니다.
- 모든 값들이 하나의 output으로 해싱되는 경우 검색에 많은 시간이 필요할 수 있습니다.

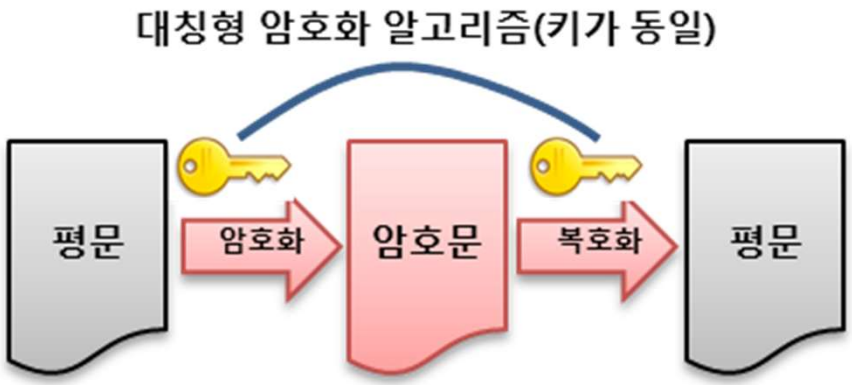


Key Derivation Function (KDF)

- 키 유도 함수라고 부릅니다.
- Master key나 어떠한 비밀 정보로부터 특정 시스템이나 네트워크의 암호화, 또는 복호화에 필요한 암호 키를 유도하는 함수를 의미합니다.
- 즉, 키 값을 만드는 일종의 키 생성기(key-gen), 혹은 난수 생성기 입니다.
- Ex) ssh-keygen 명령어를 사용하면 passphrase를 묻습니다.
  - ssh-keygen은 입력한 passphrase를 기반으로 KDF를 실행합니다.
  - KDF로부터 private key와 public key가 생성됩니다.

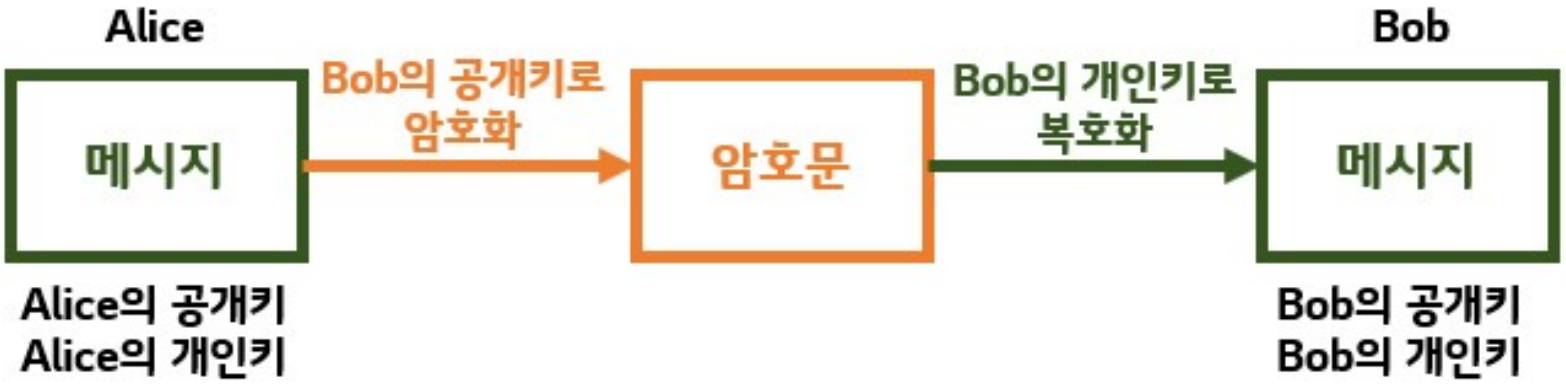
Symmetric Cryptography

- 암호화와 복호화에 동일한 키가 사용 됩니다.
- 비대칭 알고리즘에 비해 암호화/복호화 속도가 빠릅니다.
- 주로 데이터 통신에서 사용 됩니다. (통신에서는 속도가 중요하기 때문)
- 동일한 키를 암호화와 복호화에 사용하기 때문에 공격자가 키를 가로채면 암호가 쉽게 풀린다는 단점이 있습니다.



Asymmetric Cryptography

- 비대칭 암호화 방식이라고 합니다. (또는 공개키 암호화 방식)
- Public key와 Private key를 이용해 암호화와 복호화를 수행합니다.
- Public key => 암호화에 사용, Private key => 복호화에 사용
- 보안 강도가 높은 만큼 연산이 복잡하고 느립니다.



RSA 암호

- **"엄청 큰 숫자는 소인수 분해 하기 힘들다."**
- Rivet, Shamir, Adelman (만든 사람의 첫 글자를 따서 RSA)
- 전세계 대부분의 인터넷 뱅킹에서 사용하는 암호화 알고리즘.
- 공개키와 개인키를 이용하는 Asymmetric Cryptography이다.
  - 두 소수 p, q를 준비한다.
  - p-1, q-1과 서로소인 정수 e를 준비한다.
  - (e*d) % {(p-1)*(q-1)} = 1을 만족하는 d를 찾는다. -> **d가 개인키**가 된다.
  - N = p * q를 계산하고 N과 e를 공개한다. -> **(N, e)가 공개키**가 된다.

RSA 암호

- RSA 암호화 방법
  - 보내려는 평문 m (< N)가 있으면 x = m^e (mod N) 으로 암호화 한다.
- RSA 복호화 방법
  - 받은 암호문 x가 있으면, m = x^d (mod N) 으로 복호화 한다.

- 예시 (p = 11, q = 3, m = 2)
  - p-1 = 10, q – 1 = 2 이므로, 10과 2의 서로소는 3, 7, …
  - e = 3 선택, N = p * q = 33 이므로, 공개키 (N, e) = (33, 3)
  - (3 * d) % {(p-1) * (q-1)} = (3 * d) % 20 = 1을 만족하는 d는 7, 따라서 암호키 d = 7
  - 평문 m = 2를 암호화 하면, x = 2^3 (mod 33) = 8
  - 암호문 x = 29를 복호화 하면, m = 8^7 (mod 33) = 2

SSH key

- SSH는 key 등록을 통해 다른 서버에 로그인 과정 없이 접속을 가능 하게 할 수 있습니다. (학교에서는 보안상 권장은 X)
- ssh-keygen : ~/.ssh 폴더에 고유한 ssh key를 생성 합니다.

```
[cs20151509@uni06 .ssh]$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
[cs20151509@uni06 .ssh]$
```

  - id_rsa : 생성된 private key (유출되면 안됨!!)
  - id_rsa.pub : 생성된 public key
  - authorized_keys : 접속이 허용되는 public key 리스트
  - known_hosts : 접속 기록이 남아 있어서 알려진 호스트(ip)들

chmod

- 특정 파일이나 폴더의 이용 권한을 바꾸는 명령어 입니다.
- chmod [변경할 권한] [파일 또는 폴더]
- 변경할 권한은 숫자 3자리로 나타내며, 각 자릿수는 소유자, 그룹, 기타 사용자를 나타냅니다.

Ex) chmod 754 test.py

파일의 소유자는 7 (읽기, 쓰기, 실행) 권한
그룹은 5 (읽기, 실행) 권한
기타 사용자는 4 (읽기) 권한

r : 읽기 w : 쓰기 x : 실행

0 = --- = 0 + 0 + 0
1 = --x = 0 + 0 + 1
2 = -w- = 0 + 2 + 0
3 = -wx = 0 + 2 + 1
4 = r-- = 4 + 0 + 0
5 = r-x = 4 + 0 + 1
6 = rw- = 4 + 2 + 0
7 = rwx = 4 + 2 + 1

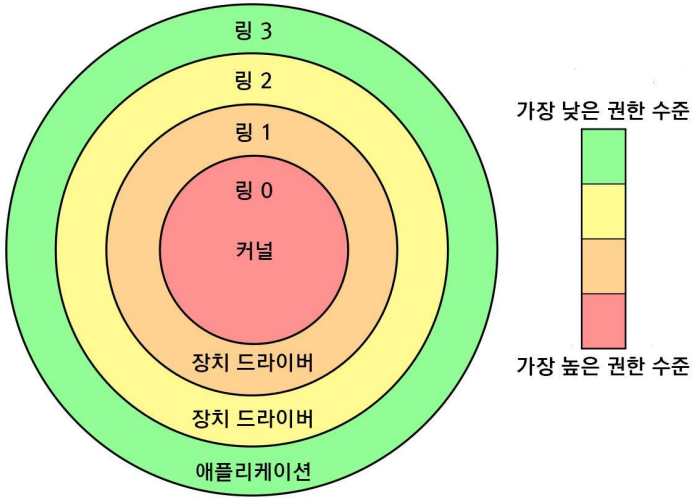
chmod

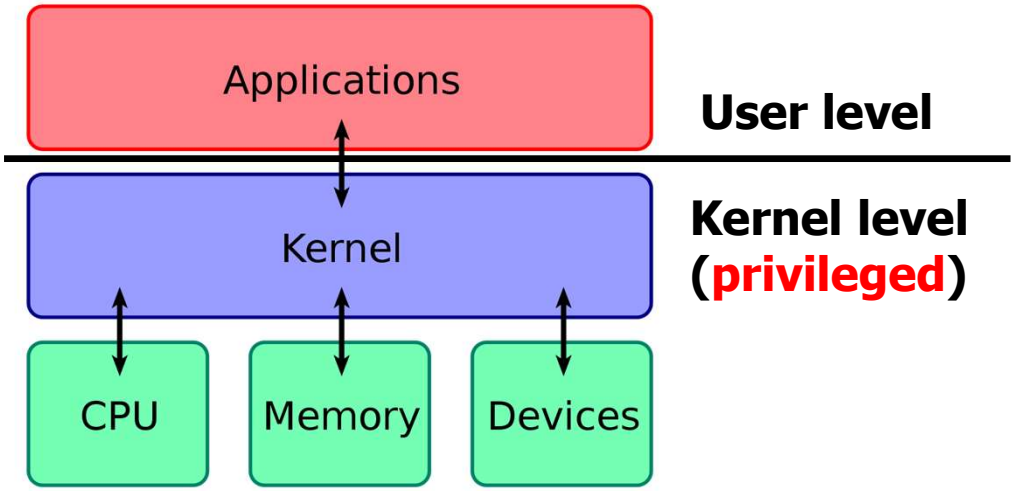
- SSH key를 이용하기 위해서는 각 파일들이 적절한 permission을 가지고 있어야 합니다.
- 지난 시간에서도 언급하였듯, private key는 유출이 될 경우 보안에 문제가 발생하기 때문에 각 파일들은 아래와 같은 permission을 유지해야 합니다.

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/id_rsa
chmod 644 ~/.ssh/id_rsa.pub
chmod 644 ~/.ssh/authorized_keys
chmod 644 ~/.ssh/known_hosts
```

Kernel

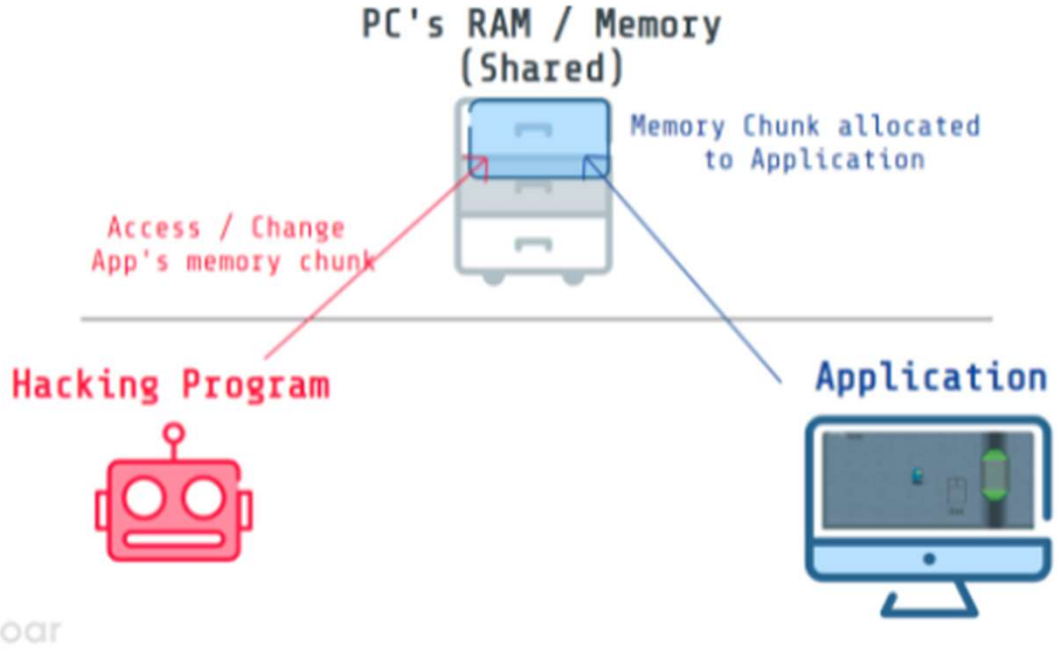
- 하드웨어의 자원을 필요한 프로그램에게 나눠주고 제어하는 운영체제의 가장 핵심적인 부분.
- 모든 프로그램은 자원이 있어야 실행되기 때문에 보안도 담당한다.





Kernel

- 대부분의 해킹은 memory 조작에서 부터 시작합니다.
  - 모든 프로그램은 메모리에 올라와야 그 때부터 무언가 작업을 하는 상태인 **"프로세스"** 가 되기 때문.

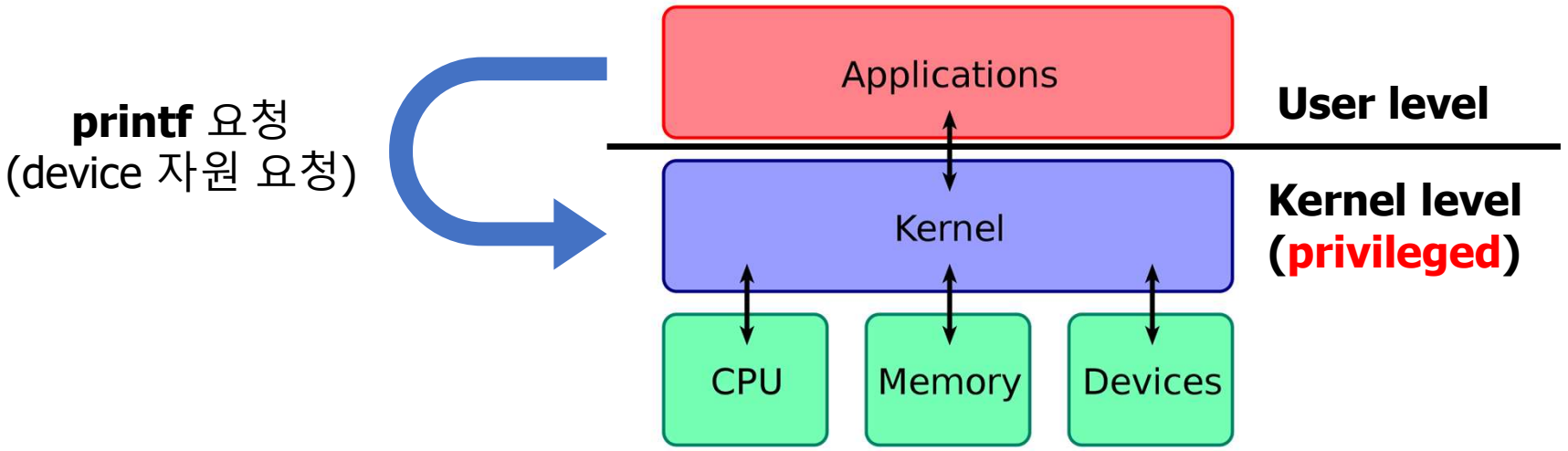


Activity

- Ubuntu의 system call을 가로채는 Hooking을 만들어 봅시다.
  - **절대 학교 서버에서 하지 마세요!**

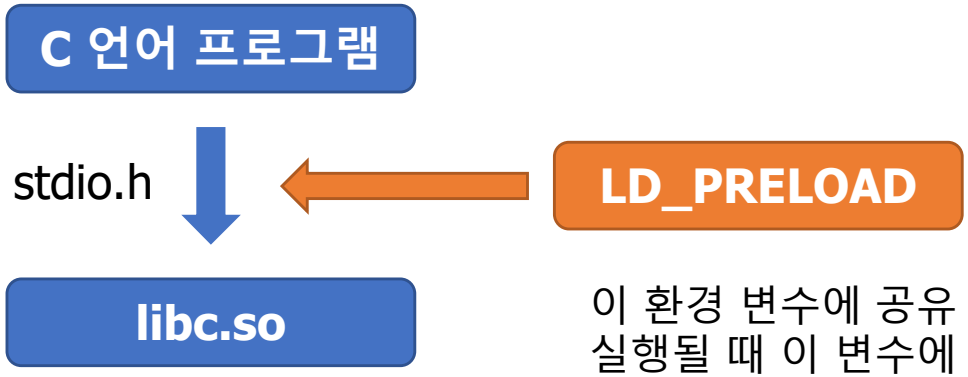
Activity

- 우선 System call이란?
  - 어떤 User level 프로그램이 kernel에 요청을 하여 kernel이 가지고 있는 기능을 사용하는 것.
  - 즉, kernel에 접근하기 위한 인터페이스.



Activity

- Shared library (공유 라이브러리 / 동적 라이브러리)
  - .so로 된 확장자를 가짐.
  - 각 라이브러리를 pointer로 참조하여 사용함.
  - OS만 같다면 다른 언어로 짜여진 함수도 실행 가능.



이 환경 변수에 공유 라이브러리가 지정 되면 프로세스가 실행될 때 이 변수에 지정된 라이브러리가 먼저 실행.

Thank you


UNIST
ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY
2009