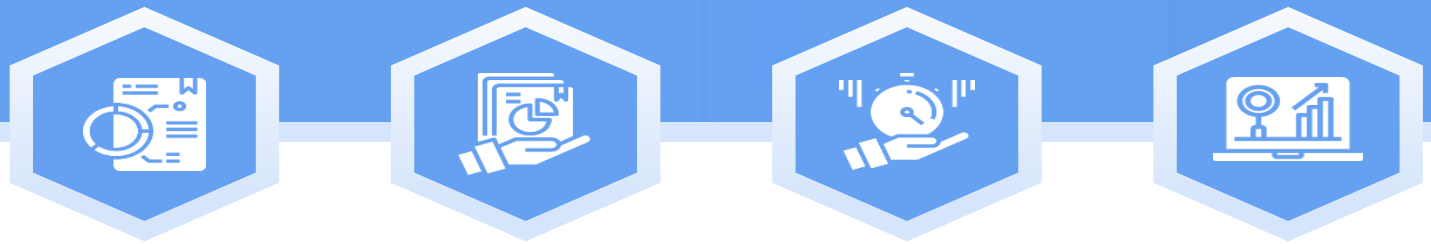
시스템 보안

#3 계정과 권한 - 1



KOREA UNIVERSITY
SEJONG CAMPUS

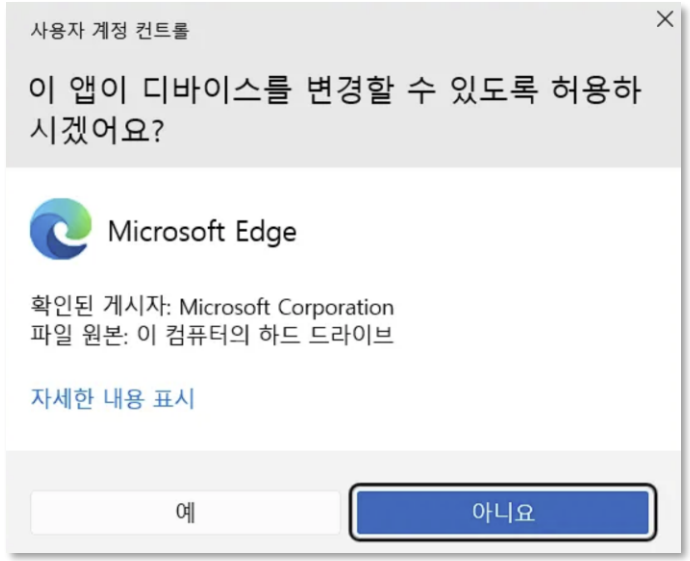
- 목차

  - 권한 상승
  - 해시와 암호화
  - 패스워드 크래킹

권한 상승 (Privilege Escalation)

권한 상승

- **권한 상승(Privilege Escalation)**
  - "권한 상승"은 일반적으로 "권한 상승을 통한 공격"을 의미하는 용어로 많이 사용되지만
  - 원래의 의미로 보면 낮은 수준의 권한을 가진 어플리케이션이 시스템의 설정을 변경하는 등과 같이 더 높은 수준의 권한(관리자 권한 등)이 필요한 작업을 수행할 필요가 있을 때 더 높은 권한을 획득하는 것으로 볼 수 있음



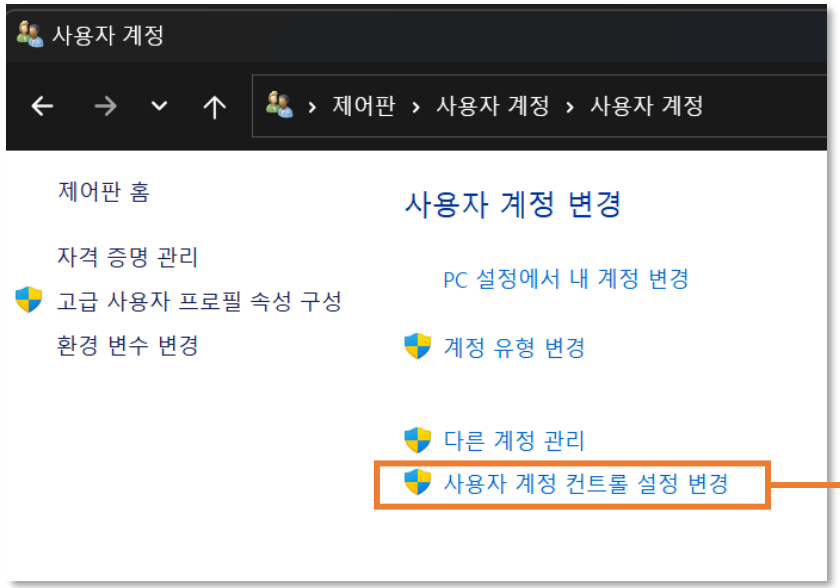
Windows에서의
사용자 계정 컨트롤(UAC, User Access Control)

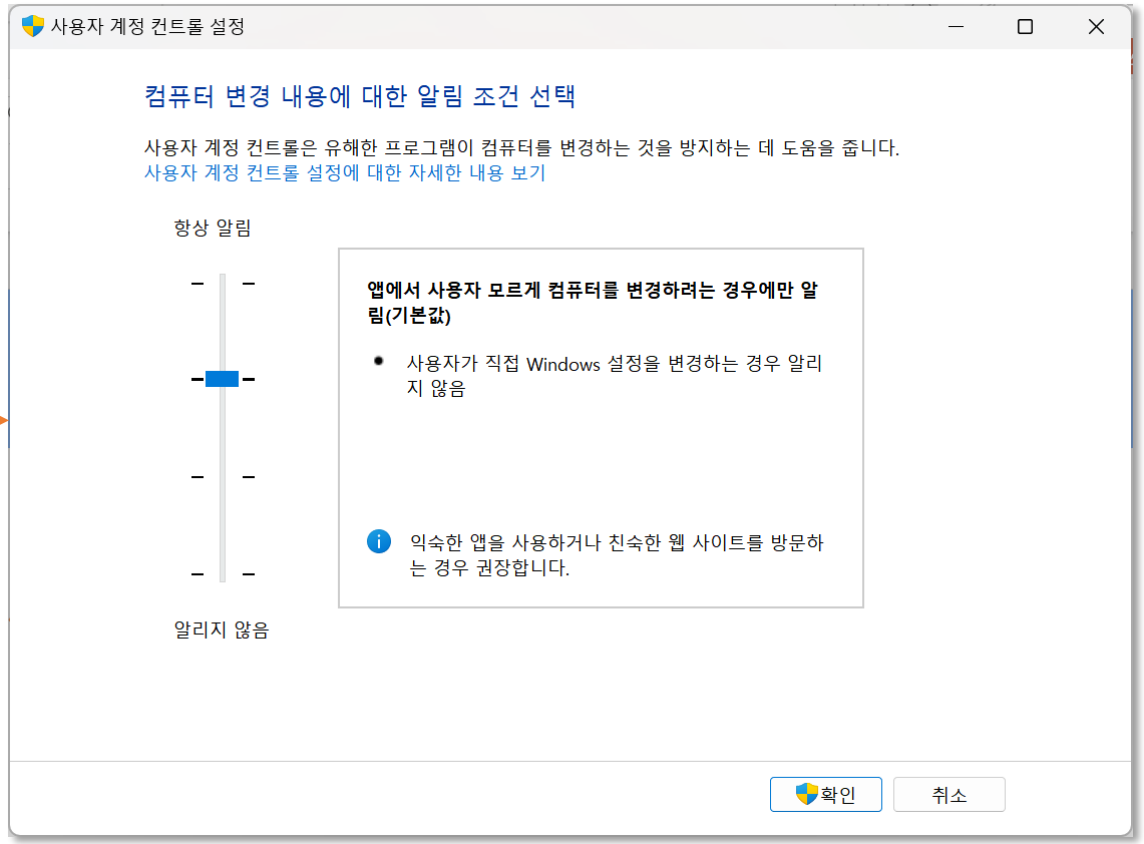
Linux에서의 sudo 명령어

※ sudo : 'Substitute User and do'의 줄임 말로 '다른 사용자 권한으로 실행한다'는 뜻
어원을 고려한 원래 발음은 '수두'이지만 '수도'로도 발음

권한 상승

- **권한 상승(Privilege Escalation)**
  - Microsoft에서는 Windows Vista, Windows Server 2008에서부터 UAC(사용자 계정 컨트롤, User Access Control)이라는 기능을 도입
  - 사용자(관리자)가 권한 수준을 높이는 것을 허용할 때까지 응용 프로그램들은 표준 사용자 권한으로 제한을 둠에 따라 운영 체제의 보안을 개선

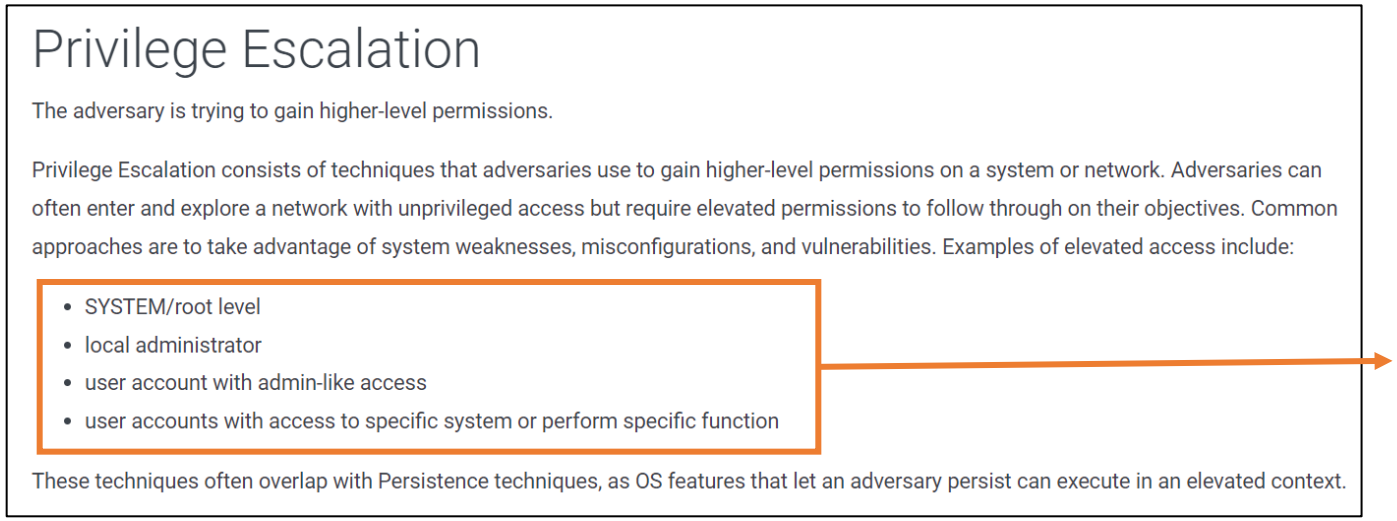




Windows 11 에서의 사용자 계정 컨트롤 설정

권한 상승

- **권한 상승(Privilege Escalation) - MITRE ATT&CK에서 정의하는 권한상승 공격**
  - 권한 상승은 공격자가 시스템이나 네트워크에서 더 높은 수준의 권한을 얻기 위해 사용하는 기술로 구성
  - 공격자는 종종 낮은 권한으로도 네트워크에 진입하고 탐색할 수 있지만 목표를 달성하려면 높은 권한이 필요
  - 일반적인 접근 방식은 시스템의 약점(weakness), 잘못된 구성(misconfiguration) 및 취약점(vulnerability)을 활용



Privilege Escalation

The adversary is trying to gain higher-level permissions.

Privilege Escalation consists of techniques that adversaries use to gain higher-level permissions on a system or network. Adversaries can often enter and explore a network with unprivileged access but require elevated permissions to follow through on their objectives. Common approaches are to take advantage of system weaknesses, misconfigurations, and vulnerabilities. Examples of elevated access include:

- SYSTEM/root level
- local administrator
- user account with admin-like access
- user accounts with access to specific system or perform specific function

These techniques often overlap with Persistence techniques, as OS features that let an adversary persist can execute in an elevated context.

- **높은 액세스 권한의 예**
  - 시스템/루트 수준
  - 로컬 관리자
  - 관리자와 같은 액세스 권한을 가진 사용자 계정
  - 특정 시스템에 접근하거나 특정 기능을 수행하는 사용자 계정

Privilege Escalation :
https://attack.mitre.org/tactics/TA0004/

권한 상승

- **권한 상승(Privilege Escalation) – 취약점을 이용한 공격 사례**
  - 가상머신을 벗어나 하이퍼바이저에 대한 액세스 권한을 획득할 수 있는 취약점

VM웨어 툴즈에서 발견된 로컬 권한 상승 취약점, 고위험군으로 분류돼

분류:업계뉴스 | **날짜**:2022-09-01 11:00:03 | **조회수**:4

원문 : http://www.boannews.com/media/view.asp?idx=109345&page=9&kind=1

기업들 사이에서 널리 사용되는 VM웨어 솔루션 중 하나에서 고위험군 취약점이 발견됐다. 패치 외에는 위험을 완화시킬 방법이 없다. 다행히 VM웨어는 여태까지 그러했듯 재빠르게 패치를 개발해 배포 중에 있다.
[보안뉴스 문가용 기자] 가상화 기술 전문 업체인 VM웨어(VMware)의 솔루션에서 고위험군 취약점이 발견됐다. 이 취약점을 익스플로잇 할 경우 공격자는 로컬에서 권한 상승을 일으킬 수 있고, 가상기계를 완전 장악하는 것도 가능하게 된다고 한다. 때문에 기업의 중요한 데이터가 위협 받을 수 있게 된다.

권한 상승

- **권한 상승(Privilege Escalation) – 취약점을 이용한 공격 사례**
  - Anti-Virus 어플리케이션으로 인해 관리자 권한을 획득할 수 있는 취약점

CVE-2021-26624 | eScan Anti-Virus 로컬 권한 상승 취약점 2022.03.31

□ 개요

o MicroWorld Technologies社의 eScan Anti-Virus for Linux에서 setuid가 걸린 프로그램의 인자를 적절히 검증하지 않아 로컬 권한 상승이 가능한 취약점

취약점 종류	영향	심각도	CVSS 점수	CVE-ID
부적절한 입력 값 검증	로컬 권한 상승	High	7.8	CVE-2021-26624

□ 설명

o eScan Anti-Virus에는 setuid 바이너리를 찾는 명령어 runasroot가 존재하는데, 이 명령어가 root로 명령어를 실행하는 조건 중 하나인 'check_argv가 1을 return 하는 경우'를 이용하여 root 권한으로 명령어를 실행할 수 있음

o check_argv는 argv[1]이 chmod 이거나, argv[3]이 "wget-op","mwav.conf" 등의 명령어 중에 하나이면 1을 return

o 이러한 검증 로직의 미흡함과 chmod 명령어가 여러 인자를 주면 그 인자들에 해당하는 파일의 권한까지 다 바꿀 수 있음을 이용하여 root 권한 탈취 가능

권한 상승

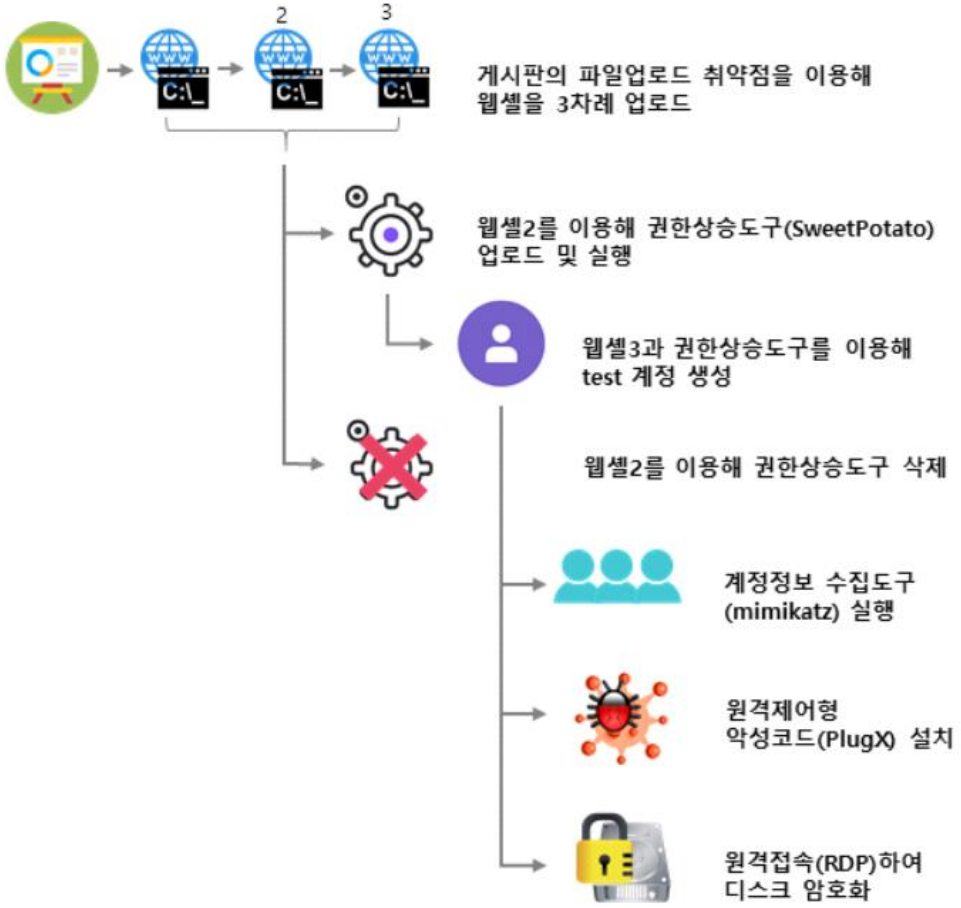
- **권한 상승(Privilege Escalation) – 취약점을 이용한 공격 사례**
  - 지속적으로 발견되는 취약점

o 취약점 요약 정보

제품 카테고리	CVE 번호	CVE 제목
YARP reverse proxy	CVE-2022-26924	YARP 서비스 거부 취약성
Visual Studio Code	CVE-2022-26921	Visual Studio Code 권한 상승 취약성
Microsoft Graphics Component	CVE-2022-26920	Windows 그래픽 구성 요소 정보 유출 취약성
LDAP - Lightweight Directory Access Protocol	CVE-2022-26919	Windows LDAP 원격 코드 실행 취약성
Windows Fax Compose Form	CVE-2022-26918	Windows 팩스 작성 양식 원격 코드 실행 취약성
Windows Fax Compose Form	CVE-2022-26917	Windows 팩스 작성 양식 원격 코드 실행 취약성
Windows Fax Compose Form	CVE-2022-26916	Windows 팩스 작성 양식 원격 코드 실행 취약성
Windows schannel	CVE-2022-26915	Windows 보안 채널 서비스 거부 취약성
Windows Win32K	CVE-2022-26914	Win32k 권한 상승 취약성
Skype for Business	CVE-2022-26911	비즈니스용 Skype 정보 유출 취약성
Skype for Business	CVE-2022-26910	비즈니스용 Skype 및 Lync 스푸핑 취약성
Azure SDK	CVE-2022-26907	Azure SDK for .NET 정보 공개 취약성
Windows User Profile Service	CVE-2022-26904	Windows 사용자 프로필 서비스 권한 상승 취약성
Microsoft Graphics Component	CVE-2022-26903	Windows 그래픽 구성 요소 원격 코드 실행 취약성
Microsoft Office Excel	CVE-2022-26901	Microsoft Excel 원격 코드 실행 취약성
Azure Site Recovery	CVE-2022-26898	Azure Site Recovery 원격 코드 실행 취약성

권한 상승

- 권한 상승(Privilege Escalation) – 취약점을 이용한 공격 사례
  - 실제 발생했던 공격 사례



- ‘비트라커(BitLocker)’ 랜섬웨어의 경우 공격자는 기업 홈페이지 고객상담 게시판에 파일 첨부 기능을 이용해 시스템 정보 유출과 내부 취약점 서비스 스캔 등의 기능을 하는 웹쉘을 업로드한 후 웹쉘이 실행될 때 시스템 권한을 획득하여 시스템 계정을 무단으로 생성하고 원격으로 접속, 윈도우에 기본 탑재된 디스크 암호화 기능인 비트라커(BitLocker)을 이용해 디스크를 암호화

권한 상승

- ## 권한 상승(Privilege Escalation) – 대응 방안

  - 운영체제와 애플리케이션 개발자들은 특정 종류의 취약점에 대한 공격을 방지하고 공격이 발생했을 때 피해를 제한하기 위해서 다양한 기술을 연구

  - 최소 권한 원칙(least-privilege principles)
  - 제로 트러스트 아키텍처(zero-trust architectures)
  - 애플리케이션 샌드박싱(application sandboxing)
  - 커널 메모리 영역 분리
  - 가상화 및 컨테이너화
  - 단일 애플리케이션을 마이크로서비스(microservices)로 분리
  - .
  - .
  - .

해시와 암호화

해시와 암호화

- **해시(Hash)와 암호화(Encryption)**

  - 운영체제는 보안을 위해서 어떤 형식이든 패스워드와 같은 데이터를 보호해야 하는데, 기본적으로 해시와 암호화를 사용

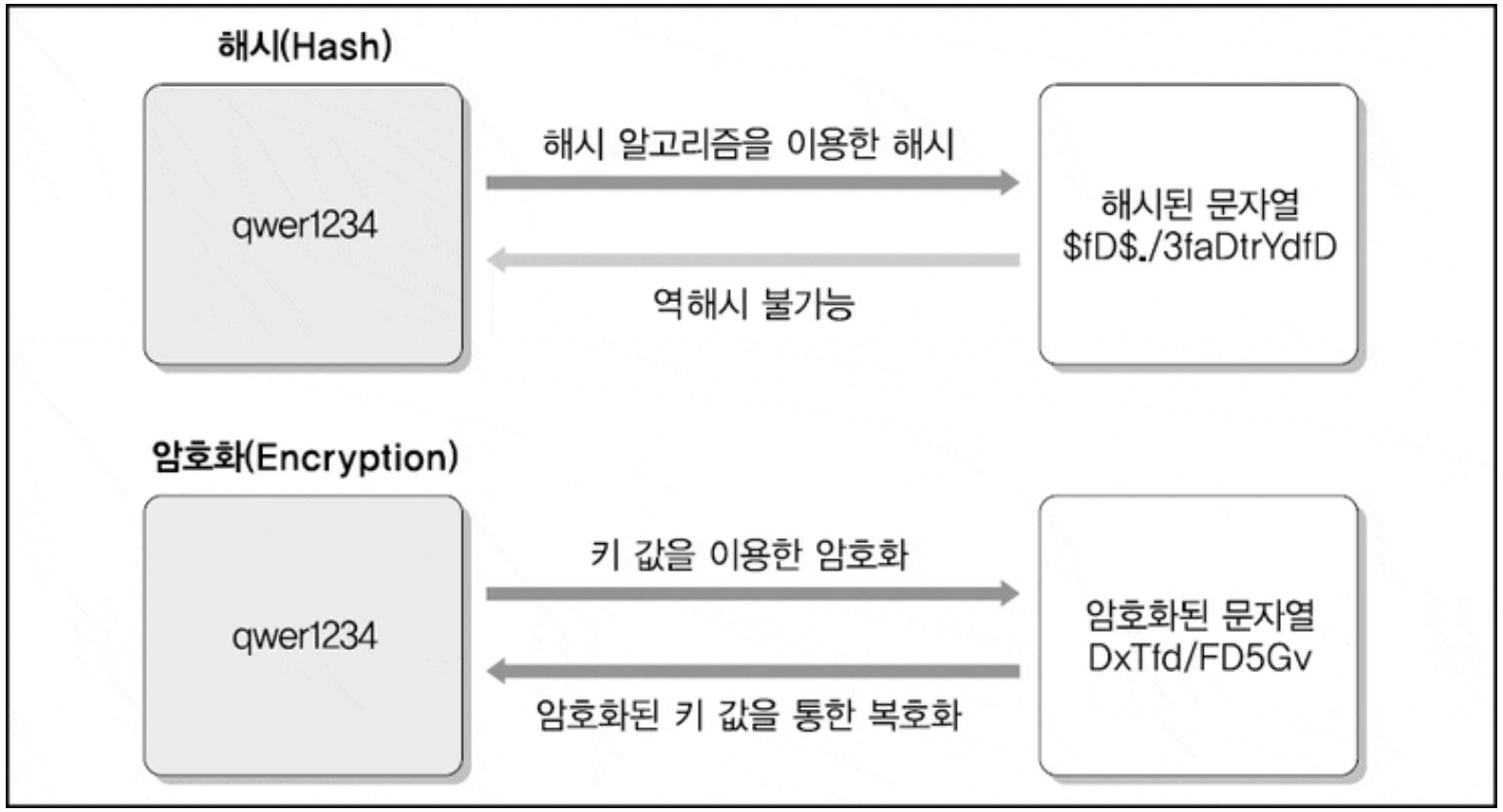
  - 해시는 다양한 길이를 가진 임의의 데이터로부터 고정된 크기의 데이터(해시 값)을 만들어 내는 것을 의미
    해시 값을 만들어 내기 위해서는 데이터를 자르고 치환하거나 위치를 바꾸는 등 방법을 사용하는데 이를 수행하는 것이 해시 함수
    만약 해시 값이 다르면 그 해시 값을 만들어 낸 원래 데이터는 달라야 하며, 해시 값에서 원래의 데이터를 구하는 것도 불가능해야 함

  - 암호화는 승인된 사람을 제외하고는 누구든지 원본 데이터를 읽을 수 없도록 변환하는 것을 의미
    암호화 알고리즘과 키를 사용하여 데이터를 변경하는 방법으로 어떤 알고리즘과 키를 사용했는지를 알고 있으면 복호화를 통해서
    원본 데이터를 복원할 수 있음

  - 해시와 암호화는 모두 패스워드와 같은 원본 데이터를 효과적으로 숨기는 기술이지만
    암호화가 원본 데이터를 복원할 수 있는 반면 해시는 복원이 불가능하다는 차이점이 있음

해시와 암호화

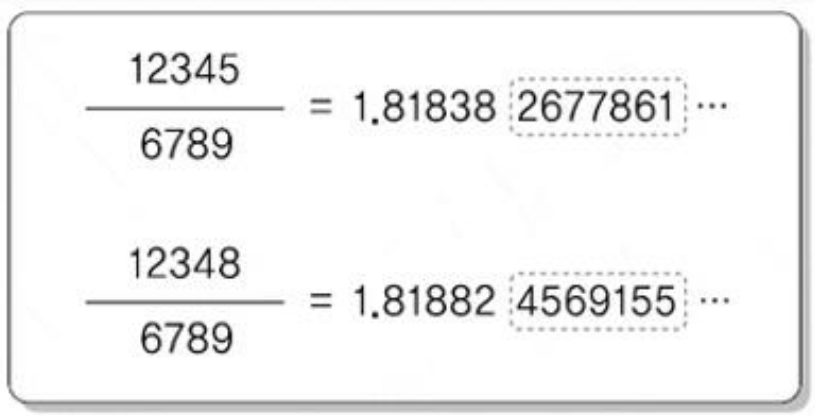
- 해시(Hash)와 암호화(Encryption)



해시와 암호화

- **해시(Hash)**

  - 나눗셈을 이용한 해시 값 획득



① 123456789라는 수와 이것과 한 자리 수만 다른 123486789이라는 수를 가정

② 두 수를 가운데를 기준으로 둘로 나누고 큰 수를 작은 수로 나누면 결과 값이 각각 1.818...이 나오게 되는데, 앞 6자리 숫자를 버리고 나머지 값을 해시 값으로 취함

③ 123456789의 해시 값은 2677861, 123486789의 해시 값은 4569155

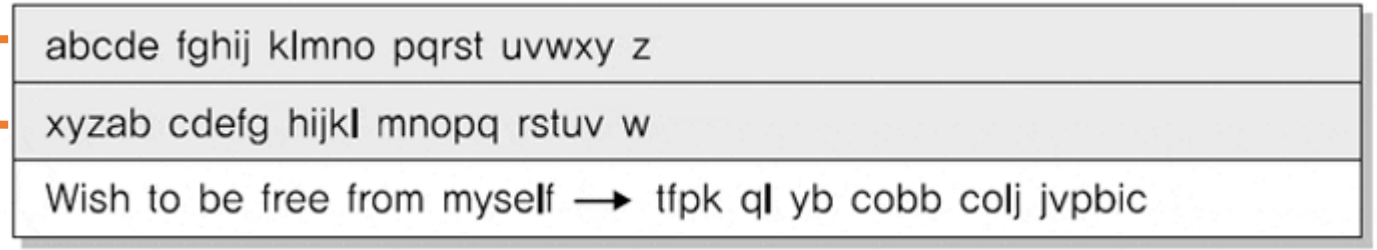
- 두 해시 값만으로 해시 전의 원래 수를 알아내는 것은 불가능에 가까운데, 로직을 알더라도 버려진 1.81838과 1.81882를 알아낼 수 없기 때문

- 이처럼 해시는 로직을 알고 있을 경우 해시의 결과 값을 구하기가 쉽지만 그 해시 결과 값을 통해 해시를 생성하기 전의 원래 값은 알기 어렵고 값이 아주 조그만 다르더라도 해시 결과 값은 무척 상이하게 생성 됨

해시와 암호화

- **암호화(Encryption)**
  - 로마시대에 사용된 암호화 방식인 차환(Substitution) 방식은 3자 씩 알파벳을 밀어내 대응되는 글자로 치환하는 방식으로 암호화
  - 이 암호화는 어떤 글을 암호화했는지 쉽게 알아볼 수 없지만, 알파벳을 3자 씩 밀어서 대응했다는 것을 알면 복호화 가능

※ '알파벳을 밀어서 대응되는 글자로 치환'하는 것이 암호화 알고리즘
'3'이라는 숫자는 암호화 키



해시와 암호화

- **Salt**
  - 기본적으로 패스워드는 그대로 저장되는 것이 아니라 해시나 암호화를 통해서 저장하는데,
    예를 들어, root 사용자와 wishfree 사용자의 패스워드가 똑같이 'eodlf@!11'라고 가정하고 저장하면 어떻게 저장될까?
  - 해시를 사용하든 암호화를 사용하든 똑같은 패스워드는 똑같은 해시 값이나 똑같은 암호문으로 저장되게 될 것이기 때문에
    똑같은 해시 결과나 암호문으로 인한 패스워드를 노출을 방지하기 위한 방법이 필요
  - Salt는 이런 상황을 막기 위해 패스워드 해시와 암호화에 사용되는 첨가물의 일종으로 운영체제별로 다양한 알고리즘이 존재

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

계정	Salt+MD5
root	a29EF83D5BEF4A7CBC6F7D4940D8447089
wishfree	4F6B796B0DD16C30CCF0B7F02E6457F024

- Salt와 패스워드를 합한 'a2eodlf@!11'과 '4Feodlf@!11'을 각각 해시 한 결과 값

- 패스워드를 저장할 때는 해시 된 값에 Salt 정보를 붙여서 저장
  (시스템이 패스워드와 어떤 것을 합해 해시를 구한 것인지 알 수 없기 때문)

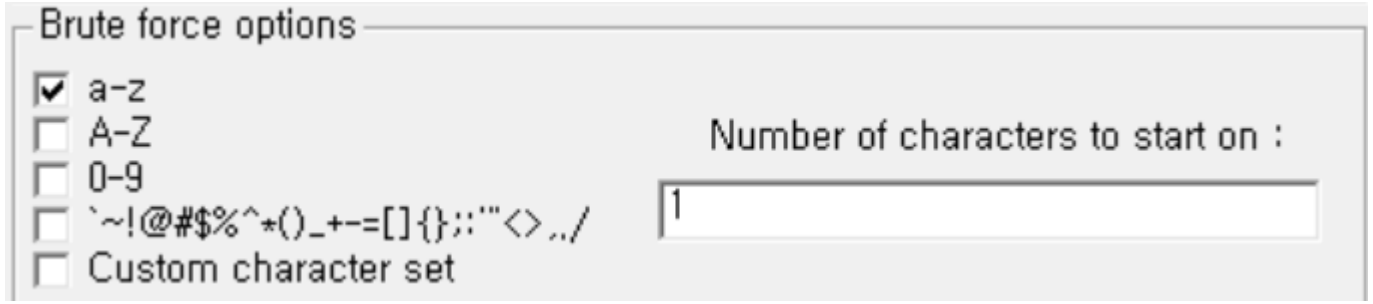
패스워드 크래킹

패스워드 크래킹

- **패스워드 크래킹의 방법**

  - 사전 대입 공격(Dictionary Attack) :
    사용자가 설정하는 대부분의 패스워드에 특정 패턴이 있음에 착안하여
    가능한 조합을 사전으로 만들어 놓고 하나씩 대입하여 확인

  - 무작위 대입 공격(Brute Force Attack) :
    패스워드에 사용될 수 있는 문자열의 범위 정하고, 그 범위 내에서 생성 가능한 모든 패스워드 생성하여 입력
    패스워드가 그다지 복잡하지 않거나 짧을 경우 단시간에 크래킹



패스워드 크래킹

- ## 패스워드 크래킹의 방법

  - 레인보우 테이블 공격 (Rainbow Table Attack)

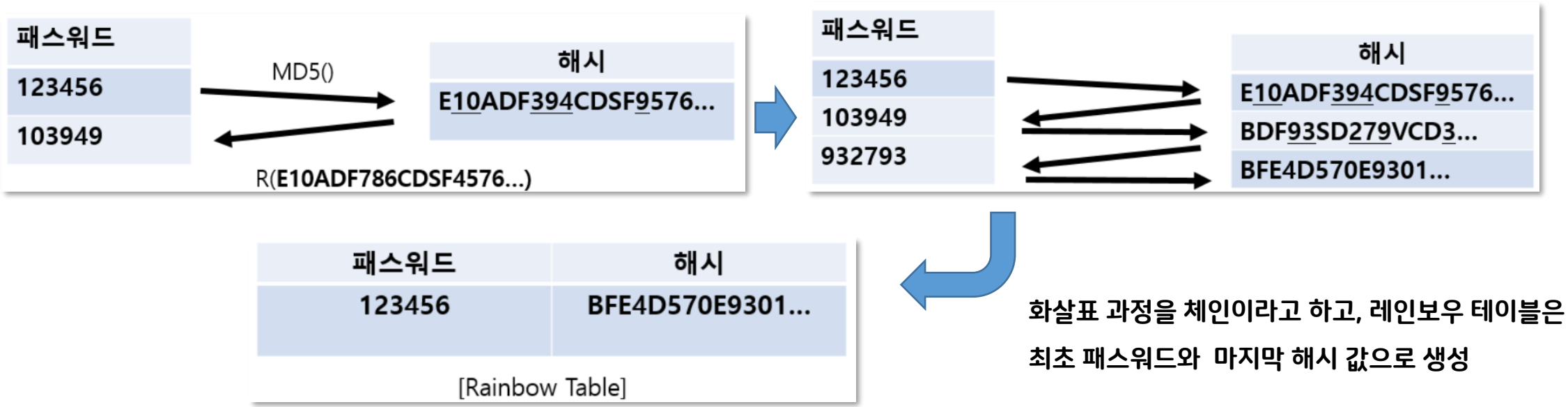
- 레인보우 테이블은 해시 함수(MD5, SHA-1, SHA-2 등)을 사용하여 변환 가능한 모든 해시 값을 저장시켜 놓은 표를 의미하며
  보통 해시 함수를 이용하여 저장된 비밀번호로부터 원래의 비밀번호를 추출해 내는데 사용됨
  실제로는 모든 경우의 수를 다 저장할 수는 없기 때문에 영어 소문자와 숫자 조합으로 일정 길이까지의 문자열에 대해서 계산하는 등의 방법을 사용

- 1980년 마틴 헬만(Martin Hellman)에 의해 소개, MD5 암호화가 쉽게 복호화 될 수 있다는 것을 보여준 해킹기법 중 하나로
  2000년대에 윈도우의 LM 패스워드를 몇 분 만에 크래킹 하며 유명해짐

- 레인보우 테이블의 가장 기본적인 개념 :
  해시는 원본 데이터 복원이 불가능하기 때문에 가능한 경우의 해시 값들을 미리 만들어 놓고 검색해서 찾는 개념으로
  그냥 가능한 해시 값들을 모두 다 저장하려면 저장 공간이 너무 많이 필요하기 때문에 계산 시간이 들더라도 저장 공간을 압도적으로 줄일 수 있는
  레인보우 테이블을 이용

패스워드 크래킹

- **패스워드 크래킹의 방법**
  - 레인보우 테이블을 R(Reduction) 함수와 해시 함수를 이용해서 최초 입력으로 부터 지정된 수 만큼의 체인을 만들어 나가고 최종적으로는 처음 입력과 마지막 해시 값 만을 저장 함으로써 저장 공간을 줄이는 방법을 사용
  - 최초 입력되는 패스워드로 부터 MD5 해시 값을 추출하고, 이 값에 R 함수(MD5 해시 값 중 앞의 6개 숫자만 뽑아낸다고 가정)를 적용해서 또다른 무작위 패스워드를 추출하는 과정을 반복



패스워드 크래킹

- **패스워드 크래킹의 방법**
  - 레인보우 테이블의 생성 예시

패스워드		MD5 해시
최초 패스워드	234342	C1F2FE224298D6E39EBA607D46F3D9CC
첫 번째 R 함수	122242	89DBCA68BE341E03B5FB59777B93067E
두 번째 R 함수	896834	22B7D9922C994737D0D9DFCCF6B415B6

패스워드		MD5 해시
최초 패스워드	346343	A62798B2BFCF406BD76FCBC7A3678876
첫 번째 R 함수	627982	570727EE4270E0C1A4D8FBB741926DB8
두 번째 R 함수	570727	86AB6B3355F33F7CD62658FDDA5AF7D6

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

패스워드	MD5 해시
234342	22B7D9922C994737D0D9DFCCF6B415B6
346343	86AB6B3355F33F7CD62658FDDA5AF7D6
898232	E2038DD2A8315D9BF7F72AE5C07530F8

생성된 레인보우 테이블 (세 개의 체인으로 구성)

패스워드 크래킹

- **패스워드 크래킹의 방법**

  - 해시 값 '570727EE4270E0C1A4D8FBB741926DB8'의 패스워드 찾기

패스워드	MD5 해시
234342	22B7D9922C994737D0D9DFCCF6B415B6
346343	86AB6B3355F33F7CD62658FDDA5AF7D6
898232	E2038DD2A8315D9BF7F72AE5C07530F8

생성된 레인보우 테이블

① Rainbow Table에 존재 하지 않음

② '570727EE4270E0C1A4D8FBB741926DB8'에 R 함수 적용
(R 함수는 MD5 해시 값 중 앞의 6개 숫자만 뽑아내는 함수로 가정)
추출된 6개의 숫자 570727에 MD5 적용
결과 : '86AB6B3355F33F7CD62658FDDA5AF7D6'

③ Rainbow Table에 해시 값 '86AB6B3355F33F7CD62658FDDA5AF7D6'가 존재
레인보우 테이블에 존재한다고 해서 패스워드가 '346343'은 아니고 과정 속에 존재한다는 뜻

패스워드 크래킹

- **패스워드 크래킹의 방법**

  - 해시 값 '570727EE4270E0C1A4D8FBB741926DB8'의 패스워드 찾기

패스워드	MD5 해시
234342	22B7D9922C994737D0D9DFCCF6B415B6
346343	86AB6B3355F33F7CD62658FDDA5AF7D6
898232	E2038DD2A8315D9BF7F72AE5C07530F8

생성된 레인보우 테이블

패스워드		MD5 해시
최초 패스워드	346343	A62798B2BFCF406BD76FCBC7A3678876
첫 번째 R 함수	627982	570727EE4270E0C1A4D8FBB741926DB8
두 번째 R 함수	570727	86AB6B3355F33F7CD62658FDDA5AF7D6

'346343' 패스워드 체인

④ Rainbow Table에 존재 시 초기 패스워드 값으로 MD5->R함수 체인 반복으로 찾고자 하는 해시 값의 패스워드를 찾을 수 있음

⑤ 체인에서 검색을 통해서 '570727EE4270E0C1A4D8FBB741926DB8'의 패스워드는 '627982'라는 것을 알 수 있음

패스워드 크래킹

- **패스워드 크래킹의 방법**
  - 레인보우 테이블로 패스워드 찾기

- 만약 2,000개의 해시를 저장한 체인 10,000개를 레인보우 테이블에 저장하고 있다면?
  10,000개 항목이 저장된 레인보우 테이블로 20,000,000(2,000 * 10,000) 개의 해시를 확인 할 수 있음

- 비밀번호는 일반적으로 일반 텍스트 형식이 아닌 해시 값으로 저장되기 때문에 해시된 비밀번호 데이터베이스가 공격자의 손에 들어간 경우 사전 계산된 레인보우 테이블을 사용하여 비밀번호를 복구할 수 있음

- 이 공격에 대한 일반적인 방어 방법은 각 비밀번호를 해시하기 전에 각 비밀번호에 “Salt(솔트)"를 추가하여 해시를 계산함으로써 레인보우 테이블 조회만으로는 패스워드를 복구할 수 없도록 함

패스워드 크래킹

- **크리덴셜 스터핑 공격(Credential Stuffing Attack)**

- 공격자가 여러 가지의 경로로 수집한 사용자들의 로그인 인증 정보(Credential)를 공격 대상 사이트에 대입(Stuffing)하여 정상적으로 로그인이 되는지를 확인하는 형태의 공격을 의미하며, 과거 여러 해 동안의 데이터 침해 사건을 통해 유출된 수십억 개의 로그인 정보를 이용

- 공격자가 탈취된 인증 정보를 이용한 일종의 사전 대입 공격으로도 볼 수 있지만, 일반적인 단어를 조합하여 만든 '사전'을 이용하는 것이 아닌 유출된 유효한 인증 정보를 이용한다는 점에서 차이가 있음

- 여러 웹사이트에 같은 비밀번호를 재사용하는 사람이 많기 때문에 보안성이 낮은 웹사이트에서 훔친 인증 정보가 민감한 데이터를 가진 서비스를 제공하는 웹사이트에서도 유효할 수 있다는 점과 정상적인 로그인 과정을 거치기 때문에 공격 대상 사이트의 정보보호 관리자들이 해킹된 사실을 인지하지 못하는 경우가 많다는 점에서 위험성이 매우 큰 공격 방식

Q A