4주차 1차시. 패스워드 크래킹에 대한 이해

【학습목표】

1. 패스워드 크래킹에 대해 설명할 수 있다.

학습내용1 : 패스워드 관리

보안 관리자의 첫 번째 방어책

1. 크래킹되기 쉬운 패스워드

* 길이가 너무 짧거나 널(Null)인 패스워드
1, a
* 사전에 나오는 단어나 이들의 조합으로 이루어진 패스워드
ehtk(도사), godqhr(행복)
* 키보드 자판을 일련순으로 나열한 패스워드
1234, asdf, qwer
*사용자 계정 정보에서 유추 가능한 단어들로 된 패스워드
사용자의 이름 또는 계정에서 유추가 가능한 단어
Wishfree 계정의 wishfree76

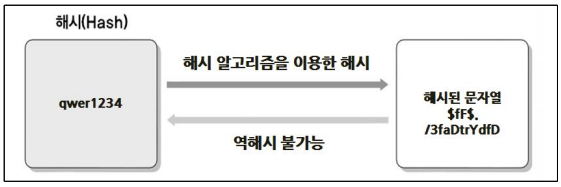
2. 크래킹되기 쉬운 패스워드

① 기억하기는 쉽고 크래킹하기 어려운 패스워드
② 최소 8자 이상의 패스워드
패스워드는 길면 길수록 좋음
③ 대문자와 소문자 조합
대문자와 소문자를 조합하여 패스워드를 추측하기 어렵게 조합
④ 글자와 숫자를 조합
영문자와 숫자를 조합하여 보안을 강화
⑤ 특수 문자를 조합
-, _, &, $, > 와 같은 특수를 문자를 조합하여 보안을 강화
⑥ 좋은 패스워드 예시
eodlf@!11’
⑦ 패스워드 검증 사이트를 이용
https://howsecureismypassword.net/

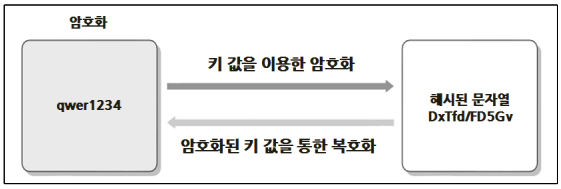
학습내용2 : 해시의 암호화

1. 운영체제에서 패스워드를 숨기는 방법

* 해시(Hash)
해시 : 임의의 데이터로부터 일종의 짧은 전자 지문을 만들어 내는 방법
해시 함수를 이용
· 데이터를 자르고 치환하거나 위치를 교환하는 방법
· 해시 값(hash value) : 해시 함수를 통해 나온 결과 값
· 해시 값에서 원래를 데이터를 구하는 것이 불가능 해야 함

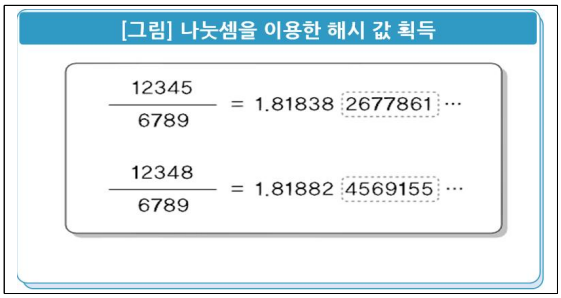


* 암호화 (Encryption)
암호화 : 특별한 알고리즘을 이용해 데이터를 전달하는 것
암호화 알고리즘을 아는 사람은 복호화(Decryption)하여 원래의 값을 읽을 수 있음



2. 해시와 암호화의 차이점

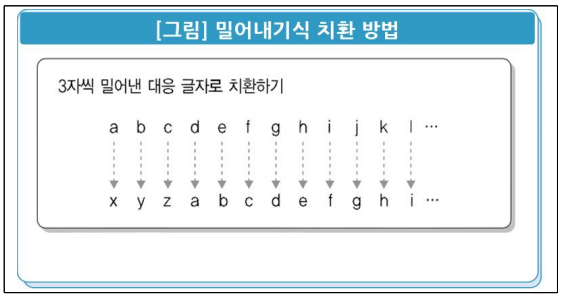
* 해시 - 나눗셈을 이용한 해시 알고리즘
· 두 수를 가운데를 기준으로 둘로 나누고 큰 수를 작은 수로 나눔
· 앞 6자리 숫자를 버리고 나머지 값이 해시의 결과 값
이용하는 수 : 123456789, 123486789
나눗셈을 이용해 얻은 해시 값
· 123456789의 해시 값 : 2677861
· 123486789의 해시 값 : 4569155

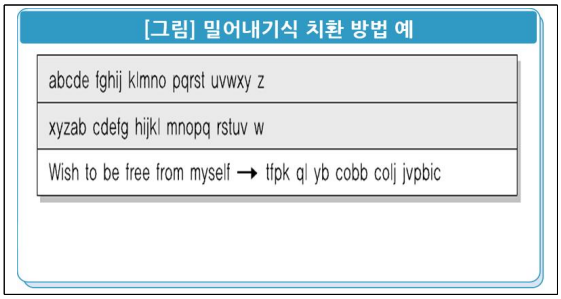


로직을 알더라도 버려진 1.81838과 1.81882를 알 수 없으므로 두 해시 값만으로 해시 전의 원래 수를 알아내는 것은 불가능

로직을 알고 있을 경우 해시의 결과 값을 구할 수 있음
해시 결과 값을 통해 해시를 생성하기 전의 원래 값은 알기 어려움
값이 아주 조금만 다르더라도 해시 결과 값은 무척 상이하게 생성

* 암호화 - 로마시대 암호화 방식
· 밀어내기식 치환 방법
· 기본적인 치환(Substitution) 방식
· 3자씩 알파벳을 밀어내 대응되는 글자로 치환





암호화 구문- Wish to be free from myself
암호화 알고리즘 - 알파벳을 밀어서 대응되는 글자로 치환
암호화 키(Key)- 3

학습내용3 : Salt

해시와 암호화의 약점

· 해시나 암호화로 패스워드 저장해도 같은 패스워드는 같은 해시 값, 같은 암호문으로 저장
· 만일 root 사용자와 wishfree 사용자의 패스워드가 eodlf@!11 이라면 동일하게
E2E783C7C3660CC594BFB35753E454F6 로 저장됨
· 같은 해시 결과나 암호문은 같은 결과만으로도 패스워드를 노출하는 약점이 있음

(안나옴) 이런 상황을 막기 위해 패스워드 해시와 암호화에 사용되는 첨가물의 일종
root 사용자와 wishfree 사용자가 동일한 패스워드(eodlf@!11)를 사용하더라도 Salt를 이용하면 서로 다른 암호화 결과를 저장하게 됨

* Salt와 패스워드를 조합한 값에 대한 MD5 해시 값의 생성 예

계정	Salt	패스워드	Salt+패스워드를 MD5로 해시한 결과 값
root	a2	eodlf@!11	9EF83D5BEF4A7CBC6F7D4940D8447089
wishfree	4F	eodlf@!11	6B796B0DD16C30CCF0B7F02E6457F024

(안나옴)

· Salt와 패스워드 합한'a2eodlf@!11'과'4Feodlf@!11'을 각각 해시한 결과 값은 다름

· 적용된 Salt는 똑같은 패스워드를 숨길 뿐만 아니라 적용 수준에 따라 패스워드 크래킹을 매우 어렵게 만드는 요소

Salt 단점

· 패스워드 파일로 저장 시 MD5 해시 값만 저장 불가능

· 시스템이 패스워드와 어떤 것을 합해 해시를 구한 것인지 알 수 없음

· 패스워드 파일에 저장 시 간단한 인코딩을 통해 해시 결과 값 앞이나 뒤에 Salt 붙임

학습내용4 : 패스워드 크래킹 방법에 대한 이해

1. 패스워드 크래킹 방법에 대한 이해

① 사전 대입 공격
미리 만들어진 패스워드 사전을 이용해서 하나씩 대입하는 방법
② 무작위 대입 공격
문자열 범위에 대해서 생성 가능한 모든 패스워드 생성하여 입력하는 방법
③ 레인보우 테이블을 이용한 공격
변이된 형태의 문자열이 저장된 레인보우 테이블을 이용한 입력하는 방법

2. 사전 대입 공격

사용자가 설정하는 대부분의 패스워드에 특정 패턴이 있는 것을 이용
패스워드로 사용할 만한 것을 사전으로 만들어놓고 하나씩 대입 패스워드 일치 여부 확인

3. 무작위 대입 공격

패스워드에 사용될 수 있는 문자열의 범위 정하고, 그 범위 내에서 생성 가능한 모든 패스워드 생성하여 입력
패스워드가 그다지 복잡하지 않거나 짧을 경우단시간에 크래킹

4. 레인보우 테이블을 이용한 공격

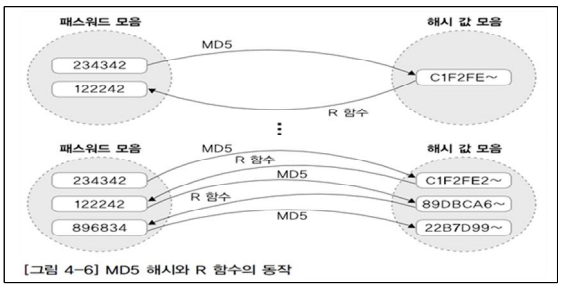
1980년 마틴 헬만이 소개
2000년대에 윈도우의 LM 패스워드를 크래킹할 때 사용
하나의 패스워드에서 시작해 변이된 형태의 여러 패스워드 생성
변이된 각 패스워드의 해시를 고리처럼 연결하여일정 수의 패스워드와 해시로 이루어진 체인(Chain)을 무수히 만들어 놓은 테이블

레인보우 테이블의 기본 개념
· 패스워드별 해시 값을 미리 생성한 후 크래킹하고자하는 해시 값을 테이블에서 검색하여 원래 패스워드를 찾는 것
· 패스워드의 해시 값이 '123452323242'라면 각 해시 테이블에서 미리 구해둔 해시 값 '123452323242'를 찾아 패스워드 '12qw' 찾음
· 가능한 모든 패스워드에 대해서 해시 값을 구해야 하기 때문에 용량이 매우 커질 수 있음

[표] 미리 계산된 해시 테이블

패스워드	해시
1dww	551234523452
12qw	123452323242
21fe	523233452333
df32	234523232345

대용량으로 생성될 수 있는 해시 테이블을 R(Reduction) 함수를 이용해 작은 크기로 줄이는 것
R 함수 : 패스워드로 사용될 수 있는 문자열을 해시 값으로 만드는 함수
[그림] MD5 해시와 R 함수의 동작



[그림 4-6] MD5 해시와 R 함수의 동작

5. 레인보우 테이블을 이용한 공격 예시

최초 패스워드 234342에서 MD 5 해시 값을 3번 구하고, R 함수가 2번 동작
* [표1] 234342의 MD5 해시와 R 함수의 반복 실행 결과

패스워드		MD5 해시
최초 패스워드	234342	C1F2FE224298D6E39EBA607D46F3D9C C
첫 번째 R 함수 동작 결과	122242	89DBCA68BE341E03B5FB59777B93067 E
두 번째 R 함수 동작 결과	896834	22B7D9922C994737D0D9DFCCF6B415 B6

* [표2] 346343의 MD5와 R 함수의 반복 실행 결과

패스워드		MD5 해시
최초 패스워드	346343	A62798B2BFCF406BD76FCBC7A367887 6
첫 번째 R 함수 동작 결과	627982	570727EE4270E0C1A4D8FBB741926DB 8
두 번째 R 함수 동작 결과	570727	86AB6B3355F33F7CD62658FDDA5AF7D 6

* [표3] 898232의 MD5와 R 함수의 반복 실행 결과

패스워드		MD5 해시
최초 패스워드	898232	91CF19DD04A05110A2D2A30D578DDA29
첫 번째 R 함수 동작 결과	911904	3B8635770F22C17E9643441A3E49992E
두 번째 R 함수 동작 결과	386357	E2038DD2A8315D9BF7F72AE5C07530F8

* [표4] 표1~표3들의 값을 이용해 생성한 레인보우 테이블

패스워드	MD5 해시
234342	22B7D9922C994737D0D9DFCCF6B415B6
346343	86AB6B3355F33F7CD62658FDDA5AF7D6
898232	E2038DD2A8315D9BF7F72AE5C07530F8

예시 : 패스워드 해시 값

· 570727EE4270E0C1A4D8FBB741926DB8

① 레인보우 테이블에 크래킹 하려는 해시 값과 같은 MD5 해시 값이 있는지 확인
[표4]에는 570727EE4270E0C1A4D8FBB741926DB8 해시 값 없음

패스워드	MD5 해시
234342	22B7D9922C994737D0D9DFCCF6B415B6
346343	86AB6B3355F33F7CD62658FDDA5AF7D6
898232	E2038DD2A8315D9BF7F72AE5C07530F8

② 레인보우 테이블에 크래킹 하려는 해시 값이 없으면크래킹 할 해시 값에 R 함수 적용 후 패스워드 구하고 다시 해시 값을 구함
570727EE4270E0C1A4D8FBB741926DB8 에 R함수 적용
패스워드 570727 구함
570727 해시 값 86AB6B3355F33F7CD62658FDDA5AF7D6 을 구함

③ 2에서 구한 해시 값 86AB6B3355F33F7CD62658FDDA5AF7D6 이 레인보우 테이블에 있는지 확인
[표4]에서 생성한 레인보우 테이블에86AB6B3355F33F7CD62658FDDA5AF7D6 값 존재

패스워드	MD5 해시
234342	22B7D9922C994737D0D9DFCCF6B415B6
346343	86AB6B3355F33F7CD62658FDDA5AF7D6
898232	E2038DD2A8315D9BF7F72AE5C07530F8

④ 레인보우 테이블에서 확인한 해시 값 발견하면 그 해시 값에 해당하는 최초 패스워드 구함
값이 없다면 같은 해시 값이 나올 때까지 2와 3과정을 해시 테이블 생성 시에 설정한 체인 수만큼 반복
[표 4]에서 86AB6B3355F33F7CD62658FDDA5AF7D6 에 해당하는 패스워드는 346343

패스워드	MD5 해시
234342	22B7D9922C994737D0D9DFCCF6B415B6
346343	86AB6B3355F33F7CD62658FDDA5AF7D6
898232	E2038DD2A8315D9BF7F72AE5C07530F8

⑤ 확인한 최초 패스워드에서 다시 패스워드와 일치하는 해시 값이 나올 때까지 MD5 해시와 R함수 반복 수행
해당 해시 값이 확인되면 찾는 패스워드는 해당 해시 값을 생성한 문자열이 됨

실제 레인보우 테이블의 형태
· 레인보우 체인 2,000개 이상
· 최초 패스워드, 최종 해시 값만 레인보우 테이블에 저장
체인을 2,000개 사용하는 레인보우 테이블에서 해시 값을 10,000개 저장하고 있다면, 레인보우 테이블에서 확인할 수 있는 패스워드의 종류는 20,000,000 (2000*10000)개

【학습정리】

1. 운영체제에서 비밀번호를 숨기는 방법은 해시와 암호화 방법이 있다.

2. 패스워드를 크래킹하는 방법은 사전 대입 공격, 무작위 대입 공격, 레인보우 테이블을 이용한 공격 등이 있다.