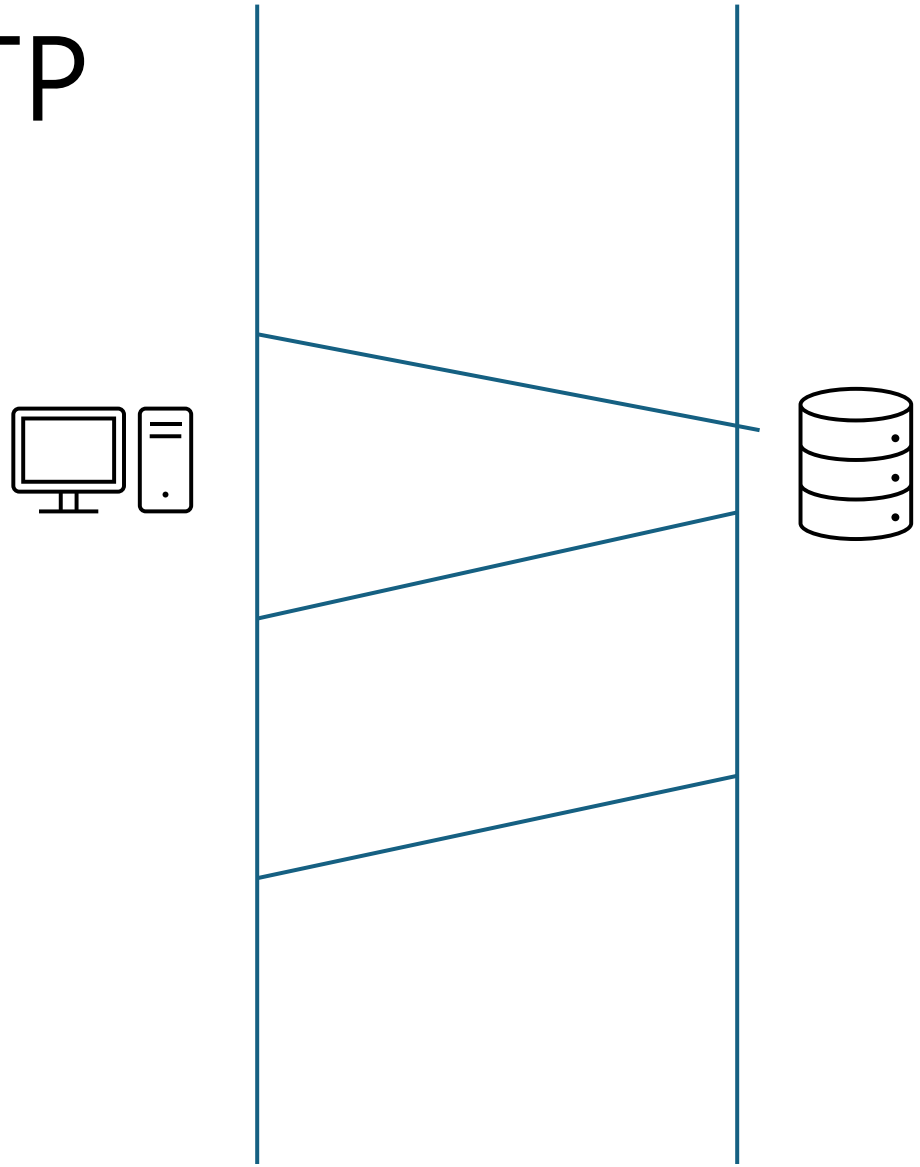
웹 보안

웹이란 무엇인가

HTTP(HyperText Transfer Protocol) + 주소 체계(URL) + 데이터 포멧(Hypertext)

[TCP/IP]

HTTP



(0) 접속(Connect)

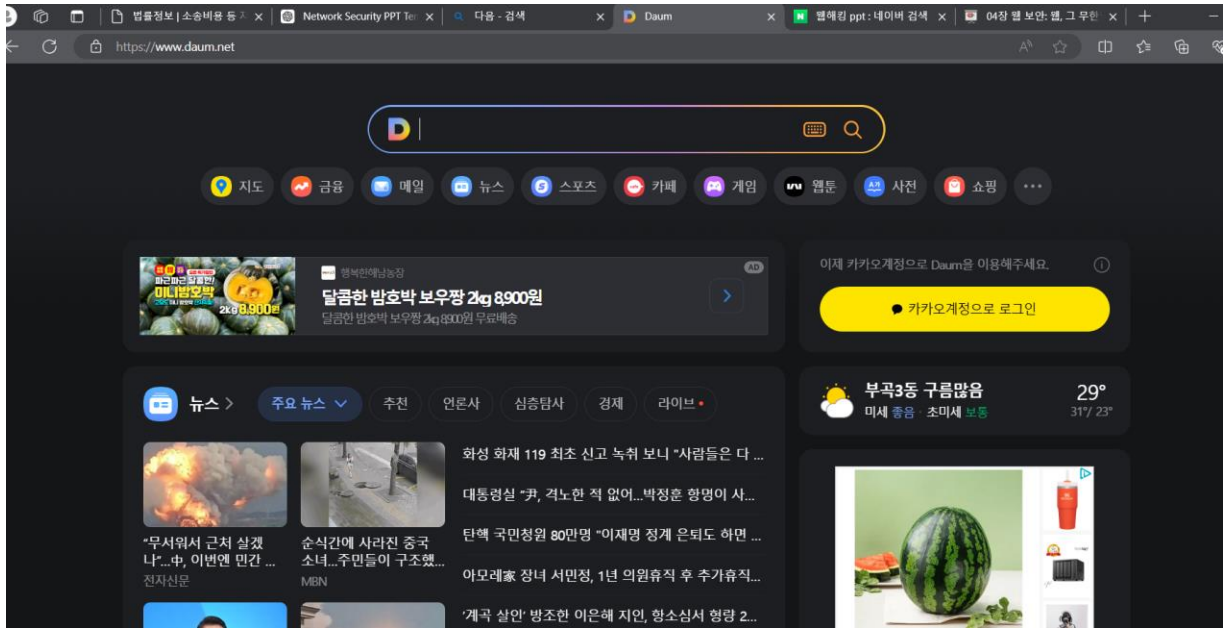
(1) 요청(Request)

(2) 응답(Response)

(4) 닫기(close)

Request p200

- Get 방식
- Post 방식



http response

실행 결과 코드	내용	설명
100번대	정보 전송	HTTP 1.0까지는 계열에 대한 정의가 이루어지지 않았기 때문에 실험 용도 외에는 100번대 서버 측의 응답이 없다.
200번대	성공	클라이언트의 요구가 성공적으로 수신 및 처리되었음을 의미한다.
300번대	리다이렉션	해당 요구 사항을 처리하기 위해 사용자 에이전트가 수행해야 할 추가적인 동작이 있음을 의미한다.
400번대	클라이언트 측 에러	클라이언트에 오류가 발생했을 때 사용한다. 예를 들면 클라이언트가 서버에 보내는 요구 메시지를 완전히 처리하지 못한 경우 등이다.
500번대	서버 측 에러	서버 자체에서 발생한 오류 상황이나 요구 사항을 제대로 처리할 수 없을 때 사용한다.

<실행결과 코드>

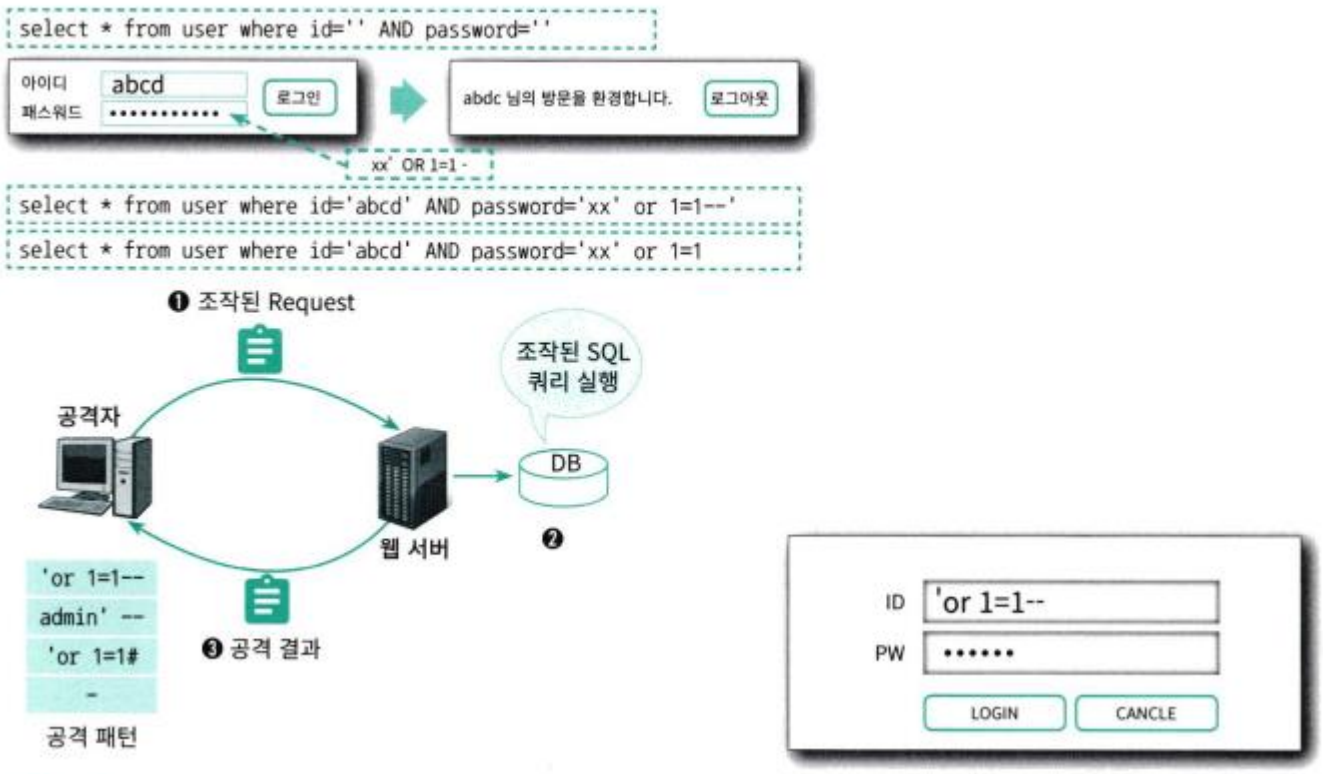
http Request

```
> Frame 6771: 269 bytes on wire (2152 bits), 269 bytes captured (2152 bits) on interface 0
> Ethernet II, Src: AsrockIn_a7:62:af (d0:50:99:a7:62:af), Dst: JuniperN_2a:48:01 (10:0e:7e:2a:48:01)
> Internet Protocol Version 4, Src: 10.100.124.127, Dst: 112.175.191.151
> Transmission Control Protocol, Src Port: 14578, Dst Port: 80, Seq: 1, Ack: 1, Len: 215
v Hypertext Transfer Protocol
  > GET /favicon.ico HTTP/1.1\r\n
    Accept: */*\r\n
    UA-CPU: AMD64\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Trident/6.0)\r\n
    Host: shopping.zum.com\r\n
    Connection: Keep-Alive\r\n
    \r\n
    [Full request URI: http://shopping.zum.com/favicon.ico]
    [HTTP request 1/1]
    [Response in frame: 6775]
```

http Response

```
Frame 6775: 1463 bytes on wire (11704 bits), 1463 bytes captured (11704 bits) on interface 0
Ethernet II, Src: JuniperN_2a:48:01 (10:0e:7e:2a:48:01), Dst: AsrockIn_a7:62:af (d0:50:99:a7:62:af)
Internet Protocol Version 4, Src: 112.175.191.151, Dst: 10.100.124.127
Transmission Control Protocol, Src Port: 80, Dst Port: 14578, Seq: 1, Ack: 216, Len: 1409
Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
  Server: nginx\r\n
  Date: Wed, 11 Apr 2018 05:24:58 GMT\r\n
> Content-Length: 1150\r\n
  Connection: keep-alive\r\n
  Keep-Alive: timeout=5\r\n
  Expires: Thu, 11 Apr 2019 05:24:58 GMT\r\n
  Cache-Control: max-age=31536000\r\n
  Last-Modified: Fri, 22 Sep 2017 06:42:26 GMT\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.004120000 seconds]
  [Request in frame: 6771]
  File Data: 1150 bytes
> Data (1150 bytes)
```

SQL 인젝션



일반적인 경우	SELECT * FROM member WHERE ID = 'admin' AND PW = '진짜암호'
SQL 인젝션의 경우	SELECT * FROM member WHERE ID = '' or 1=1 AND PW = '아무거나'

보안대책

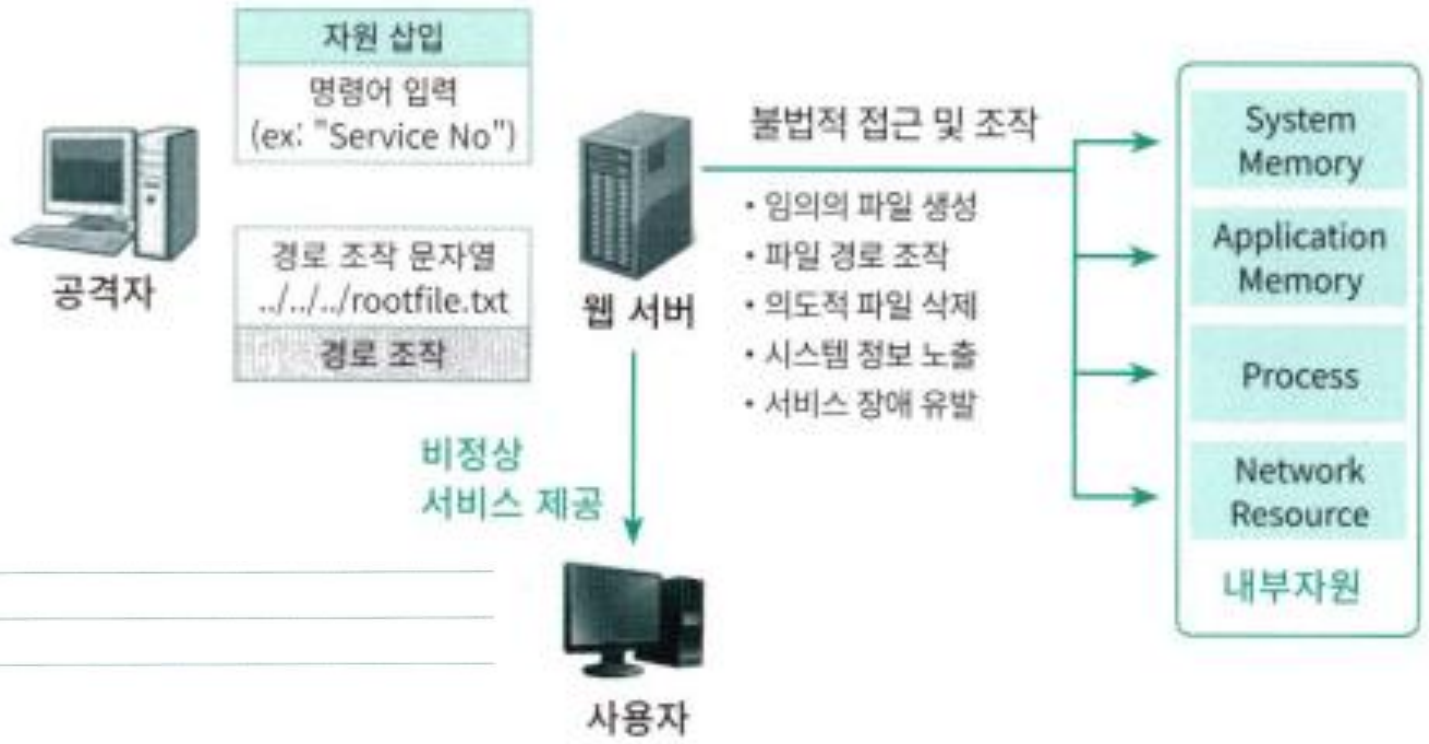
1. 매개변수 바인딩

매개변수 사용	SELECT * FROM member WHERE ID = ? AND PW = ? 사용자가 입력한 ID와 PW 값은 별도의 매개변수로 전달

2. 입력 값에 대한 특수 문자 검증

“.” -> “”

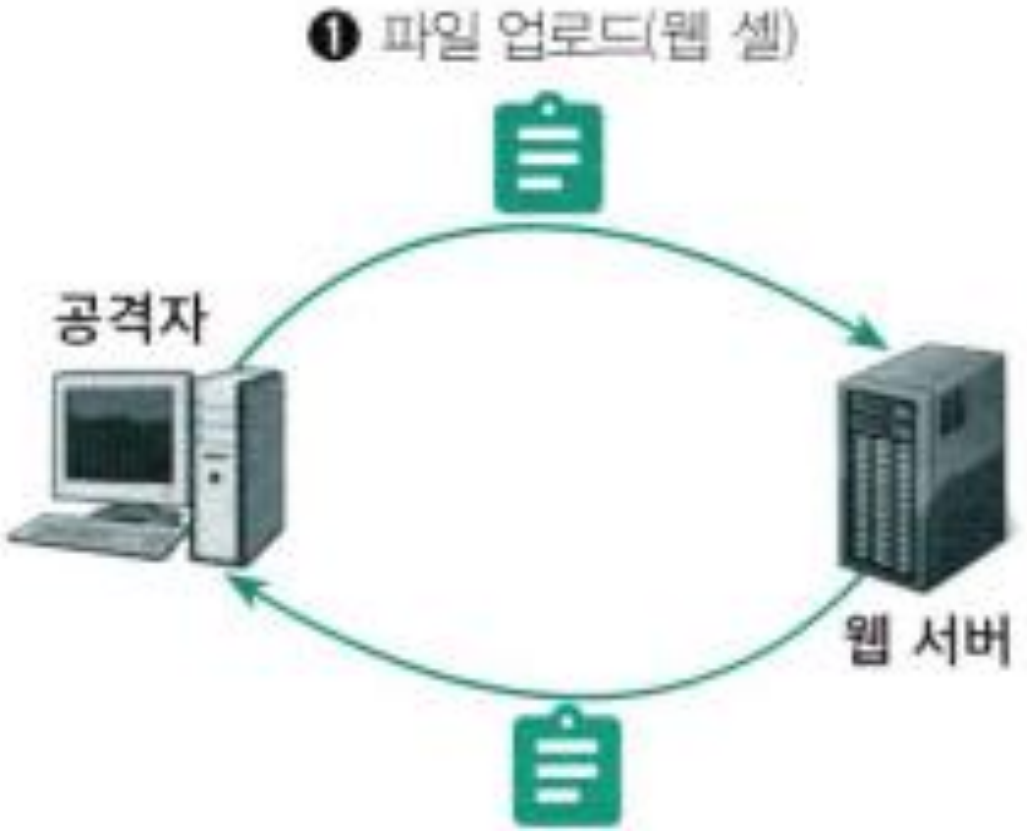
경로조작 공격!



일반적인 경우	myfile.doc
경로 조작의 경우	..\..\mysql\my.ini

위험한 형식의 파일 업로드

- Php, jsp, asp 등 웹서버에 지원되는 언어



보안 대책

- 화이트리스트 방식의 허용된 확장자만 업로드 허용

크로스 사이트 스크립트

- 웹브라우저 사용자가 입력하는 입력 값에 대해 유효성 검증 없이 웹서버로 전송했을 때 발생

보안대책

- 입력값에 대한 특수문자 검증