1 OWASP TOP 10

1 OWASP TOP 10

연도별 OWASP TOP 10(2007~2013)(비영리단체)

OWASP TOP 10	2007년	2010년	2013년
A1	크로스 사이트 스크립팅(XSS)	인젝션 취약점	인젝션 취약점
A2	인젝션 취약점	크로스 사이트 스크립팅(XSS)	인증 및 세션 관리 취약점
A3	악성 파일 실행	취약한 인증 및 세션 관리	크로스 사이트 스크립팅(XSS)
A4	불안전한 직접 객체 참조	안전하지 않은 직접 객체 참조	취약한 직접 객체 참조

1 OWASP TOP 10

연도별 OWASP TOP 10

OWASP TOP 10	2007년	2010년	2013년
A5	크로스 사이트 요청 변조(CRSF)	크로스 사이트 요청 변조(CRSF)	보안 설정 오류
A6	정보 유출과 부적절한 에러 처리	보안상 잘못된 구성	민감한 데이터 노출
A7	취약한 인증 및 세션관리	안전하지 않은 암호 저장	기능 수준의 접근 통제 누락
A8	불안정한 암호화 저장	URL 접근 제한 실패	크로스 사이트 요청 변조(CRSF)

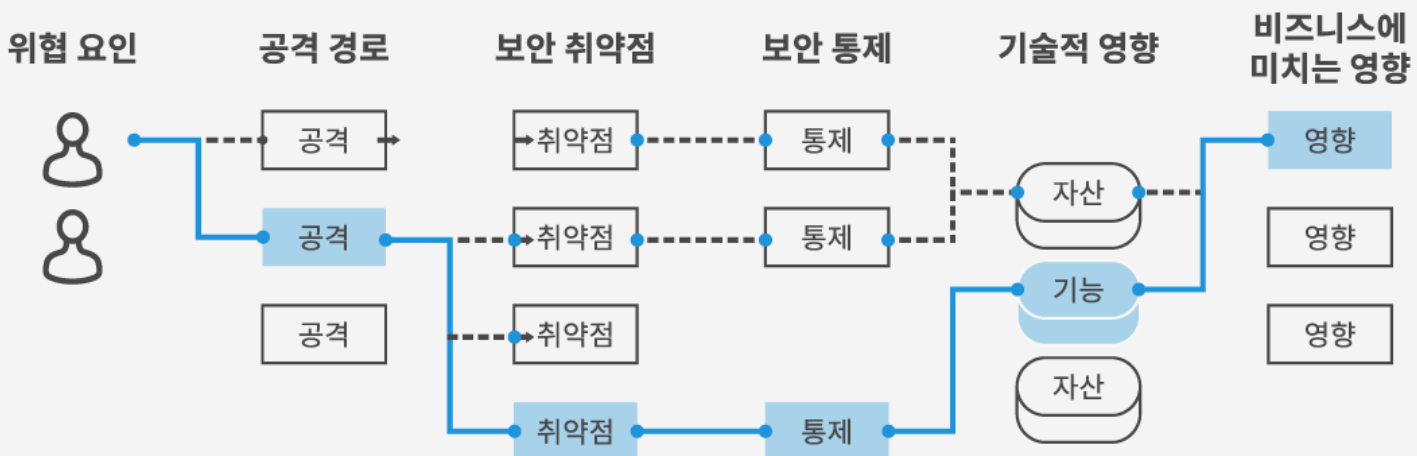
1 OWASP TOP 10

연도별 OWASP TOP 10

OWASP TOP 10	2007년	2010년	2013년
A9	불안전한 통신	불충분한 전송 계층 보호	알려진 취약점이 있는 컴포넌트 사용
A10	URL 접근 제한 실패	검증되지 않은 리다이렉트 및 포워드	검증되지 않은 리다이렉트 및 포워드

1 OWASP TOP 10

웹 애플리케이션의 취약점이 기업의 비즈니스에 미치는 영향



1 OWASP TOP 10

2013년의 각 요소별 평가

OWASP Top 10-2013년	공격 용이도	보안 취약점	기술적 영향
A1 인젝션 취약점	쉬움	취약점 분포 : 보통 탐지 용이도 : 평균	심각
A2 인증 및 세션 관리 취약점	평균	취약점 분포 : 광범위 탐지 용이도 : 평균	심각
A3 크로스 사이트 스크립팅(XSS)	평균	취약점 분포 : 매우 광범위 탐지 용이도 : 쉬움	중간
A4 취약한 직접 객체 참조 (사용자 계정 정보)	쉬움	취약점 분포 : 보통 탐지 용이도 : 쉬움	중간

1 OWASP TOP 10

2013년의 각 요소별 평가

OWASP Top 10-2013년	공격 용이도	보안 취약점	기술적 영향
A5 보안 설정 오류 (디렉토리 리스팅)	쉬움	취약점 분포 : 보통 탐지 용이도 : 쉬움	중간
A6 민감한 데이터 노출 (신용카드 정보)	어려움	취약점 분포 : 드묾 탐지 용이도 : 평균	심각
A7 기능 수준의 접근 통제 누락 (접근 통제 검사)	쉬움	취약점 분포 : 보통 탐지 용이도 : 평균	중간
A8 크로스 사이트 요청 변조(CSRF)	평균	취약점 분포 : 보통 탐지 용이도 : 쉬움	중간

1 OWASP TOP 10

2013년의 각 요소별 평가

OWASP Top 10-2013년	공격 용이도	보안 취약점	기술적 영향
A9 알려진 취약점이 있는 컴포넌트 사용(취약한 컴포넌트)	평균	취약점 분포 : 광범위 탐지 용이도 : 어려움	중간
A10 검증되지 않은 리다이렉트 및 포워드(피싱 사이트)	평균	취약점 분포 : 드묾 탐지 용이도 : 쉬움	중간

1 OWASP TOP 10

2013년의 각 요소별 평가

A1. 인젝션 취약점

- 웹 애플리케이션에서 내부 데이터베이스와 연동한 결과를 웹 애플리케이션에 보내주는 부분에 주로 발생
- 주로 SQL 쿼리문을 대상으로 하지만 OS나 LDAP도 악용될 수 있음(SQL Injection)

1 OWASP TOP 10

- 2013년의 각 요소별 평가

A2. 인증 및 세션 관리 취약점

- 웹 애플리케이션이 사용자에 대한 인증 및 세션 처리를 잘못했을 때 발생

A3. 크로스 사이트 스크립팅(XSS)

- 해당 웹 사이트를 방문하는 사용자를 공격하는 기법
- 해당 웹 사이트 방문자의 쿠키나 세션을 추출할 수 있고, 방문자를 악성 코드가 포함된 사이트로 리다이렉트 할 수도 있음(Cookie)

1 OWASP TOP 10

- 2013년의 각 요소별 평가

A4. 취약한 직접 객체 참조

- 인증 및 세션 관리 취약점과 유사하지만 특정 객체를 직접 참조할 때 발생한다는 점이 다름
- 예)URL 뒤에 다른 사용자의 아이디를 입력하여 사용자 계정 정보를 볼 수 있음

```
http://example.com/app/
accountInfo?acct=notmyacct
```

1 OWASP TOP 10

- 2013년의 각 요소별 평가

A5. 보안 설정 오류

- 애플리케이션을 설치할 때 생기는 기본 페이지를 삭제하지 않고 그냥 방치하거나 최신 보안 패치를 업데이트하지 않는 경우에 발생
- 디렉터리 리스팅과 같은 애플리케이션 설정 오류도 포함

1 OWASP TOP 10

- 2013년의 각 요소별 평가

A6. 민감한 데이터 노출

- 신용카드 정보, 개인 식별 정보, 인증 정보와 같은 중요한 데이터를 제대로 보호하지 않을 때 발생

A7. 기능 수준의 접근 통제 누락

- 애플리케이션에서 각 기능에 접근하는 서버에 동일한 접근 통제 검사를 수행할 경우 적절한 확인을 하지 않으면 공격자는 권한 없이 요청 위조 가능

1 OWASP TOP 10

2013년의 각 요소별 평가

A8. 크로스 사이트 요청 변조(CSRF)

- 요청 변조

  - 공격자가 입력한 스크립트를 웹 사이트 방문자가 실행함으로써 해당방문자의 권한으로 스크립트가 실행되도록 만드는 방식

1 OWASP TOP 10

2013년의 각 요소별 평가

A8. 크로스 사이트 요청 변조(CSRF)

On-Site 요청 변조

- 한 사이트 내에서 요청 변조가 발생하는 취약점

VS

Cross-Site 요청 변조

- 웹 사이트에 악성 스크립트를 입력한 후 해당 사이트 방문자가 악성 스크립트에 의해 특정 행동(개인 정보 수정, 회원 탈퇴 등)을 실행 하도록 만드는 것

1 OWASP TOP 10

- 2013년의 각 요소별 평가

A9. 알려진 취약점이 있는 컴포넌트 사용

- 취약한 컴포넌트를 악용하여 공격하면 데이터 손실이나 서버 장악 문제가 발생할 수 있음

1 OWASP TOP 10

2013년의 각 요소별 평가

A10. 검증되지 않은 리다이렉트 및 포워드

- 목적 페이지를 결정할 때 신뢰되지 않은 데이터를 사용하거나 해당 사이트에 대한 적절한 검증이 없을 경우 피싱 사이트나 악성 코드가 포함된 사이트로 리다이렉트 하거나 권한 없는 페이지로의 접근을 위해 포워드를 사용할 수 있음

1 OWASP TOP 10

- 2013년의 각 요소별 평가

A10. 검증되지 않은 리다이렉트 및 포워드

- 예) 특정 페이지를 리다이렉트한다면 공격자는 해당 페이지를 악성 코드가 포함된 사이트로 리다이렉트 할 수 있음

```
http://www.example.com/
redirect.jsp?url=evilsite.com
```

1 OWASP TOP 10

2013, 2017 OWASP TOP 10 비교

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – 인젝션	→	A1:2017 – 인젝션
A2 – 취약한 인증과 세션 관리	→	A2:2017 – 취약한 인증
A3 – 크로스 사이트 스크립팅 (XSS)	↘	A3:2017 – 민감한 데이터 노출
A4 – 안전하지 않은 직접 객체 참조 [A7 항목과 병합됨]	∪	A4:2017 – XML 외부 개체 (XXE) [신규]
A5 – 잘못된 보안 구성	↘	A5:2017 – 취약한 접근 통제 [합침]
A6 – 민감한 데이터 노출	↗	A6:2017 – 잘못된 보안 구성
A7 – 기능 수준의 접근 통제 누락 [A4 항목과 병합됨]	∪	A7:2017 – 크로스 사이트 스크립팅 (XSS)
A8 – 크로스 사이트 요청 변조 (CSRF)	☒	A8:2017 – 안전하지 않은 역직렬화 [신규, 커뮤니티]
A9 – 알려진 취약점이 있는 구성요소 사용	→	A9:2017 – 알려진 취약점이 있는 구성요소 사용
A10 – 검증되지 않은 리다이렉트 및 포워드	☒	A10:2017 – 불충분한 로깅 및 모니터링 [신규, 커뮤니티]

※출처 : https://www.owasp.org/images/b/bd/OWASP_Top_10-2017-ko.pdf

1 OWASP TOP 10

OWASP TOP 10(2017)

A1. 인젝션

- SQL, OS, XXE, LDAP 인젝션 취약점은 신뢰할 수 없는 데이터가 명령어나 쿼리문의 일부분으로써, 인터프리터로 보내질 때 발생
- 공격자의 악의적인 데이터는 예기치 않은 명령을 실행하거나 올바른 권한 없이 데이터에 접근하도록 인터프리터를 속일 수 있음(SQL Injection)

1 OWASP TOP 10

OWASP TOP 10(2017)

A2. 취약한 인증

- 인증 및 세션 관리와 관련된 애플리케이션 기능이 종종 잘못 구현되어 공격자들이 암호, 키, 세션 토큰을 위험에 노출시킬 수 있거나 일시적 또는 영구적으로 다른 사용자의 권한 획득을 위해 구현상 결함을 악용하도록 허용

1 OWASP TOP 10

OWASP TOP 10(2017)

A3. 민감한 데이터 노출

- 다수의 웹 애플리케이션과 API는 금융 정보, 건강 정보, 개인 식별 정보와 같은 중요한 정보를 제대로 보호하지 않음
- 공격자는 신용카드 사기, 신분 도용 또는 다른 범죄를 수행하기 위해 보호가 취약한 데이터를 훔치거나 수정 가능
- 중요한 데이터는 저장 또는 전송할 때 암호화 같은 추가 보호조치가 없으면 탈취 당할 수 있으며, 브라우저에서 주고 받을 대 각별한 주의가 필요

1 OWASP TOP 10

OWASP TOP 10(2017)

A4. XML 외부 개체(XXE)

- 오래되고, 설정이 엉망인 많은 XML 프로세서들은 XML 문서 내에서 외부 개체 참조를 평가
- 외부 개체는 파일 URL 처리기, 내부 파일 공유, 내부 포트 스캔, 원격코드 실행과 서비스 거부 공격을 사용하여 내부 파일을 공개하는데 사용할 수 있음

1 OWASP TOP 10

OWASP TOP 10(2017)

A5. 취약한 접근 통제

- 인증된 사용자가 수행할 수 있는 작업에 대한 제한이 제대로 적용되어 있지 않음
- 공격자는 이러한 결함을 악용하여 다른 사용자의 계정에 접근하거나 중요한 파일을 보거나 다른 사용자의 데이터를 수정하거나 접근 권한을 변경하는 등 권한 없는 기능과 데이터에 접근할 수 있음

1 OWASP TOP 10

OWASP TOP 10(2017)

A6. 잘못된 보안 구성

- 잘못된 보안 구성은 가장 흔하게 보이는 이슈
- 취약한 기본 설정, 미완성(또는 임시 설정), 개방된 클라우드 스토리지, 잘못 구성된 HTTP 헤더 및 민감한 정보가 포함된 장황한 에러 메시지로 인한 결과임
- 모든 운영체제, 프레임워크, 라이브러리와 애플리케이션을 안전하게 설정해야 할 뿐만 아니라 시기 적절하게 패치/업그레이드를 진행해야 함 (가장 기본)

1 OWASP TOP 10

OWASP TOP 10(2017)

A7. 크로스 사이트 스크립팅(XSS)

- XSS 취약점은 애플리케이션이 올바른 유효성 검사 또는 필터링 처리 없이 새 웹 페이지에 신뢰할 수 없는 데이터를 포함하거나 자바스크립트와 HTML을 생성하는 브라우저 API를 활용한 사용자 제공 데이터로 기존 웹 페이지를 업데이트할 때 발생

1 OWASP TOP 10

OWASP TOP 10(2017)

A7. 크로스 사이트 스크립팅(XSS)

- 피해자의 브라우저에서 공격자에 의해 스크립트를 실행시켜 사용자 세션을 탈취할 수 있게 만들고, 웹사이트를 변조시키고, 악성 사이트로 리다이렉션 할 수 있도록 허용

1 OWASP TOP 10

OWASP TOP 10(2017)

A8. 안전하지 않은 역직렬화

- 역직렬화 : 일련의 바이트로부터 데이터 구조를 추출(직렬화)
- 종종 원격 코드 실행으로 이어짐
- 역직렬화 취약점이 원격 코드 실행 결과를 가져오지 않더라도 이는 권한 상승 공격, 주입 공격과 재생 공격을 포함한 다양한 공격 수행에 사용될 수 있음

1 OWASP TOP 10

OWASP TOP 10(2017)

A9. 알려진 취약점이 있는 구성요소 사용

- 라이브러리, 프레임워크 및 다른 소프트웨어 모듈 같은 컴포넌트는 애플리케이션과 같은 권한으로 실행
- 만약 취약한 컴포넌트가 악용된 경우, 심각한 데이터 손실을 일으키거나 서버가 장악됨
- 알려진 취약점이 있는 컴포넌트를 사용한 애플리케이션과 API는 애플리케이션 방어를 약화시키거나 다양한 공격과 영향을 주게 함(중성화)

1 OWASP TOP 10

OWASP TOP 10(2017)

A10. 불충분한 로깅 & 모니터링

- 사고 대응의 비효율적인 통합 또는 누락과 함께 공격자들이 시스템을 더 공격하고, 지속성을 유지하며, 더 많은 시스템을 중심으로 공격할 수 있도록 만듦
- 데이터를 변조, 추출 또는 파괴할 수 있음
- 대부분의 침해 사례에서 침해를 탐지하는 시간이 200일이 넘게 걸리는 것을 보여주고, 일반적으로 내부 프로세스와 모니터링보다 외부 기관이 탐지 (실시간)

2 웹 애플리케이션의 취약점

2 | 웹 애플리케이션의 취약점

1 Webgoat

- 웹 응용 프로그램 보안을 위해 대화형 웹 보안 환경을 구성하고, JAVA 기반의 보안 벤치마킹 플랫폼, 허니팟으로 확장한 개념(Honeypot)
- 이러한 플랫폼을 이용하여 웹 해킹을 실습해보고, 취약점 연구를 해볼 수 있음

2 | 웹 애플리케이션의 취약점

1 Webgoat

- 웹 애플리케이션의 취약점을 알고 보안적으로 조치를 할 수 있는 공부환경을 제공
- 가장 초기 버전은 OWASP 취약점을 포함한 웹 페이지를 시작으로 현재 8버전을 제공하고 있음
- 약 50개 이상의 취약점을 포함하고 있으며 자바로 작성되어 어떠한 OS에서도 사용이 가능

2 | 웹 애플리케이션의 취약점

1 Webgoat

- 개발자와 비개발자 모드로 나뉨

비개발자

- 단순히 문제 위주로 풀게 됨

VS

개발자

- 소스 코드를 고칠 수 있음
- 비개발자 모드에 없는 문제도 풀 수 있음

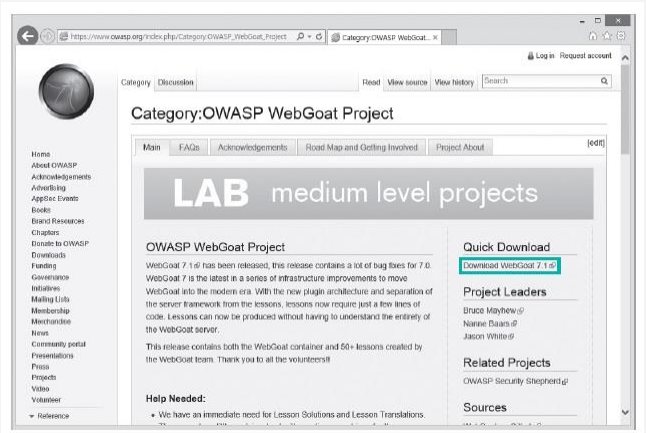
2 Webgoat 설치하기(현재 : 8.0)

실습 환경

- 윈도우 기반의 운영체제
- 필요 프로그램 : WebGoat 7.1

2 Webgoat 설치하기

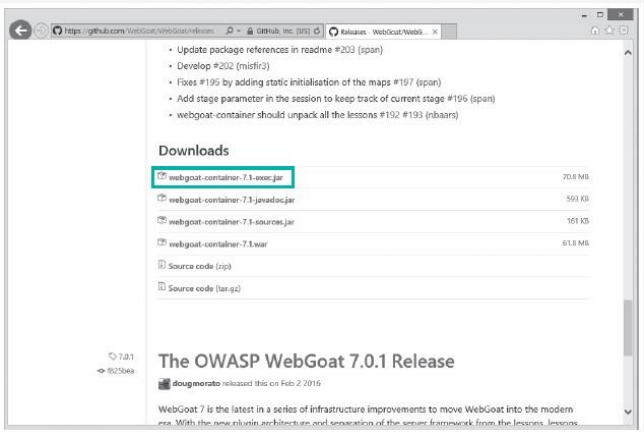
① WebGoat 사이트 접속



https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

2 Webgoat 설치하기

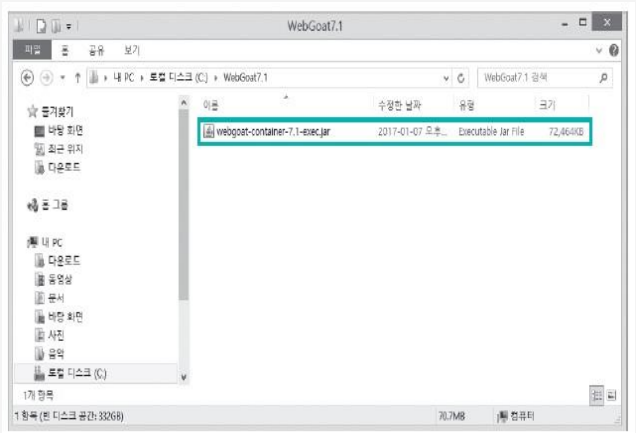
② WebGoat 다운로드



https://github.com/WebGoat/WebGoat/releases

2 Webgoat 설치하기

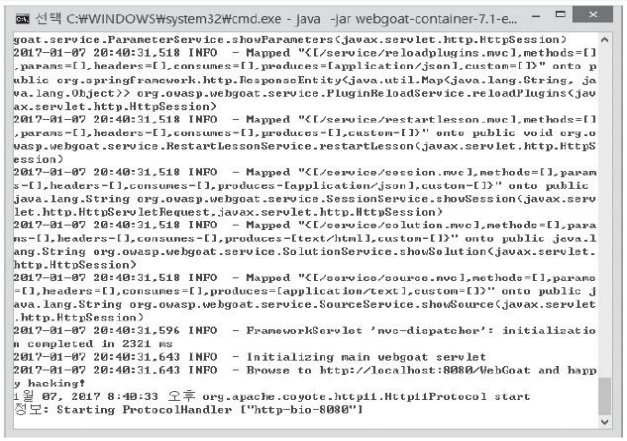
③ WebGoat 실행 파일 복사



C:₩WebGoat7.1 폴더를 만든 후
내려 받은 파일을 해당 폴더에 복사

2 Webgoat 설치하기

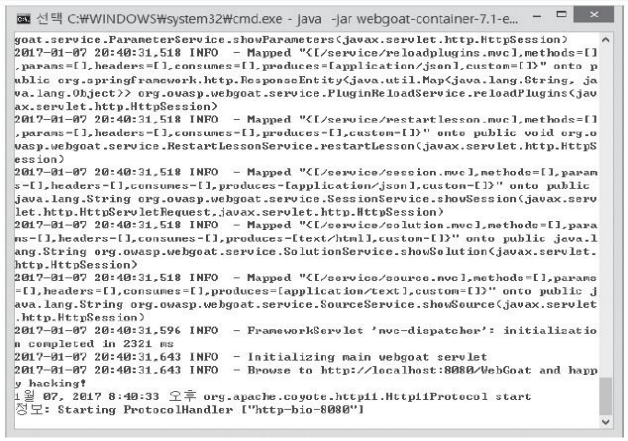
④ WebGoat 실행



도스창에서 C:₩WebGoat7.1 폴더로 이동하여 다음 명령어 입력 java -jar webgoat-container-7.1-exec.jar

2 Webgoat 설치하기

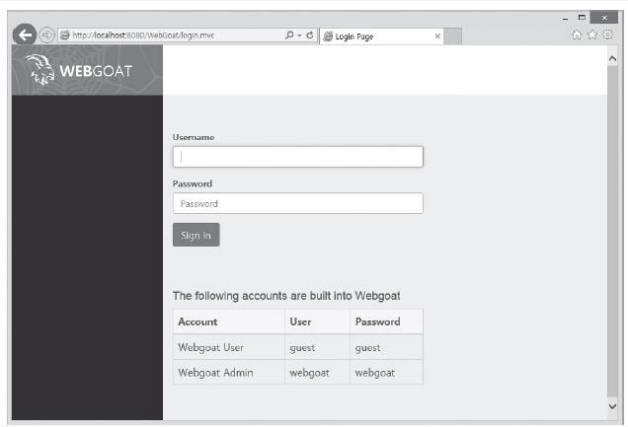
④ WebGoat 실행



맨 마지막에 다음과 같은 메시지가 나오면 제대로 실행된 것
INFO - Browse to http://localhost:8080/WebGoat and happy hacking!

2 Webgoat 설치하기

⑤ 브라우저를 통해 WebGoat 접속



'http://localhost:8080/WebGoat' 입력
연습 목적이므로 Username과 Password에 동일하게 guest 입력 후 로그인

3 암호알고리즘과 암호프로토콜

1 대칭 암호와 비대칭 암호

- 대칭 암호(symmetric cryptography)
  - 암호화를 할 때 사용하는 키와 복호화 할 때 사용하는 키가 동일한 암호알고리즘

- 비대칭 암호(asymmetric cryptography)
  - 암호화를 할 때 사용하는 키와 복호화를 할 때 사용하는 키가 서로 다른 암호알고리즘
  - 공개키 암호

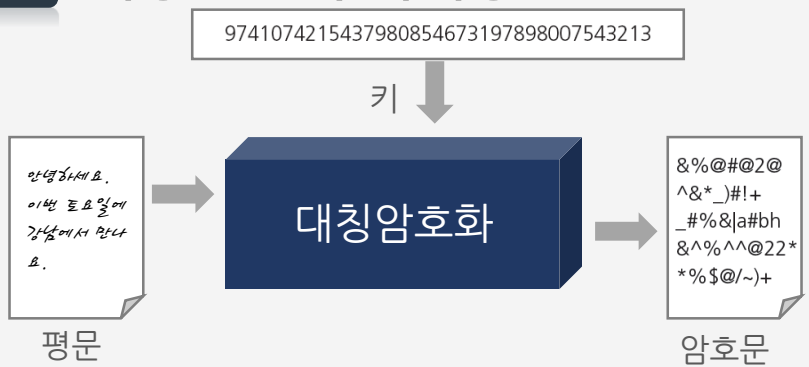
1 대칭 암호와 비대칭 암호

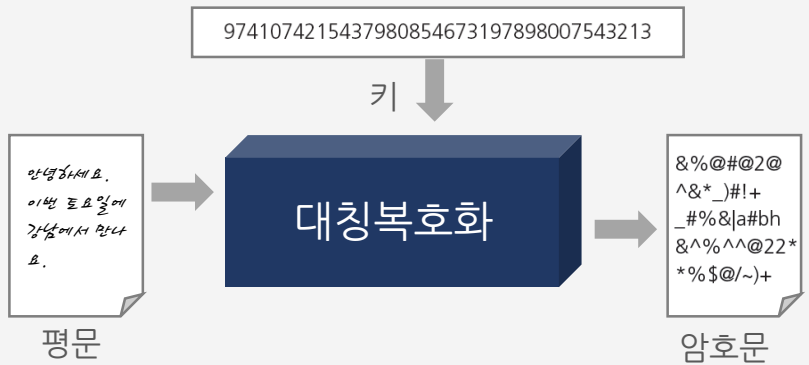
- 비대칭 암호 요소
  - 송신자: 한 쌍의 키
  - 수신자: 한 쌍의 키
    - 이 한 쌍의 키: (공개키, 개인키)

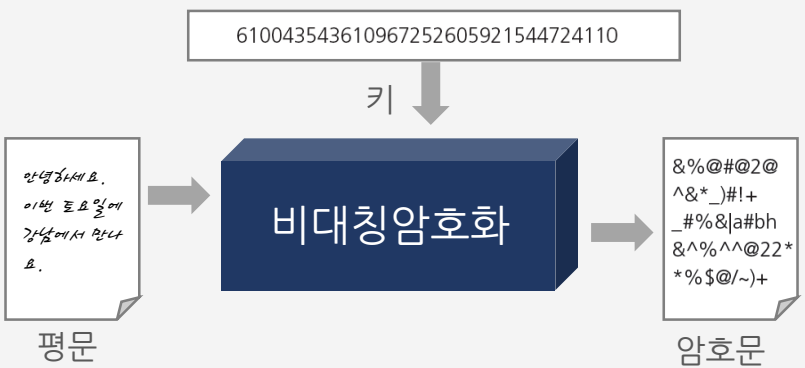
    - 이 두 개의 키는 수학적으로 밀접한 연관
    - 연관되어 있지만 유추할 수는 없다!

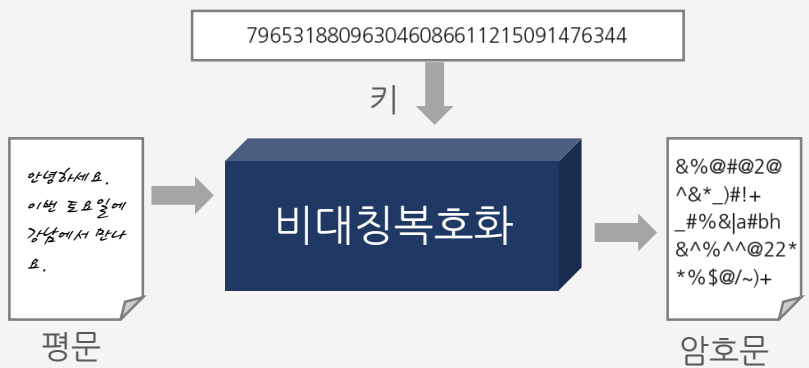
- 공개키 암호(public-key cryptography)이름
  - 공개키는 공개를 하므로 이 암호 알고리즘을 공개키 암호라 함

1 대칭 암호와 비대칭 암호

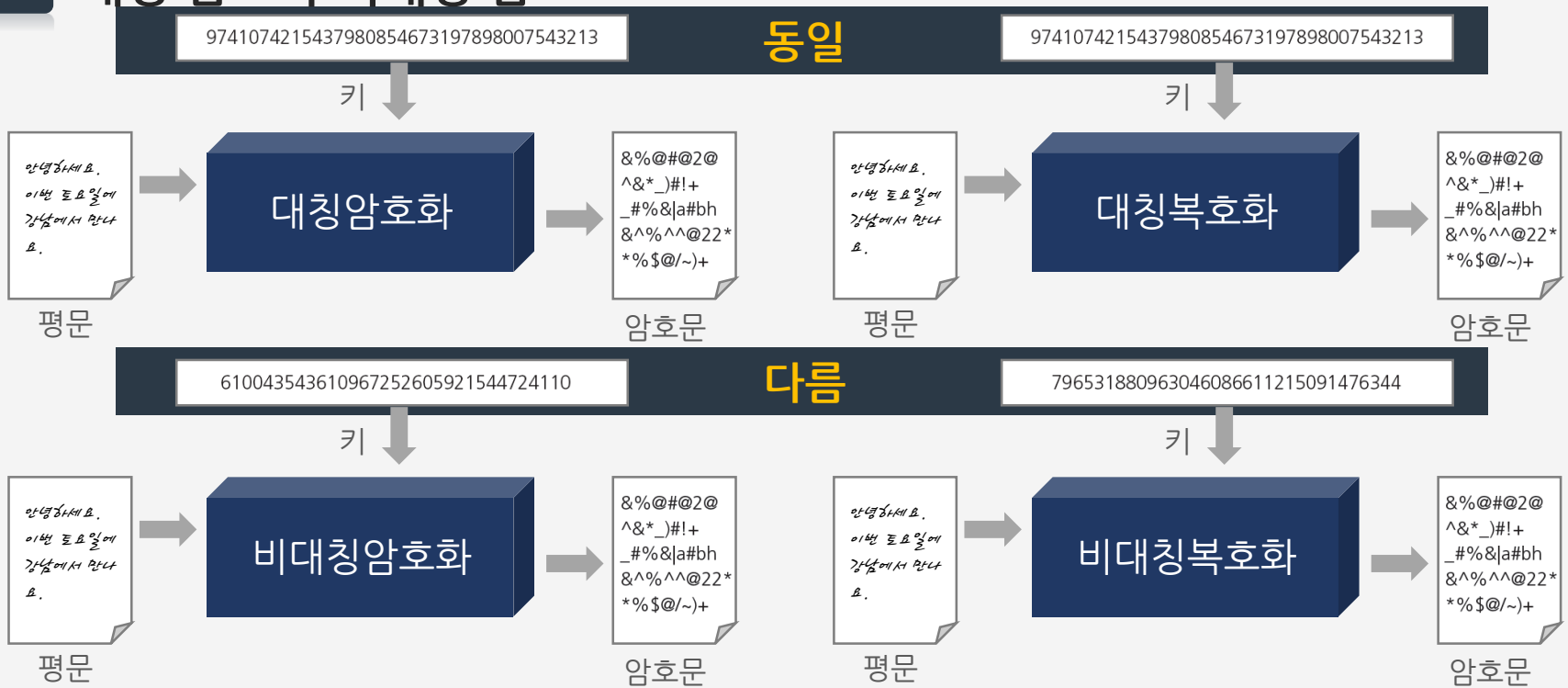








1 대칭 암호와 비대칭 암호



2 하이브리드 암호 시스템

- 하이브리드 암호 시스템(hybrid cryptosystem)
  - 대칭 암호와 공개 키 암호를 조합한 암호방식
  - 대칭 암호와 공개 키 암호의 장점을 조합

3 일방향 해시 함수

- 해시값이란 일방향 해시함수(one-way hash function)를 사용하여 계산한 값
  - 일방향: A → B는 되나, B → A는 불가능
- 문서의 기밀성이 아니라, 무결성(integrity) 점검
  - 무결성: 메시지가 위조되었는가?

4 메시지 인증 코드

- 메시지 인증 코드(MAC: message authentication code)
  - 메시지가 생각했던 통신 상대로부터 온 것임을 확인하는 코드

5 디지털 서명

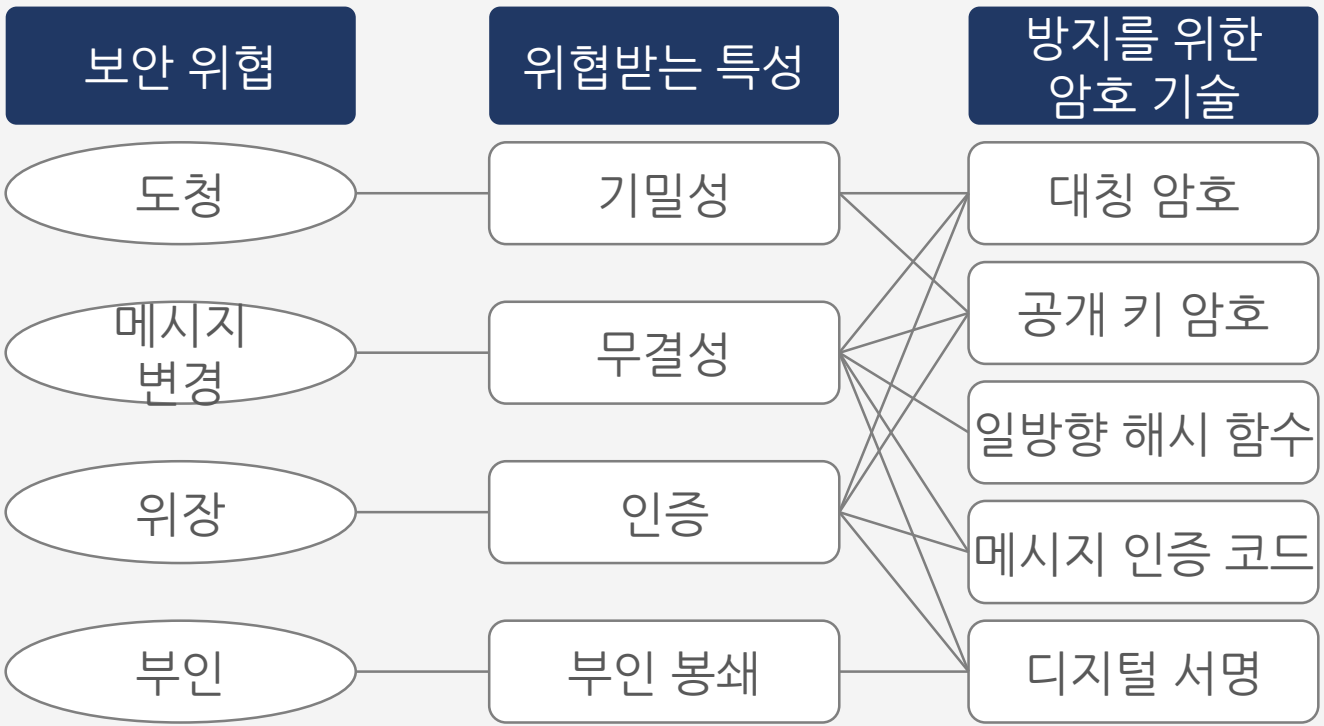
- 디지털 서명(digital signature)
  - 거짓 행세, 변경, 부인같은 위협을 방지하는 기술

- 부인(repudiation)
  - 통신 사실을 나중에 아니라고 하는 것

6 의사 난수 생성기

- 의사 난수 생성기(pseudo random number generator; PRNG)는 난수열을 생성하는 알고리즘
  - 의사 난수 vs. 실제 난수
  - Key를 만들 때 사용!

7 보안 위협과 암호 기술에 의한 방지



7 스테가노그래피와 디지털 워터마킹

- 크립토그래피(cryptography)
  - 메시지의 내용을 읽지 못하게 하는 기법
- 스테가노그래피(steganography)
  - 메시지의 내용을 읽지 못하게 하는 것이 아니라, 메시지의 존재 자체를 숨기는 기법
  - 메시지를 숨겨 넣는 방법을 알게 되면 메시지의 내용은 금방 노출
  - 예: 전체가 메시지가 아니라 숨겨논 일부분만이 메시지

7 스테가노그래피와 디지털 워터마킹

- 디지털 워터마킹(digital watermarking)
  - 파일 중에 저작권자나 구입자의 정보를 집어넣는 기술
- 디지털 워터마킹 기술에 스테가노그래피 사용
  - 워터 마킹을 숨김

8 알고리즘 사용시 유의사항

- 비밀 암호 알고리즘을 사용하지 말 것
  - 비밀 암호 알고리즘을 만들어서 사용할 것이 아니라, 공개되어 있는 강한 암호 알고리즘을 사용해야 함

  - 암호알고리즘의 비밀은 반드시 폭로: 만들어 사용하면 쉽게 깨짐!
  - 강한 암호 알고리즘을 만드는 것은 매우 어려움
  - 만들 수도 없지만 만든다 하더라도 쉽게 깨짐!

8 알고리즘 사용시 유의사항

- 숨기는 것에 의한 보안(security by obscurity)
  - 암호 알고리즘을 비밀로 해서 보안을 유지하려고 하는 행위
  - 전문가가 볼 때 위험하고, 어리석은 행위로 간주: 숨기는 것 자체는 보안이 될 수 없다!
  - 예: 비밀은 언젠가 밝혀지고, 암호 알고리즘이 약하다면…

8 알고리즘 사용시 유의사항

- 약한 암호는 암호화 하지 않는 것보다 위험
  - 사용자는 암호의 강도와는 상관없이 '암호화되어 있다'는 사실만으로 안심하는 경향
  - 약한 암호를 사용하려면, 처음부터 암호 따위를 사용하지 않는 것이 좋음
  - 예: 암호화가 안되어 있는데, 암호화되어 있다고 생각하면…

8 알고리즘 사용시 유의사항

- 어떤 암호라도 언젠가는 해독
  - 모든 키를 하나도 빠짐없이 시도해 봄으로서 언젠가는 반드시 해독: brute-force attack
  - 암호문이 해독되기까지 시간과, 암호를 사용하여 지키고 싶은 평문의 가치와의 밸런스(tradeoff)가 중요

너무 복잡하게 암호화를 하면?

8 알고리즘 사용시 유의사항

- 일회용 패드(one-time pad)
  - 절대로 해독되지 않는 암호 알고리즘: 해독을 해도 제대로 했는지 알 수가 없다!
  - 현실적으로 활용하기에는 적합하지 않은 암호: 일회용 패드를 안전하게 전송할 수 있다면 평문도 안전하게 전송할 수 있다!

8 알고리즘 사용시 유의사항

- 암호는 보안의 아주 작은 부분
  - 사회공학적인 공격 방법은 암호의 강도 그 자체와는 아무런 관계가 없음

    사장님이 메일을 보내면…

  - 보안 시스템의 강도는 보안 시스템을 구성하는 여러 링크 중 가장 약한 링크의 강도와 같음

    가장 약한 링크는 암호가 아니라 사람