Lec 8

Only owner (root) sets file permissions.

r/w/x bits for owner, group other.

Reason of Groups is Compaction.

Root can do anythings

Privilege Escalation → run some code as another user.

Real user ID - 실제 실행하고 있는 사람

Effective user ID - 현재 실행하는 권한 :

Saved user ID - EUID를 저장 → SUID 권한 낮출때
권한 복귀때는 SUID → EUID

UID bit = 1 when It is 1 EUID = owner of executable.

GID bit // Effective group ID
= the group of executable.

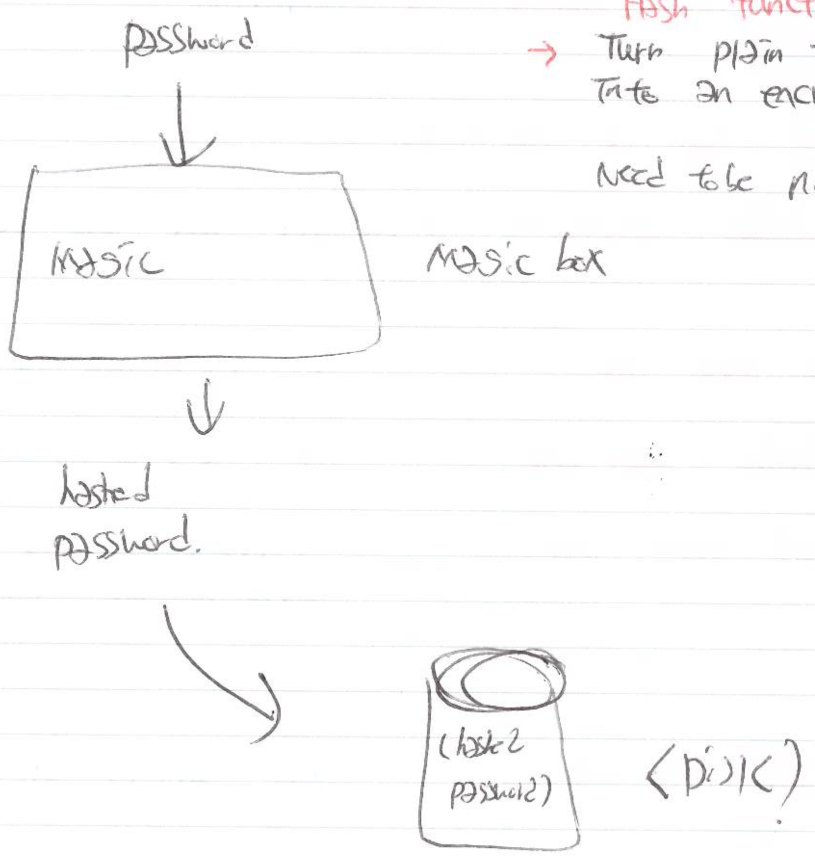
Capabilities = Specific task (few lines of code)
핵심코드 e.g.) 디스크 읽어라

Policy KH 이걸 대신 해주게 하는 것은 Policy KH 이라고 한다

Windows - two authentication tokens
① normal unprivileged user
② administrator right.

예를 들어서 window 에서 Cob실에 [illegible] 같은데 privileged executable
있고 ~~unprivilisel~~ unprivilige user로 실행하고
다른 물어보면 switch token 해서 administrator 권한으로
변경

Hash ← ~~box~~ password generally secured on the computer called Hash
↳ mathematical things



Hash functions
→ Turn plain text password Into an encrypted representation

Need to be non-reversable.

① 그리고 해서 저장된 hashed password 가 읽히더라도 좋다.

누구나 password를 볼수 있지만 실제 password가 아닌 hashed password 이다.

비번 잃어버리면 찾을 수가 없다 Disk에 저장되어 있는 것은 Hashed password 이기 때문에 하지만 비번을 바꿀수는 있다.

Hash attack

1) Hash functions work because it can't be ~~be~~ non-reversable. but if we use rainbow table it can be changed
rainbow table = 해시 functions를 사용하여 변환 가능한 모든 Hash value 값을 저장시켜 놓은 table.

Hash attack

1) obious password is obvious

2) Brute-force (rainbow table)

3) Social engineering (just ask)

SQL 공격방법 escaping character. ' ; --

XSS (Cross site ~~site cross~~ scripting attack)

- Non persistant
  - modify, URL parameter
  - HTTP header
  - Information attack
- persistant
  - More serious

d

131