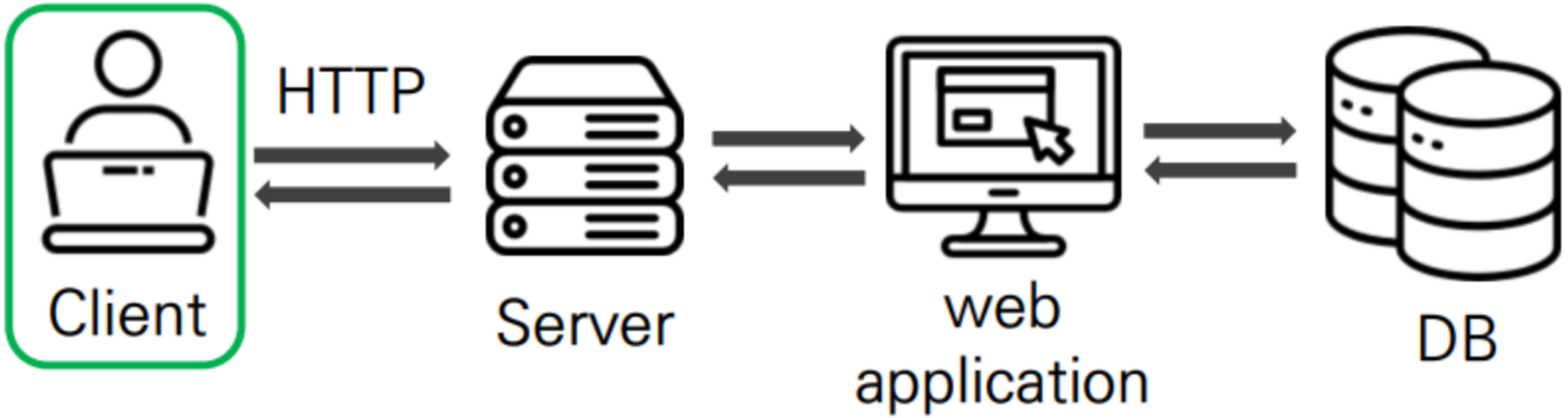
Web hacking

week3

SWING 31기| STUDY

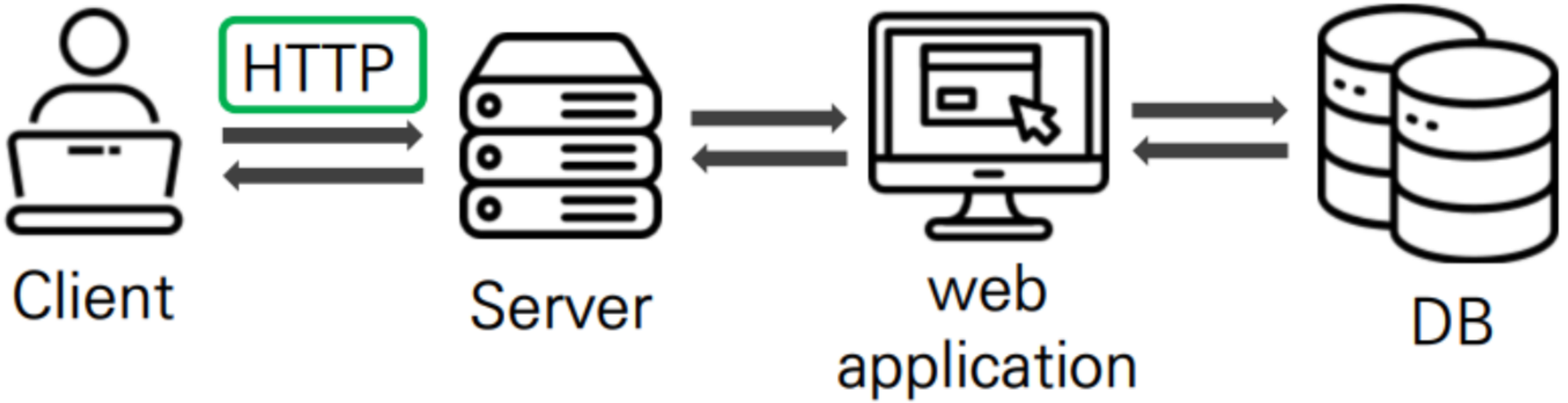
웹 구성요소: Client



1.웹 클라이언트(Web Client)

필요한 데이터를 웹 서버에 요청(request)하는 주체

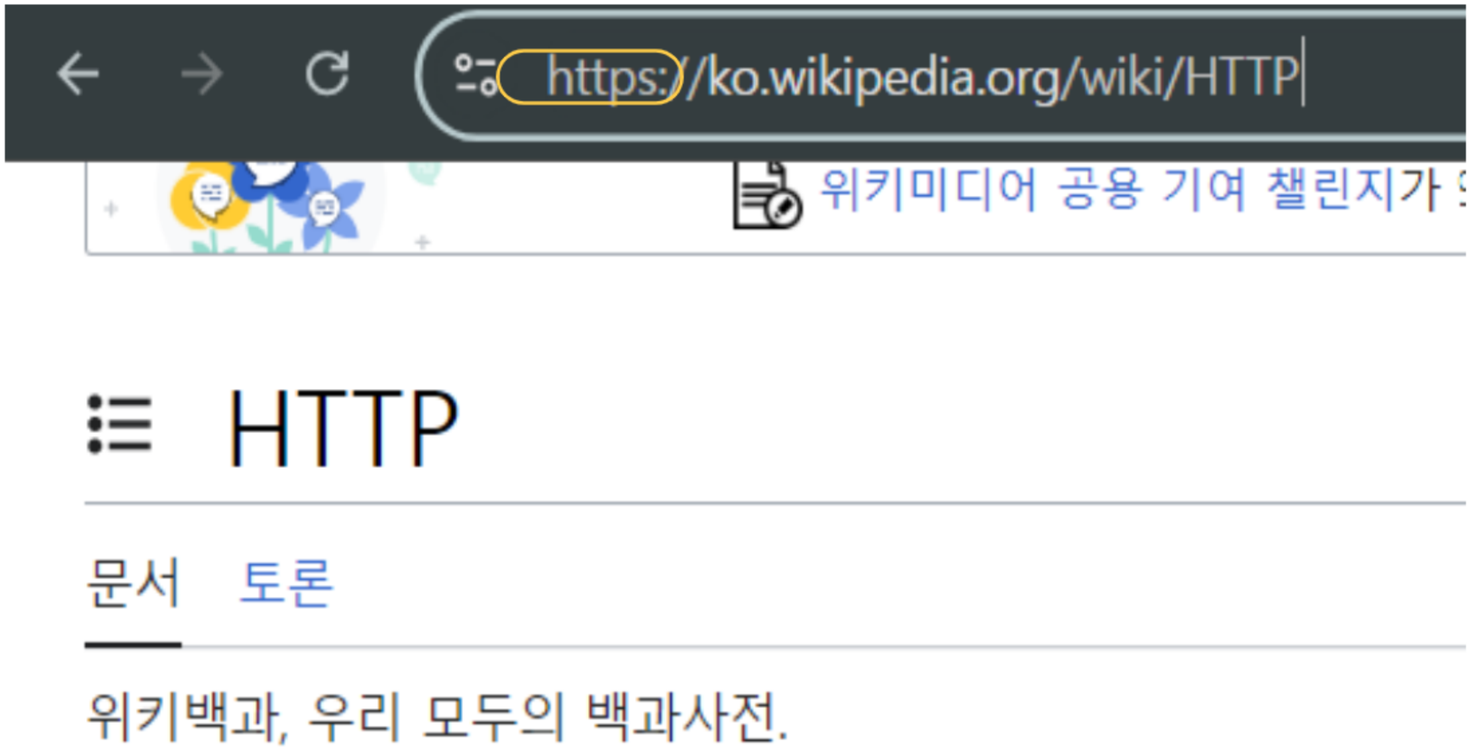
웹 구성요소: HTTP



2.HTTP(Hyper Text Transfer Protocol)

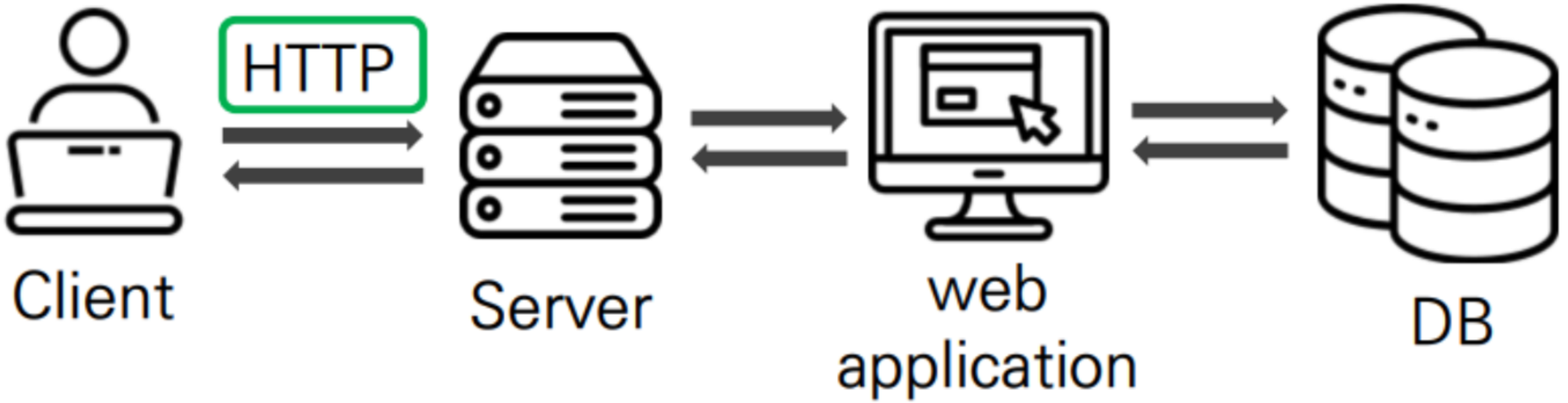
웹에서 정보를 주고 받을 수 있는 프로토콜(규약)
암호화 되지 않은 평문 전송 ➡ Sniffing 공격에 취약
OSI 7 Layer 7계층

웹 구성요소: HTTP



http 프로토콜로 전달되는 자료 : url의 "httlp(s)"로 확인

웹 구성요소: Web Browser

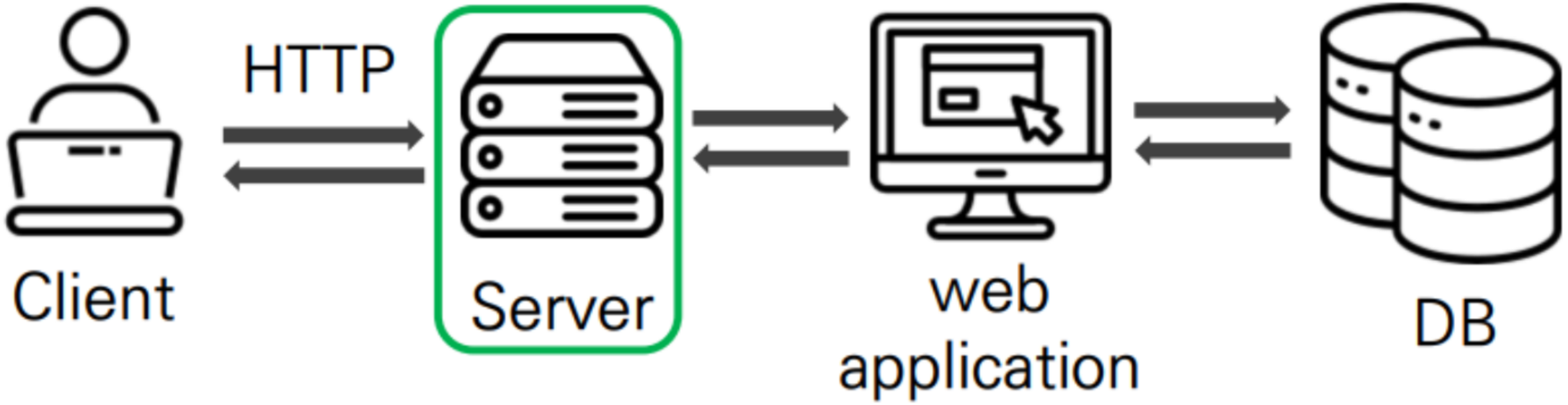


3.웹 브라우저(Web Browser)

클라이언트로의 요청으로 Request를 작성해 웹 서버에 전달

웹 서버로부터 응답 받은 Response를 해석해 사용자에게 보여주는 소프트웨어(Tool)
ex) Internet Explorer, FireFox, Chrome, Safari

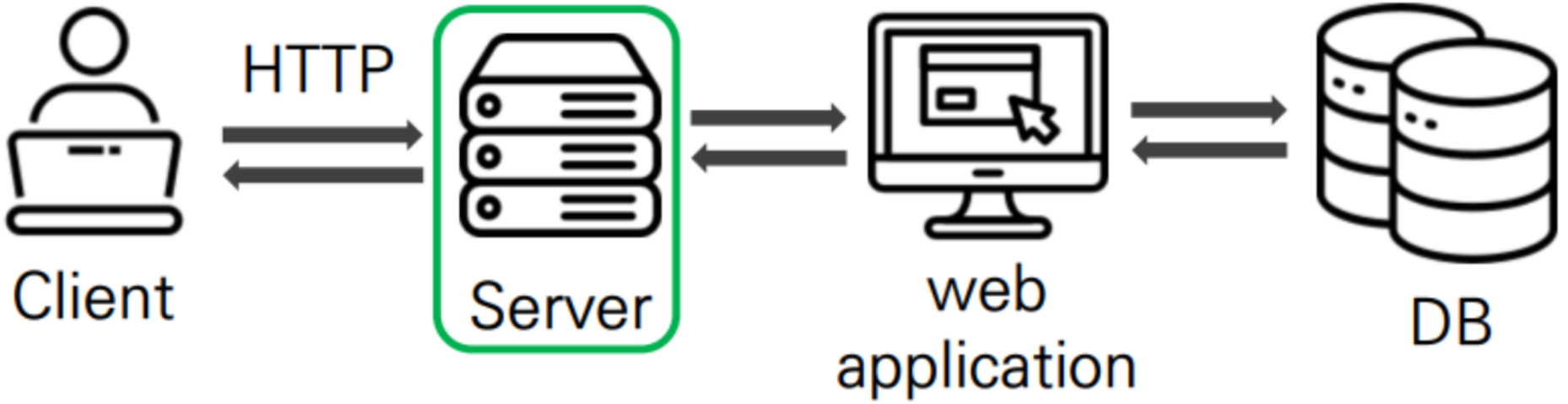
웹 구성요소: Server



4. 웹 서버(Web Server)

클라이언트의 요청에 따라 HTML 문서를 클라이언트에게 제공
웹 리소스 관리 및 제공

웹 구성요소: Web application



5. 웹 어플리케이션(Web Application)

브라우저를 통해 접근할 수 있는 응용 프로그램
HTTP에서 동작되는 프로그램
ex) jsp, php 등

웹 리소스: 정적/동적 리소스

정의

웹 사이트에 접속하거나 API 호출 시 /서버로부터 제공받는/식별 가능한 자원들

정적 리소스

고정적 자원
서버에 미리 저장된 파일이 그대로 전달된다.
ex) 텍스트 파일, html 파일, jpeg 이미지 파일, 유튜브 로고

동적 리소스

가변적 자원들.
"요청"에 따라 콘텐츠를 생산하는 프로그램
ex) 주식 거래, 인터넷 검색엔진, 실시간 검색어,유튜브 실시간 트렌드 영상 등

웹 리소스: URI

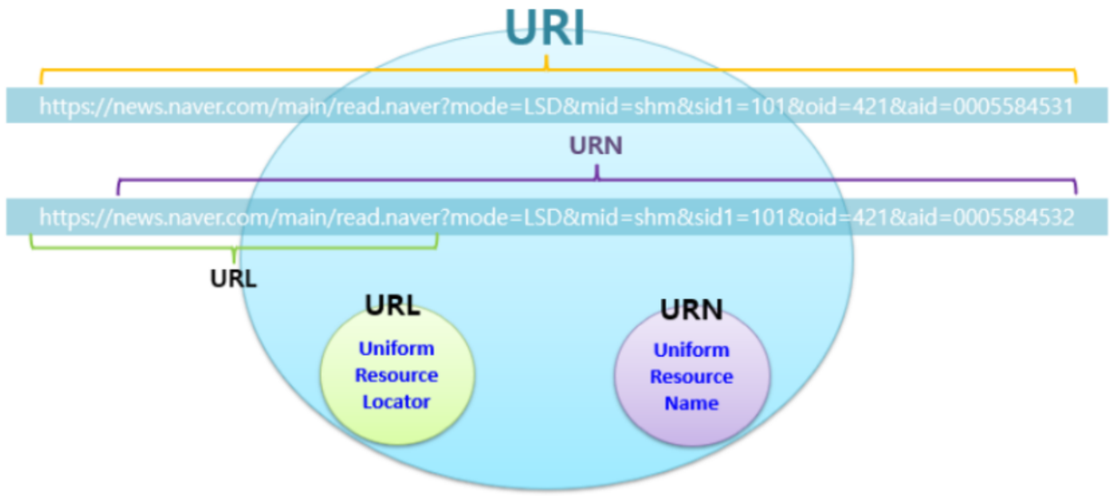
URI(통합 자원 식별자)

웹 서버 리소스의 이름

➡ 정보 리소스를 고유하게 식별하고 위치 저장.

➡ URI를 통해 서버 내 원하는 리소스 접근

URN과 URL을 포함한다.



웹 리소스: URL

URL(통합 자원 지시자)

가장 대중적인 리소스 식별자

특정 서버의 한 리소스에 대한 구체적 위치 서술

➡ 리소스의 위치를 옮기면 해당 url을 더 이상 사용할 수 없다.

웹 리소스: URN

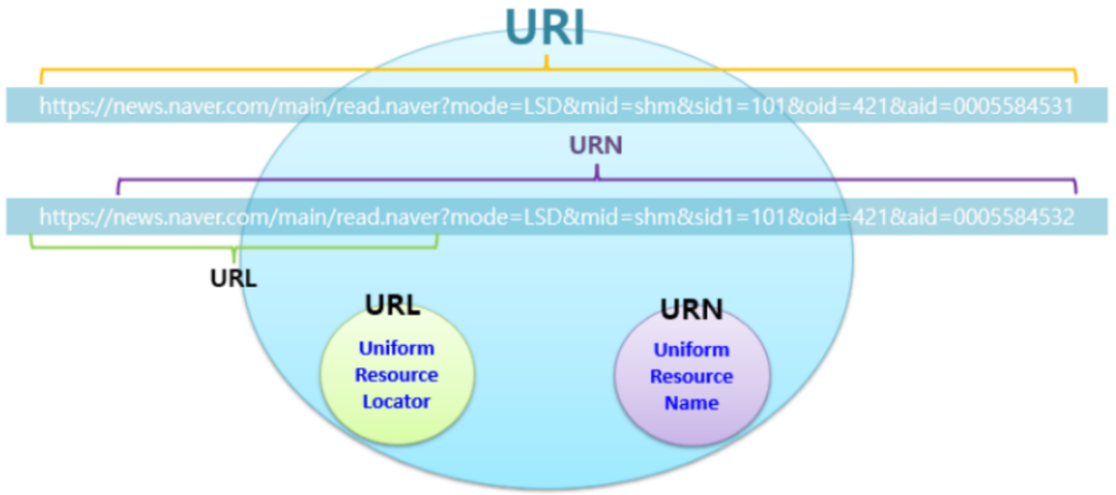
URN(통합 자원 이름)

리소스의 위치와 관계없이 이름만으로 식별 가능한 고유 이름

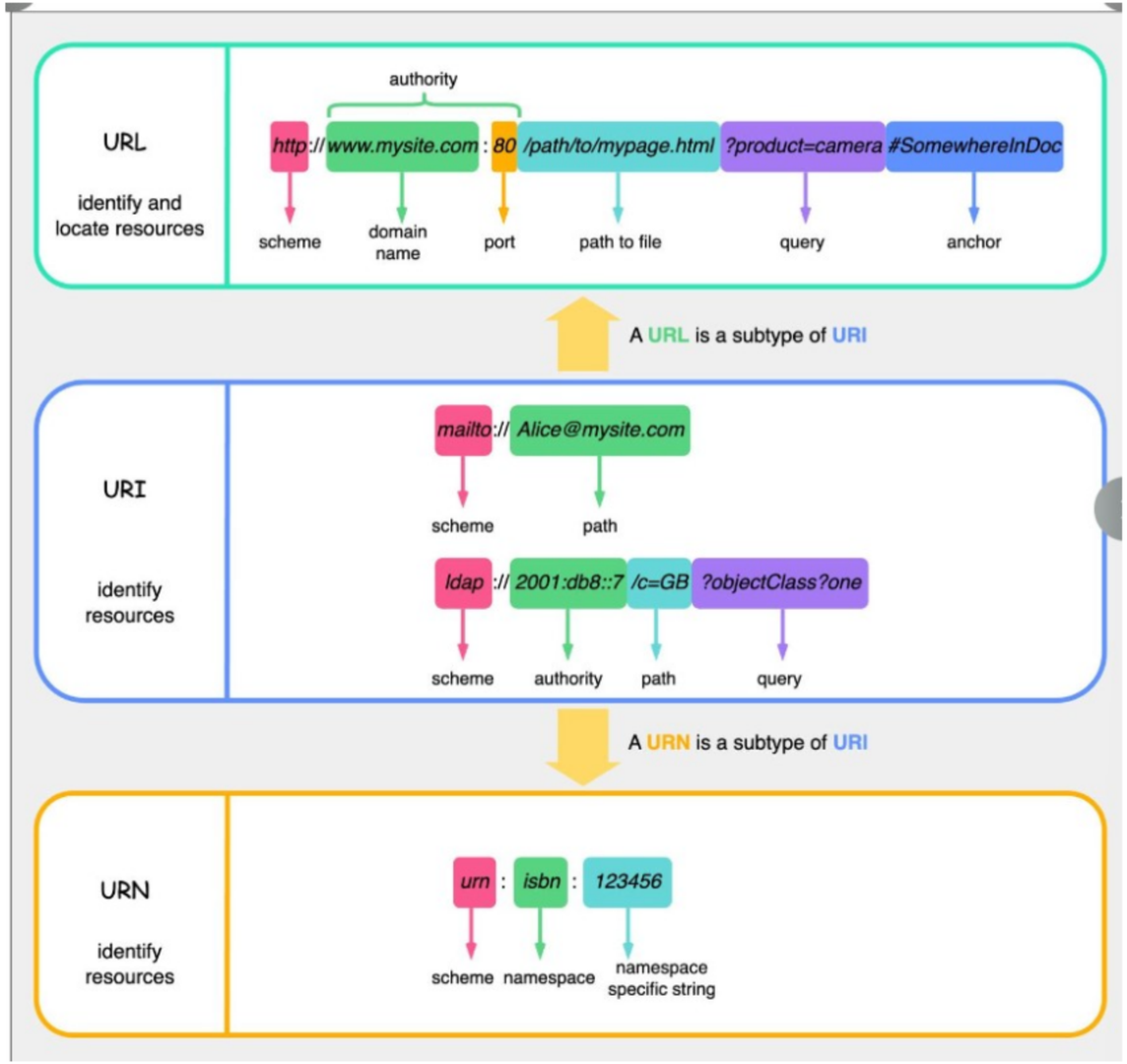
➡ 이름이 변하지 않는 한 리소스 위치가 변경되더라도 문제 없이 동작

URL의 한계로 등장

➡ 아직까지 대중화는 X



URI/URN/URL의 관계

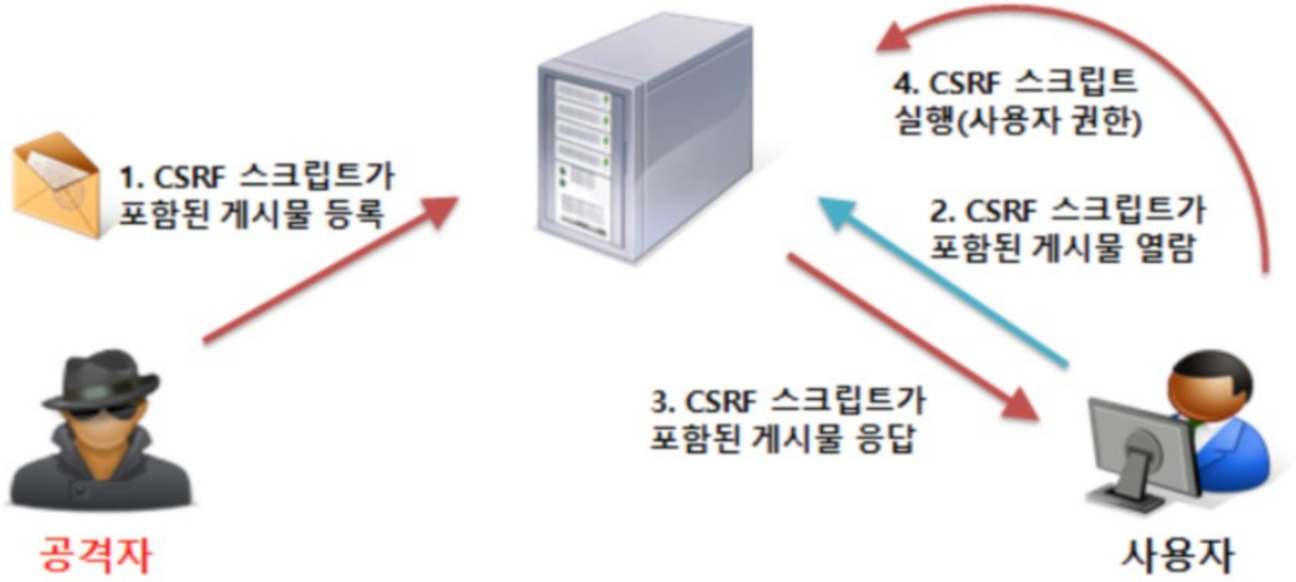


CSRF 공격

Cross SIte Request Forgery

사이트간 요청 위조

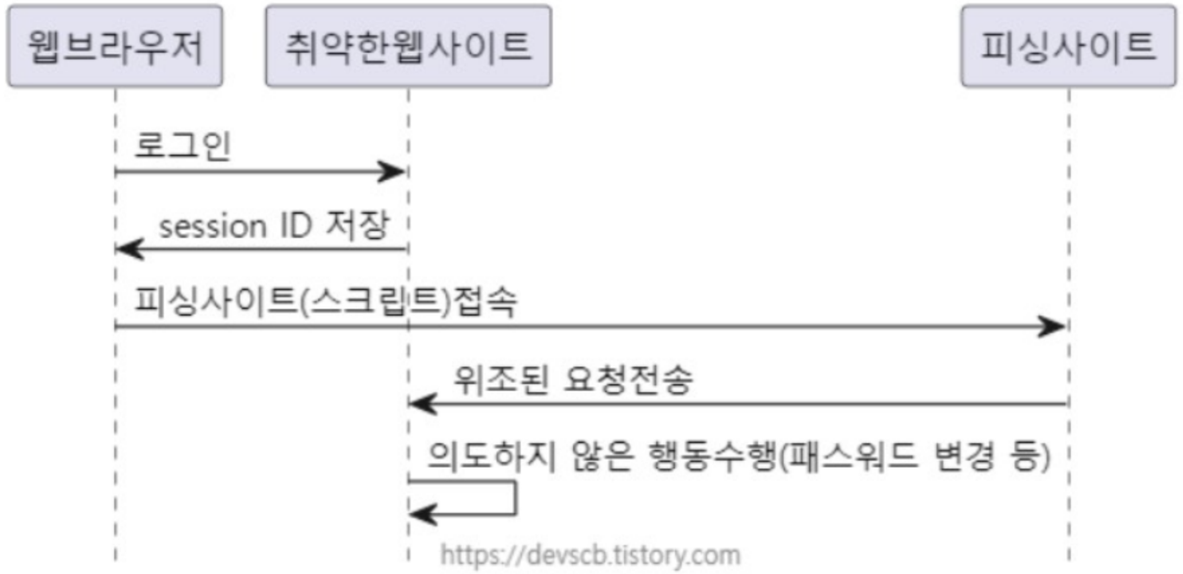
사용자가 자신의 의지와 무관하게 공격자가 의도한 행위를 특정 웹사이트에 요청하게 된다.



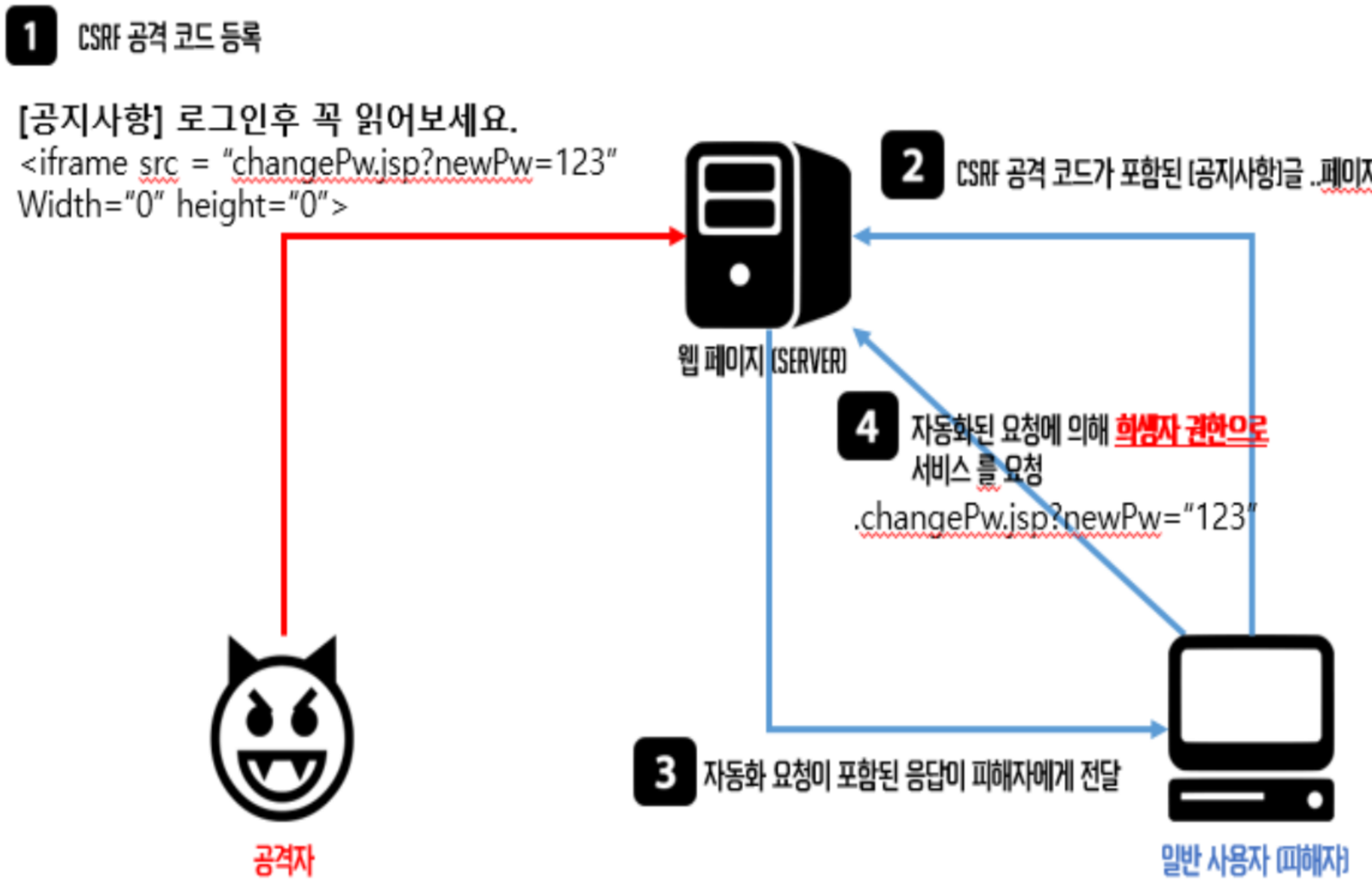
CSRF 공격

공격조건

1. 사용자) 보안 취약 서버 로그인 완료
2. 쿠키 기반의 서버 세션 정보 획득
3. 공격자) 서버 공격하기 위한 요청 방법 미리 파악



CSRF 공격 기법



1) 로그인 상태의 사용자가 서버에
악성코드 포함된 페이지 '요청'

2) 서버: 페이지 응답

3) 페이지 열람한 사용자:
자동적으로 페이지에 포함된 악성코드
<changePw.sjpg?newPw="123"
서버에 요청
(*현재 비밀번호를 "123"으로 변경하는 코드

4) 서버:
로그인한 사용자를 신뢰할 수 있는 존재로
판단해 요청 응답.

5) 공격자 의도에 따라 사용자 계정
비밀번호 변경
➡ 피해 계정은 공격에 사용됨

CSRF 공격 기법(ex)

옥션 해킹 사건

악성코드:
<img src="http://auction.com/changeUserAcoount?id=admin&password=admin" width="0" height="0">

➡ 너비, 높이 "0"의 이미지 첨부 코드
➡ 해당 이미지에는 현재 로그인 계정의 id/pwd 를 "admin"으로 변경하는 악성 코드가 첨부되어 있다.

공격 원리

해당 이미지를 메일에 첨부해 직원에게 전송.

➡ 이미지 크기가 "0"이므로 직원은 이미지의 존재를 알 수 없음.
➡ 직원이 로그인된 계정으로 메일을 열람할 경우, 이미지 소스를 받아오기 위한 url(악성코드)가 자동 실행.
➡ 서버에 id/pw "admin" 변경 요청
➡ 서버: 직원 계정을 신뢰할 수 있는 클라이언트로 판단해 요청 실행
➡ 공격 성공(id/pw 변경)

CSRF 방어기법

Referer 체크

HTTP Referer 헤더: 웹 페이지를 요청한 브라우저/애플리케이션 기록.
유입 경로 파악 및 접근 통제 역할

GET/POST 요청 구분

둘 모두 서버에 정보를 요청하는 HTTP 메서드
GET: 서버로부터 정보 요청. 파라미터에 내용이 노출되기 때문에 민감한 데이터 다룰 때 사용 X
POST: 서버에 데이터 전송(정보 생성/업데이트).

token 사용

서버에서 사용자에게 암호화된 token 발급
➡ 사용자 매 요청마다 token 전송
➡ 서버 token 검증 결과에 따라 응답

CSRF / XSS

	CSRF	XSS
공통점	이용자가 악성 스트립트를 포함한 페이지에 접속하도록 유도	
목적	임의 페이지에 HTTP 요청 보냄	세션 및 쿠키 탈취
공격 대상	서버	클라이언트

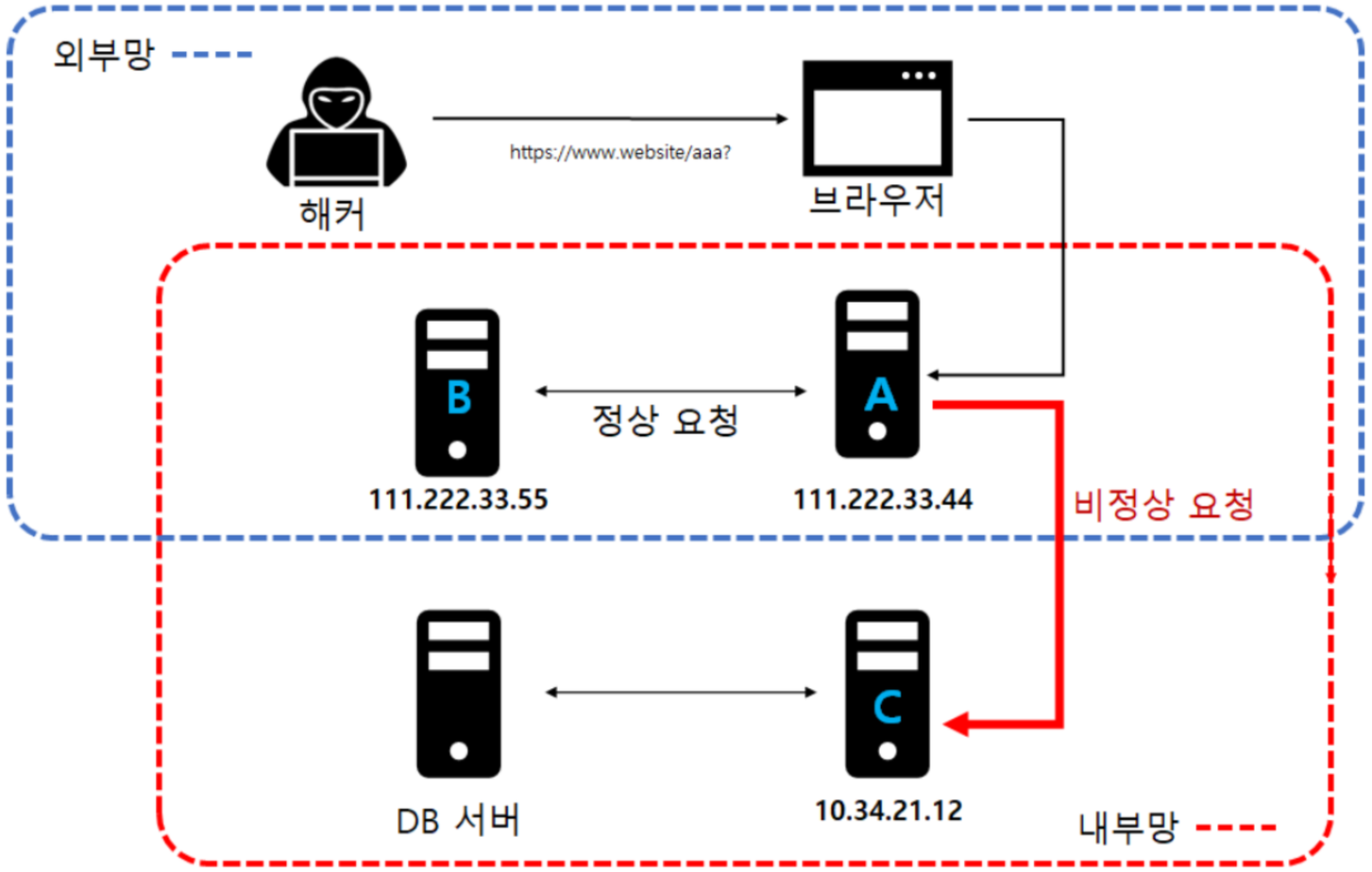
SSRF 공격

Server Side Request Forgery

서버에서 위조된 http 요청 발생시켜 서버 내부 자원에 접근
➡ 외부로 데이터 유출 및 오동작 유발

*csrf : 클라이언트 측 위조 ↔ ssrf : 서버 측 위조

SSRF 공격


외부망
https://www.website/aaa?
해커
브라우저
B
A
정상 요청
111.222.33.55
111.222.33.44
비정상 요청
C
DB 서버
10.34.21.12
내부망

SSRF 공격

취약한 대표 서비스

Website Translator

Image Viewer()

Document Reader

Web Crawler

URL Previewer

과제

1. 수업 내용 문서화

웹 구성요소/ 웹 리소스/ CSRF / SSRF

2. 문제 풀이 write-up

➡Dreamhack csrf-1

➡Dreamhack csrf-2

➡[Root Me] CSRF - 0 protection

제출방법

티스토리 [비밀글] 설정 뒤 카페 댓글에 게시물 링크 게시.