

# 인터넷 프로토콜과 웹 보안

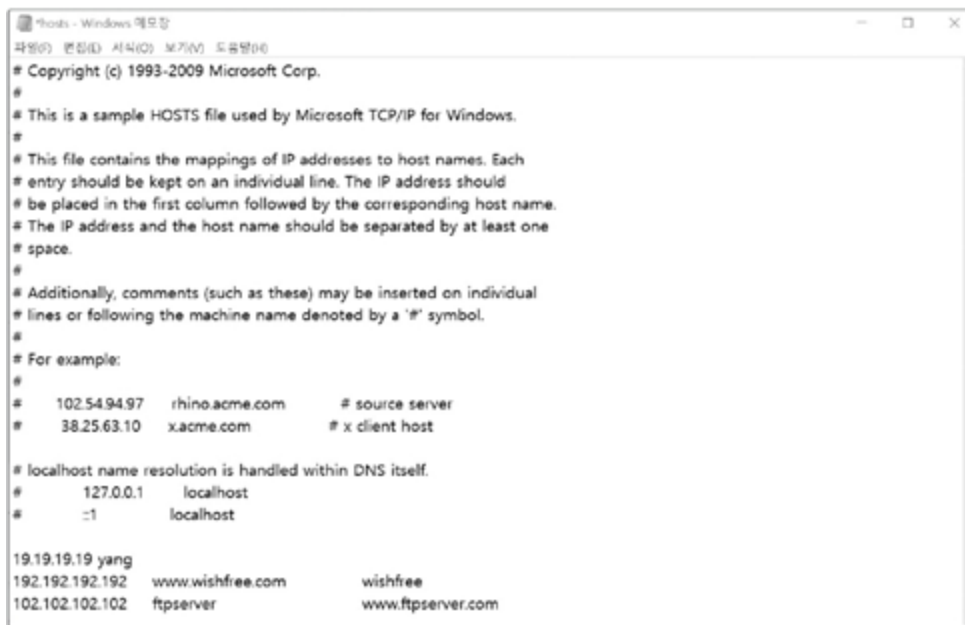
융합보안대학원 장대희

# 01. DNS와 hosts 파일

# hosts 파일을 이용한 정보 수집

## ❖ Hosts 파일

- DNS가 존재하기 전에 사용했고, 현재도 목적에 따라 많이 사용됨
- 윈도우 계열 시스템은 (윈도우 설치 디렉터리)\windows\system32\drivers\etc\hosts, 리눅스는 /etc/hosts가 이에 해당



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host
#
# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost
#
19.19.19.19 yang
192.192.192.192 www.wishfree.com wishfree
102.102.102.102 ftpserver www.ftpserver.com
```

그림 3-1 hosts 파일의 예(C:\windows\system32\drivers\etc\hosts)

# hosts 파일을 이용한 정보 수집

## ❖ Hosts 파일의 기본 구조

IP 주소	도메인 이름 또는 임의의 명칭
-------	------------------

그림 3-2 hosts 파일의 구조

- hosts 파일은 보통 비어 있음
- DNS 서버가 작동하지 않을 때 별도의 네트워크를 구성해 임의로 사용하고자 할 때, 다른 IP 주소를 가진 여러 대의 서버가 같은 도메인으로 클러스터링(Clustering)되어 운영되는 상태에서 특정 서버에 접속하고자 할 때 유용

# [실습 3-2] hosts 파일을 이용해 이름 해석하기

## 도메인 등록하기

### 1 ping www.hanbit.co.kr

```
명령 프롬프트
C:\Users\alpha>ping www.hanbit.co.kr

Ping www.hanbit.co.kr [218.38.58.195]: 32바이트 데이터 사용:
218.38.58.195의 ping: 바이트=32 시간=5ms TTL=128
218.38.58.195의 ping: 바이트=32 시간=4ms TTL=128
218.38.58.195의 ping: 바이트=32 시간=5ms TTL=128
218.38.58.195의 ping: 바이트=32 시간=5ms TTL=128

218.38.58.195에 대한 Ping 통계:
패킷: 보낸 = 4, 받은 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 4ms, 최대 = 5ms, 평균 = 4ms

C:\Users\alpha>
```

### 218.38.58.195 www.hanbit.co.kr hanbit

```
hosts - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
# 127.0.0.1 localhost
# ::1 localhost
218.38.58.195 www.hanbit.co.kr hanbit
```

# [실습 3-2] hosts 파일을 이용해 이름 해석하기

## hosts 파일 동작 확인하기

### 2 ping hanbit

```
관리자: 명령 프롬프트
C:\Windows\system32>ping hanbit

Ping www.hanbit.co.kr [218.38.58.195] 32바이트 데이터 사용:
218.38.58.195의 패킷: 바이트=32 시간=23ms TTL=128
218.38.58.195의 패킷: 바이트=32 시간=6ms TTL=128
218.38.58.195의 패킷: 바이트=32 시간=7ms TTL=128
218.38.58.195의 패킷: 바이트=32 시간=6ms TTL=128

218.38.58.195에 대한 Ping 통계:
패킷: 보낸 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 6ms, 최대 = 23ms, 평균 = 12ms

C:\Windows\system32>
```

### ping -a www.hanbit.co.kr

```
명령 프롬프트
C:\Users\alpha>ping www.hanbit.co.kr

Ping www.hanbit.co.kr [218.38.58.195] 32바이트 데이터 사용:
218.38.58.195의 패킷: 바이트=32 시간=5ms TTL=128
218.38.58.195의 패킷: 바이트=32 시간=4ms TTL=128
218.38.58.195의 패킷: 바이트=32 시간=5ms TTL=128
218.38.58.195의 패킷: 바이트=32 시간=5ms TTL=128

218.38.58.195에 대한 Ping 통계:
패킷: 보낸 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 4ms, 최대 = 5ms, 평균 = 4ms

C:\Users\alpha>
```

# [실습 3-2] hosts 파일을 이용해 이름 해석하기

잘못된 주소를 등록하여 사이트 접속 차단하기

3 200.200.200.200 www.hanbit.co.kr



# DNS의 작동 원리

## ❖ DNS(Domain Name System)

- 숫자로 구성된 IP 주소를 사람이 이해하기 쉬운 명칭인 도메인 이름으로 상호 매칭시켜주는 시스템

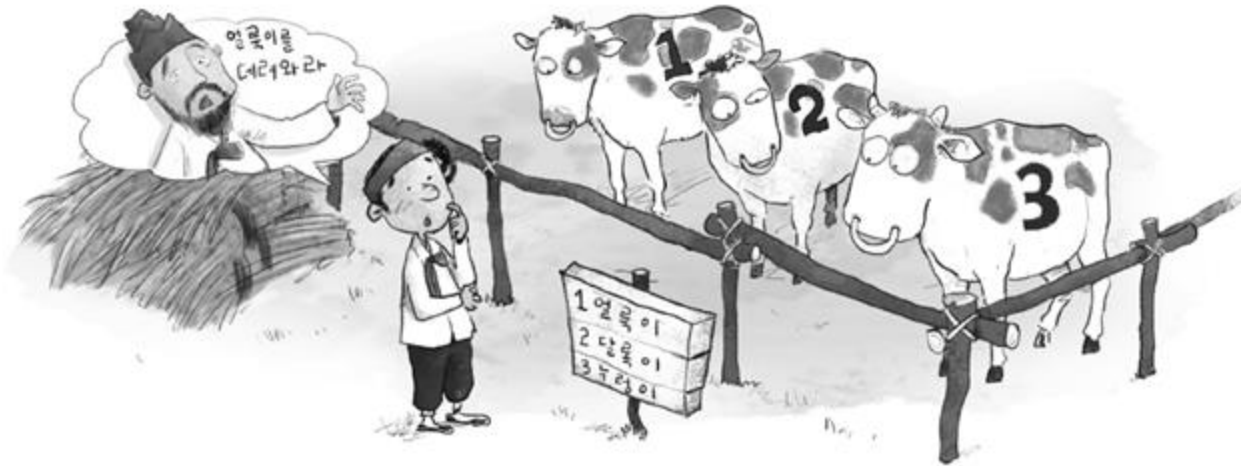


그림 3-3 DNS의 작동 원리 예



# DNS의 작동 원리

## ❖ DNS의 계층 구조

- 가장 상위 개체는 ‘.’ (Root)
- 두 번째 개체는 국가와 조직체의 특성
- 보통 맨 앞에 자신의 DNS 서버에서 지정해놓은 www, ftp와 같은 특정 서버의 이름이 옴
- FQDN(Fully Qualified Domain Name) : 완성된 주소(예 : www.wishfree.co.kr)

표 3-2 DNS의 두 번째 개체에 대한 내용

항목	내용	항목	내용
com	영리 기관	mil	군사 기관
net	네트워크 기관	edu	교육 기관
org	비영리 기관	int	국제 기관
gov	정부 기관	kr(Korea), jp(Japan)	국가 이름

# DNS의 작동 원리

## ❖ DNS의 계층 구조

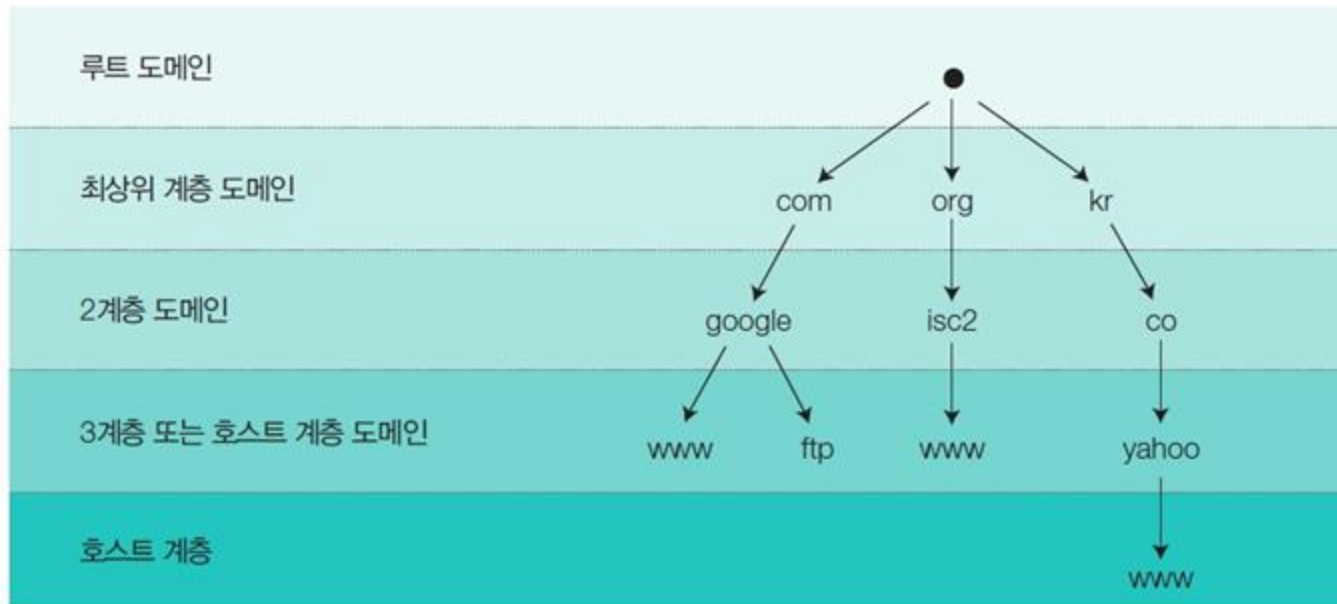
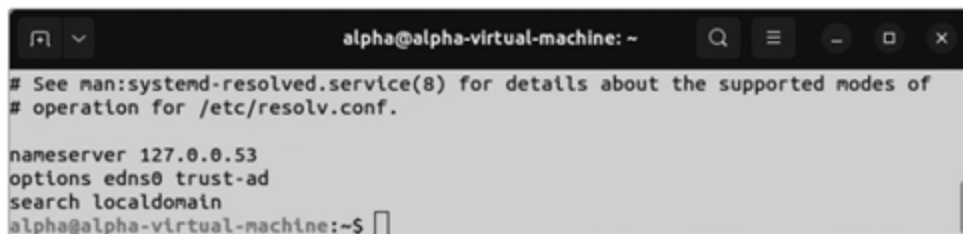


그림 3-4 DNS의 계층 구조

# DNS의 작동 원리

## ❖ 운영체제별 DNS 서버 등록

- 리눅스 : /etc/resolv.conf 파일에 DNS 서버를 입력

A terminal window titled 'alpha@alpha-virtual-machine: ~' showing the contents of the /etc/resolv.conf file. The text inside the terminal is: '# See man:systemd-resolved.service(8) for details about the supported modes of # operation for /etc/resolv.conf.', 'nameserver 127.0.0.53', 'options edns0 trust-ad', 'search localdomain', and 'alpha@alpha-virtual-machine:~\$' with a cursor.

```
alpha@alpha-virtual-machine: ~
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search localdomain
alpha@alpha-virtual-machine:~$
```

그림 3-5 리눅스의 DNS 서버 설정(vi /etc/resolv.conf)

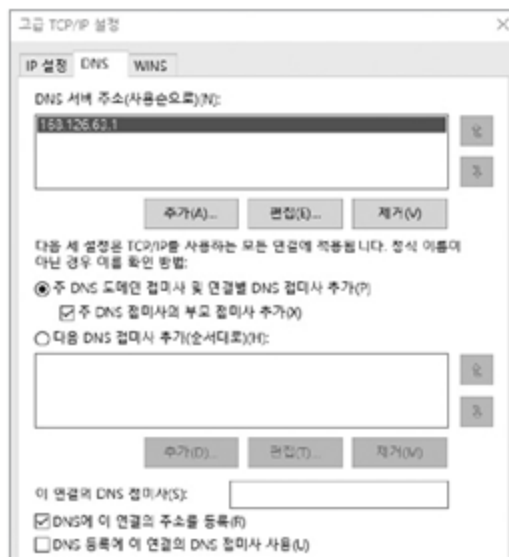
# DNS의 작동 원리

## ❖ 운영체제별 DNS 서버 등록

- 윈도우 : 인터넷 프로토콜(TCP/IP) 등록 정보에서 DNS 서버 두 개까지 입력
- [고급] 버튼을 누르면 좀 더 다양한 설정도 가능



(a) 인터넷 프로토콜(TCP/IP) 속성



(b) TCP/IP 고급 설정

그림 3-6 윈도우의 DNS 서버 설정

# DNS의 작동 원리

- 명령 창에 ipconfig /all 명령 입력



```
C:\Windows\system32> ipconfig /all

Windows IP 구성

호스트 이름 . . . . . : DESKTOP-BCDT852
주 DNS 접미사 . . . . . :
노드 유형 . . . . . : 혼성
IP 라우팅 사용 . . . . . : 아니요
WINS 프록시 사용 . . . . . : 아니요

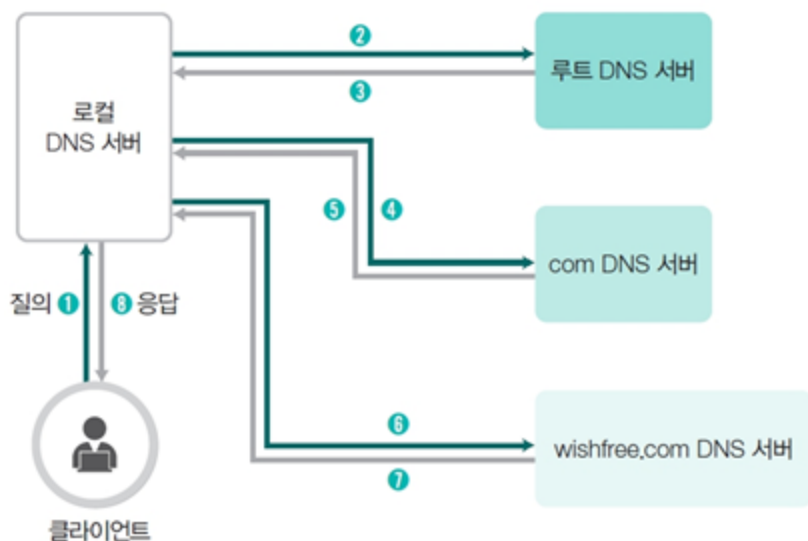
이더넷 어댑터 Ethernet0:

연결된 DNS 접미사 . . . . . :
물리적 주소 . . . . . : Intel(R) 82574L Gigabit Network Connection
00-0C-29-67-88-9E
DHCP 사용 . . . . . : 아니요
자동 구성 사용 . . . . . : 예
IPv4 주소 . . . . . : 192.168.15.201(기본 설정)
서브넷 마스크 . . . . . : 255.255.255.0
기본 게이트웨이 . . . . . : 192.168.15.2
DNS 서버 . . . . . : 193.126.63.1
Tcpip를 통한 NetBIOS . . . . . : 사용
```

그림 3-7 DNS 서버 확인하기(ipconfig /all 명령)

# DNS의 작동 원리

## ❖ DNS 서버의 이름 해석 순서



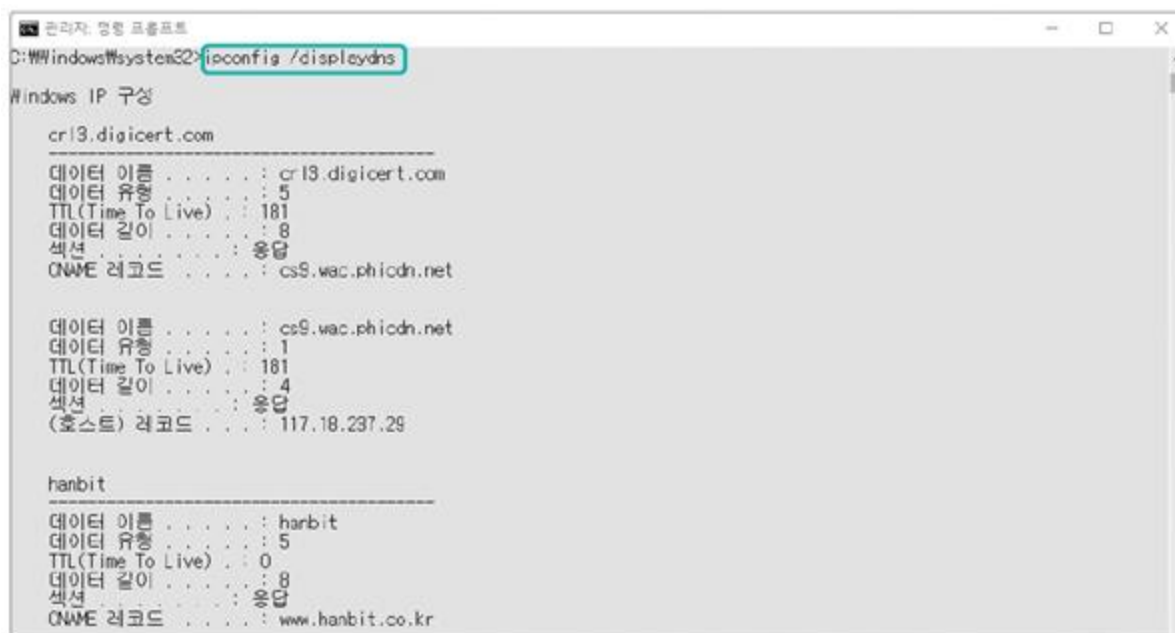
- 1 hosts 파일에 정보가 없으면 시스템에 설정된 DNS 서버인 로컬 DNS 서버에 질의한다.
- 2 로컬 DNS 서버에도 해당 정보가 없으면 루트 DNS 서버에 질의를 보낸다.
- 3 루트 DNS 서버에 `www.wishfree.com`에 대한 정보가 없으면 `com`을 관리하는 DNS 서버에 대한 정보를 보내 준다.
- 4 로컬 DNS 서버는 `com` DNS 서버에 `www.wishfree.com`에 대해 다시 질의한다.
- 5 해당 정보가 없을 경우, `com` DNS 서버는 다시 `wishfree.com`에 질의하도록 로컬 DNS 서버에 보낸다.
- 6 로컬 DNS 서버는 마지막으로 `wishfree.com`의 DNS 서버에 질의한다.
- 7 `wishfree.com`의 DNS 서버로부터 `www.wishfree.com`에 대한 IP 주소를 얻는다.
- 8 해당 IP 주소를 클라이언트에 전달한다.

그림 3-8 DNS 서버의 이름 해석 순서

# DNS의 작동 원리

## ❖ 윈도우에서 캐시된 DNS 정보 확인

**ipconfig /displaydns**



```
관리자: 명령 프롬프트
C:\Windows\system32>ipconfig /displaydns

Windows IP 구성

crl3.digicert.com
-----
데이터 이름 . . . . . : crl3.digicert.com
데이터 유형 . . . . . : 5
TTL(Time To Live) . . : 181
데이터 길이 . . . . . : 8
섹션 . . . . . : 응답
ONAME 레코드 . . . . . : cs9.wac.phicdn.net

데이터 이름 . . . . . : cs9.wac.phicdn.net
데이터 유형 . . . . . : 1
TTL(Time To Live) . . : 181
데이터 길이 . . . . . : 4
섹션 . . . . . : 응답
(호스트) 레코드 . . . . : 117.18.237.29

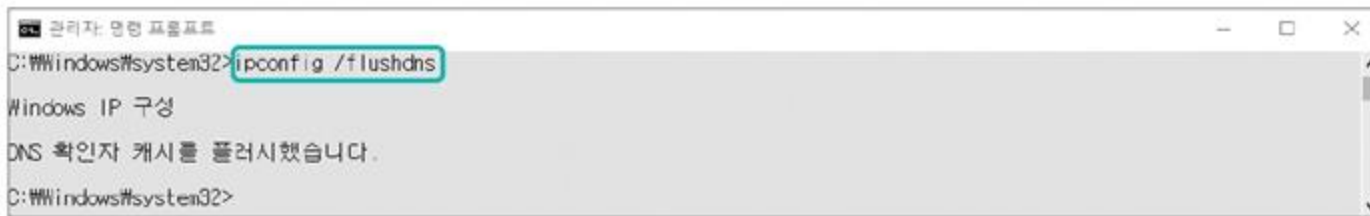
hanbit
-----
데이터 이름 . . . . . : hanbit
데이터 유형 . . . . . : 5
TTL(Time To Live) . . : 0
데이터 길이 . . . . . : 8
섹션 . . . . . : 응답
ONAME 레코드 . . . . . : www.hanbit.co.kr
```

그림 3-9 윈도우에서 캐시된 DNS 정보 확인

# DNS의 작동 원리

## ❖ 윈도우에서 캐시된 DNS 정보 삭제

**ipconfig /flushdns**



```
관리자: 명령 프롬프트
C:\Windows\system32>ipconfig /flushdns
Windows IP 구성
DNS 확인자 캐시를 풀러시했습니다.
C:\Windows\system32>
```

그림 3-10 윈도우에서 캐시된 DNS 정보 삭제



# DNS 레코드의 종류

표 3-3 DNS 레코드의 종류

종류	내용
A <sup>Address</sup>	<p>호스트 이름 하나에 IP 주소가 여러 개 있을 수 있고 IP 주소 하나에 호스트 이름이 여러 개 있을 수도 있다. 이를 정의하는 레코드 유형이 A이며, 다음과 같이 정의한다.</p> <ul style="list-style-type: none"> <li>- www A 200.200.200.20</li> <li>- ftp A 200.200.200.20</li> </ul>
PTR <sup>PoinTeR</sup>	A 레코드와 상반된 개념이다. A 레코드는 도메인에 대해 IP 주소를 부여하지만 PTR 레코드는 IP 주소에 대해 도메인 이름을 매핑하는 역할을 한다.
NS <sup>Name Server</sup>	DNS 서버를 가리키며, 각 도메인에 적어도 한 개 이상 있어야 한다.
MX <sup>Mail eXchanger</sup>	도메인 이름으로 보낸 메일을 받는 호스트 목록으로 지정한다.
CNAME <sup>Canonical NAME</sup>	호스트의 다른 이름을 정의하는 데 사용한다.
SOA <sup>Start Of Authority</sup>	도메인에 대한 권한이 있는 서버를 표시한다.
HINFO <sup>Hardware INFO</sup>	해당 호스트의 하드웨어 사양을 표시한다.
ANY(ALL)	DNS 레코드를 모두 표시한다.

# IP 주소 추적에 대한 이해

## ❖ IP 주소 추적의 기본

- IP 주소 추적의 기본은 출발지 IP 주소 확인하기
  - ✓ 모든 패킷에는 출발지와 목적지가 존재함
  - ✓ 인터넷을 떠돌아다니는 패킷에는 모두 출발지 IP 주소와 목적지 IP 주소가 입력되어 있음

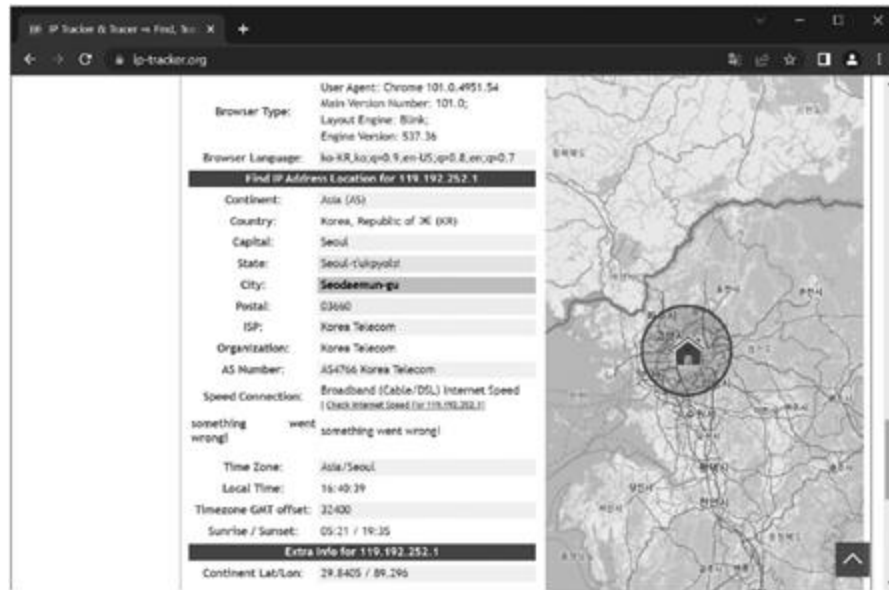


그림 4-1 IP 정보를 확인할 수 있는 사이트(www.jp-tracker.org)

# IP 주소 추적에 대한 이해

## ❖ IP 주소 추적의 기본

- 중간에 출발지 IP 주소를 바꿔주는 시스템이 존재할 경우, 진정한 출발지 IP 주소를 확인하는 데 어려움 존재
- 해커가 자신을 숨기기 위해 경유 시스템에 간단한 툴을 만들어 출발지 IP 주소를 숨길 수도 있음
  - ✓예) IP 주소가 부족해 사설 네트워크를 쓰는 경우 인터넷을 이용하기 위해 NAT을 많이 사용. 이때도 출발지 IP 주소는 확인할 수 없음
  - ✓예) 인터넷 회사의 서비스를 이용하는 경우에도 서비스를 제공하는 회사의 프로그램을 거치면서 출발지 IP 주소가 바뀌기도 함(이메일 서비스나 메신저 서비스 등이 해당)

# 서버로그를 통한 IP 추적

---

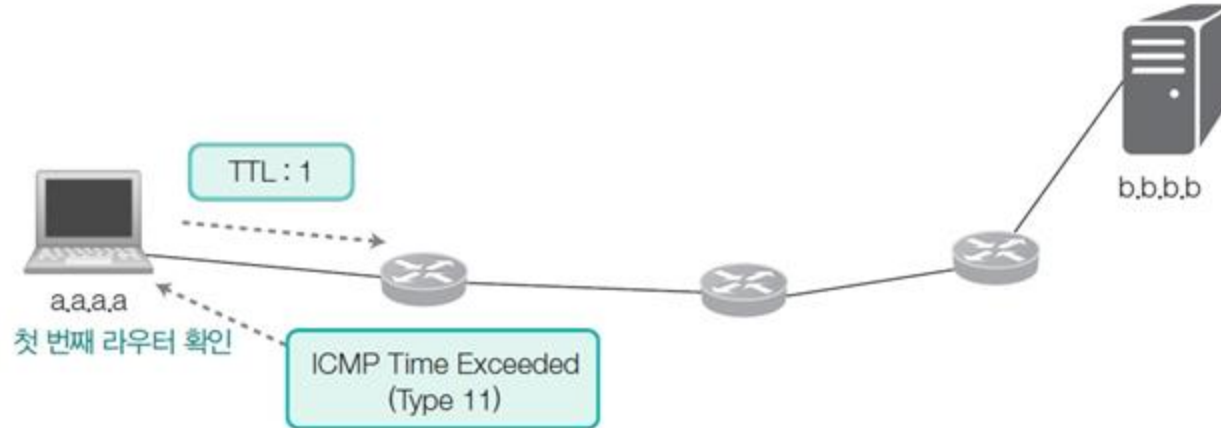
## ❖ 웹 해킹 공격

- 게임이나 포털 업체를 통해 개인 정보가 유출되는 사고의 상당수가 해당
- 해커는 웹 사이트의 구조를 파악하고 공격하기 위해 웹 게시판에 접근
- 서비스의 로그를 분석하면 해커의 IP를 확인 가능

# traceroute를 통한 IP 추적

- traceroute는 패킷이 목적지까지 도달하는 동안 거쳐 가는 라우터의 IP를 확인하는 툴
- 운영체제에서 기본으로 제공하므로 별도로 설치할 필요는 없음
- UDP와 ICMP, IP의 TTL 값을 이용

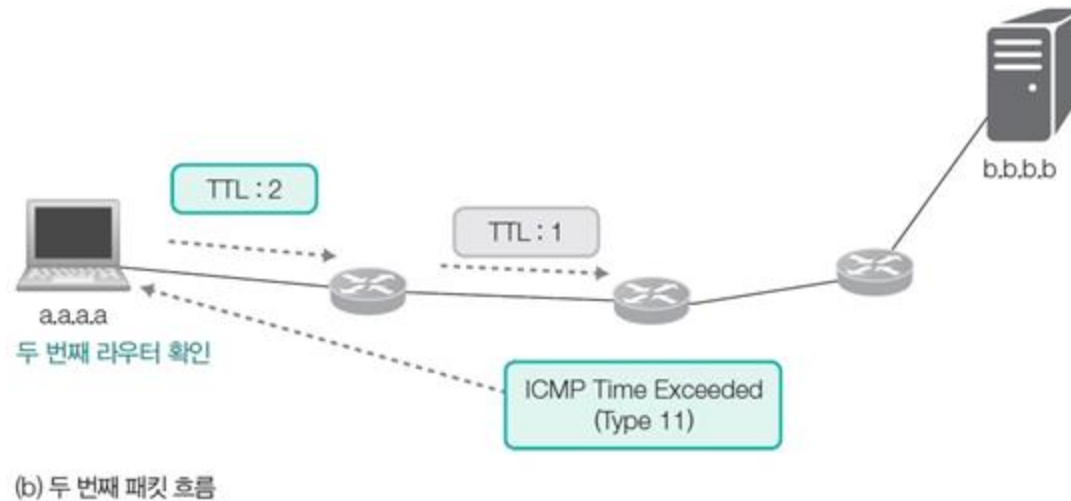
❖ a.a.a.a에서 b.b.b.b까지 traceroute를 한다고 가정한 동작 순서



(a) 첫 번째 패킷 흐름

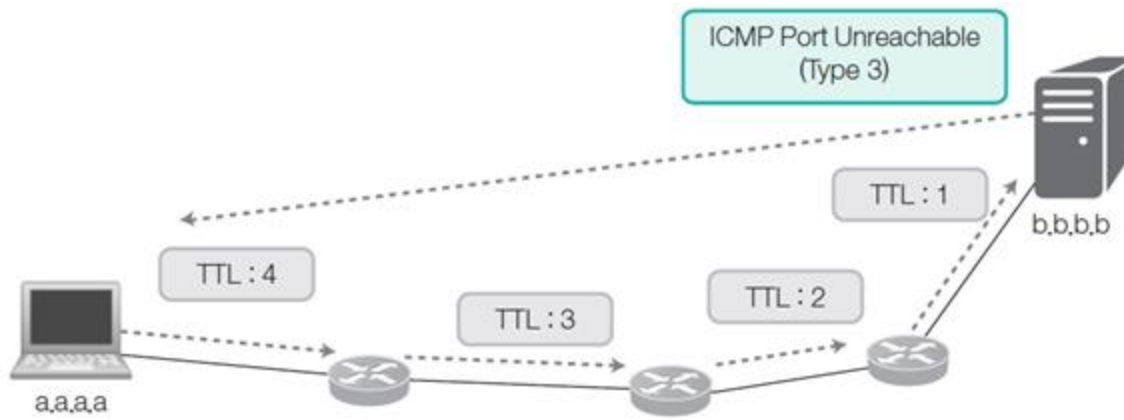
# traceroute를 통한 IP 추적

❖ a.a.a.a에서 b.b.b.b까지 traceroute를 한다고 가정한 동작 순서



# traceroute를 통한 IP 추적

❖ a.a.a.a에서 b.b.b.b까지 traceroute를 한다고 가정한 동작 순서



(c) 목적지 도달 시 패킷 흐름

그림 4-6 traceroute의 패킷 흐름

# 다크웹과 IP추적

---

## ■ 다크웹?

- 토르 브라우저
- Onion 프로토콜
- Hidden Server
- 익명성 보장 인터넷

## ■ 다크웹 문제

- 익명성 보장 -> 좋은 목적
- 범죄악용 -> 부작용  
✓수사기관의 IP 추적대상

## ■ 서버의 IP 노출

- 다크웹 서버에게는 매우 치명적인 보안문제



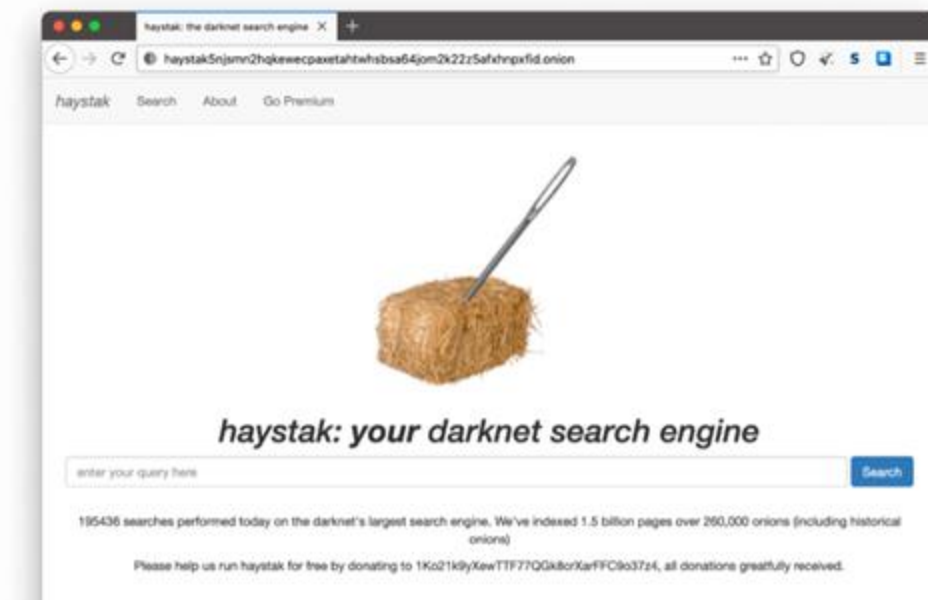
# Onion URL

## ■ Onion URL

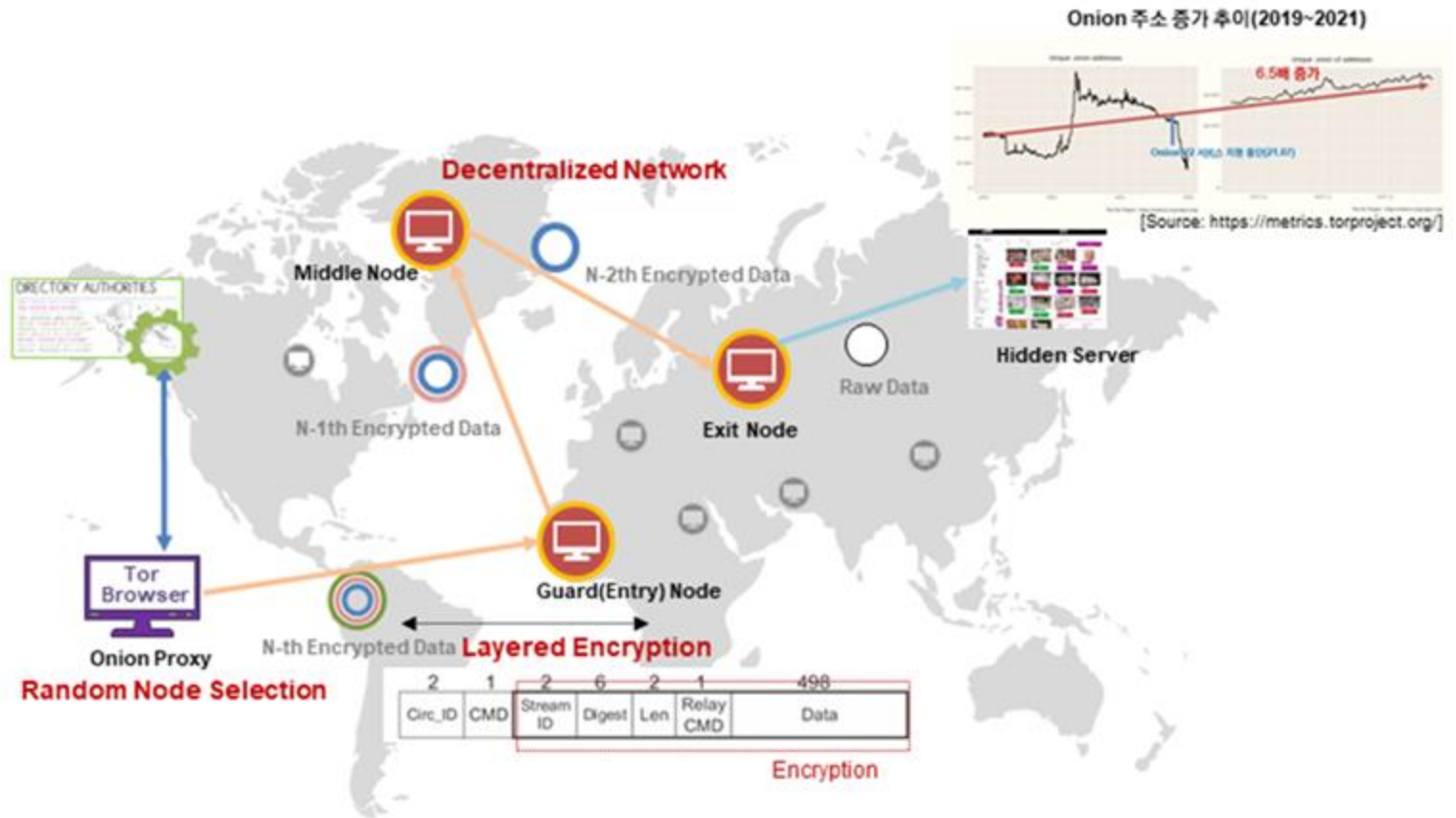
- 다크웹 hidden 서버의 public key 에 기반한 hash 값 (Base32 인코딩)

- 예) Haystak

<http://haystak5njsmn2hqkewecpaxetahtwhsbsa64jom2k22z5afxhnpxfid.onion/>



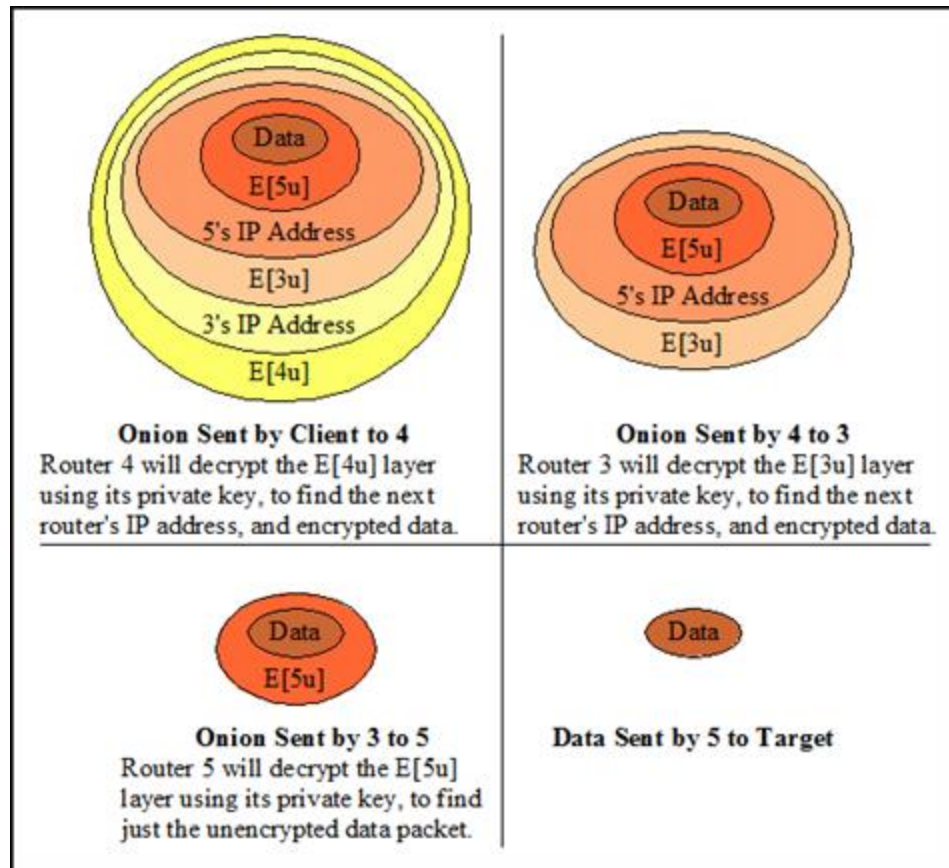
# Onion 라우팅



# Onion Protocol

## ■ 비대칭 암호화 알고리즘을 활용

- 비대칭 암호화 -> 이후 다룰 예정



# 다크웹 + 암호화폐

---

## ■ 다크웹 + 암호화폐

- 랜섬웨어 악성코드의 시대

## ■ 예시

- 다크웹에서 랜섬웨어를 암호화폐로 구입
- 랜섬웨어를 활용하여 암호화폐를 획득  
✓랜섬웨어? 악성코드?

# 웹해킹

---

## ❖수많은 웹사이트 존재

## ❖웹사이트는 "웹서버" 프로그램에 의해 작동

- Apache
- Nginx
- PHP
- ASP, JSP
- Nodejs, Django
- ...

## ❖웹 서버 프로그램

- 웹 어플리케이션
- 취약점 존재 가능
  - ✓ 웹 서버의 로직오류

# 웹 어플리케이션이란?

## ❖웹사이트

## ❖특정한 기능을 제공

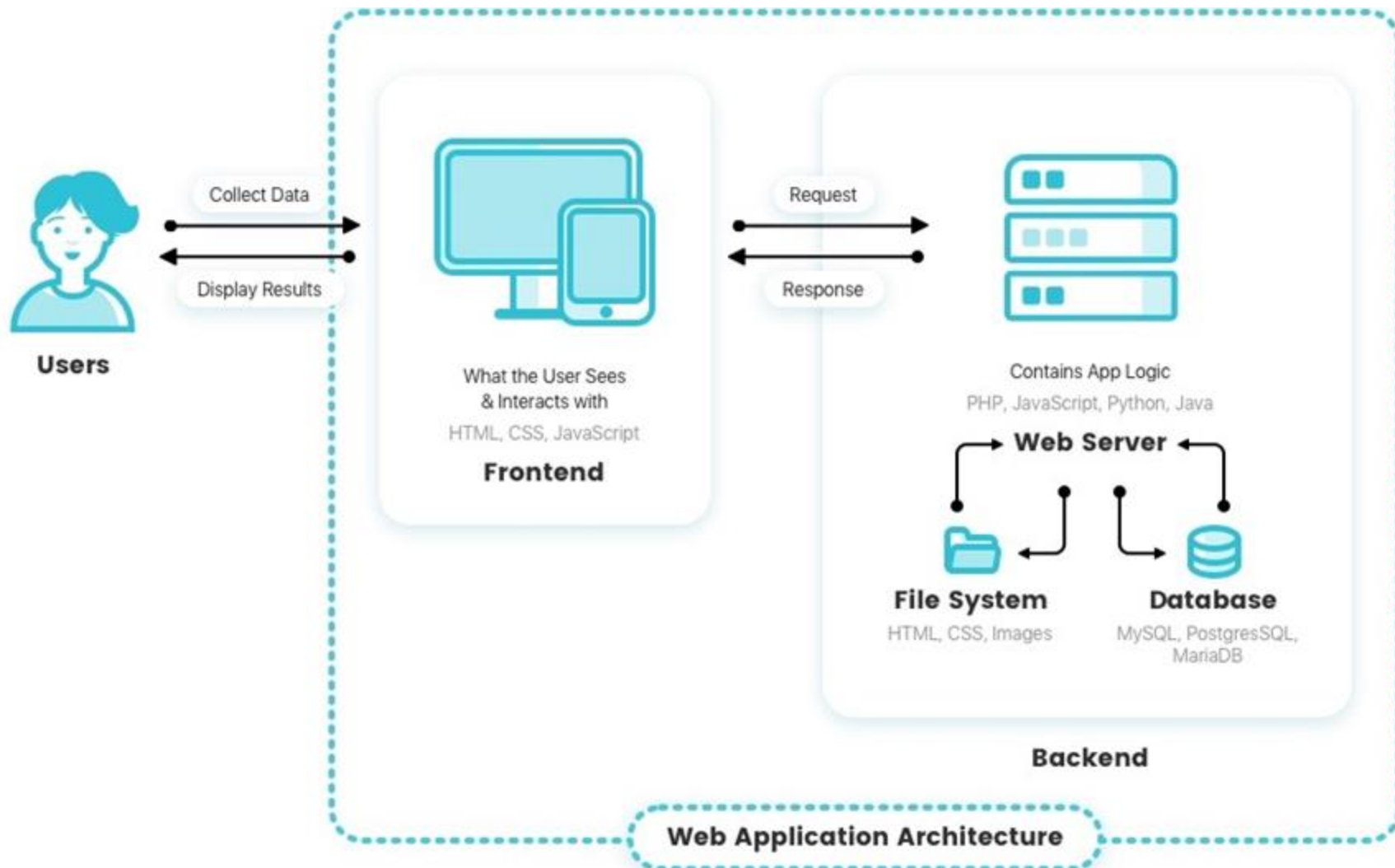
- e.g. 게시판

## ❖웹 개발언어로 작성

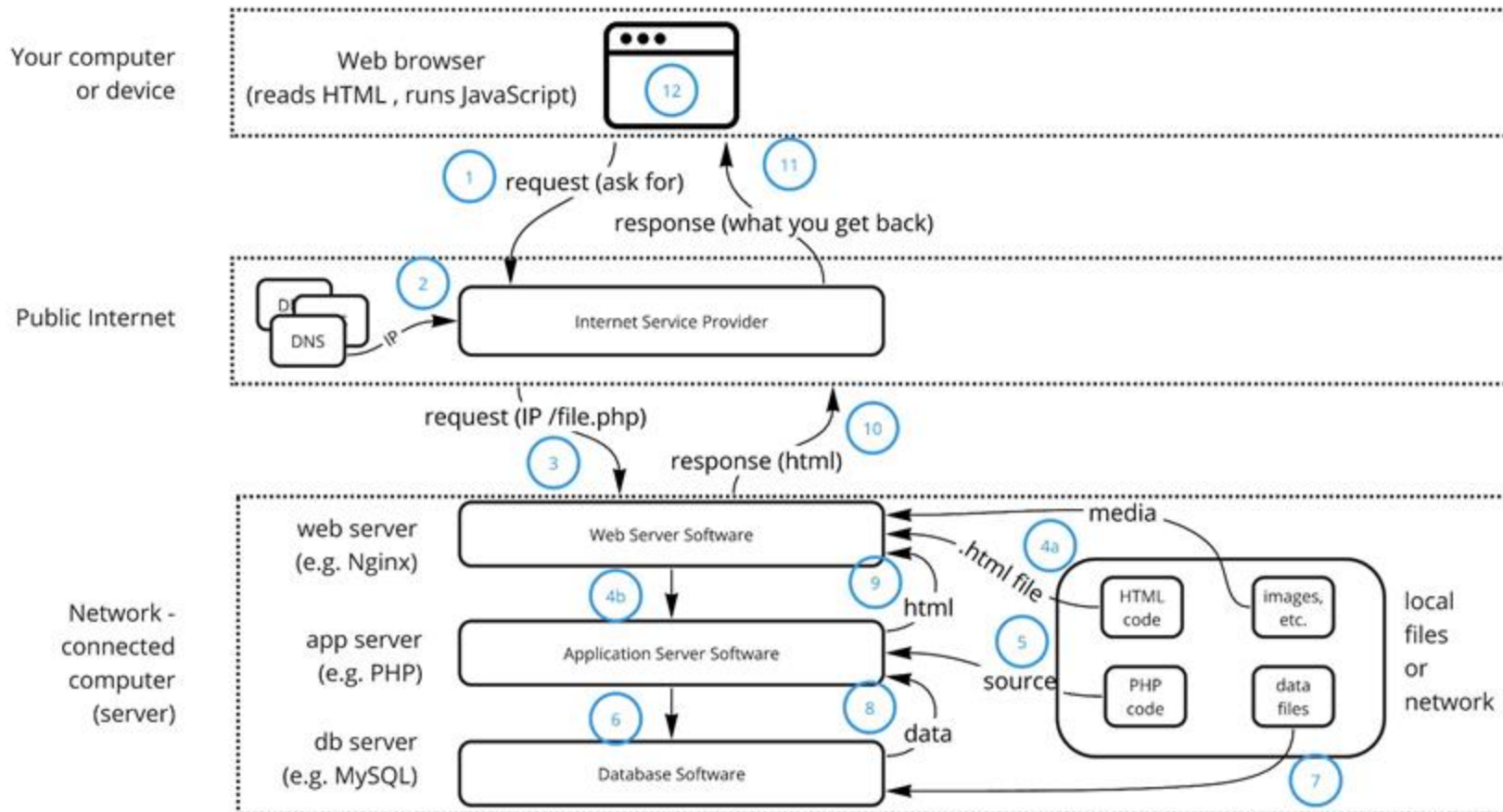
- PHP/ASP/JSP
- Flask
- nodejs, Django
- ...



# 웹 어플리케이션 구조



# 웹 어플리케이션 구조





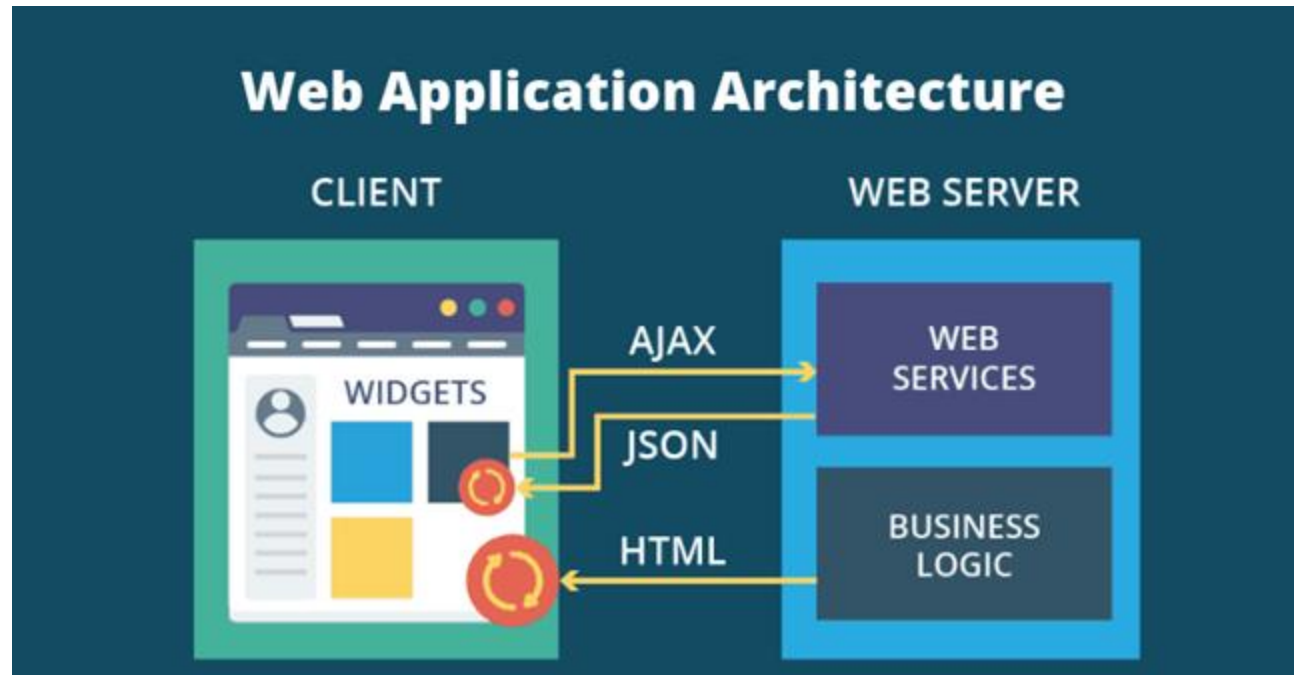
# 웹 어플리케이션 서비스들



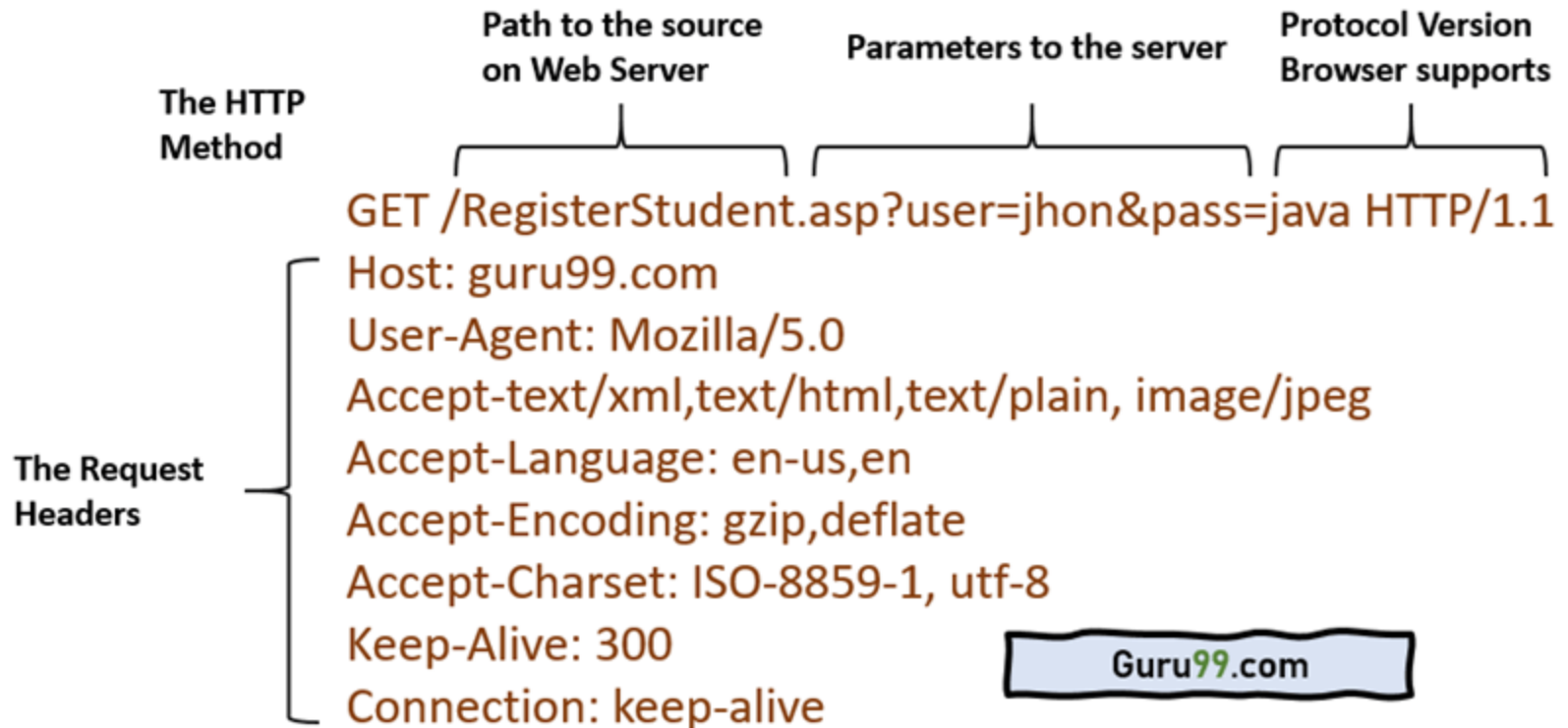
# 웹 어플리케이션의 형태

## ❖ 웹 어플리케이션 구성요소들

- HTML
- JavaScript
- AJAX
- JSON
- DBMS
- SQL
- Docker
- ...



# HTTP Protocol

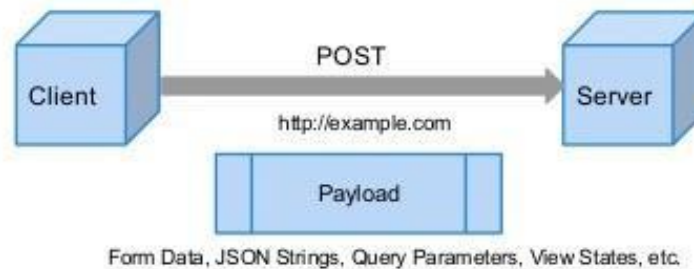


# GET / POST

## ❖클라이언트가 서버에게 보내는 프로토콜

- 데이터 전달
- 데이터 요구
- 이벤트 전송
- ...

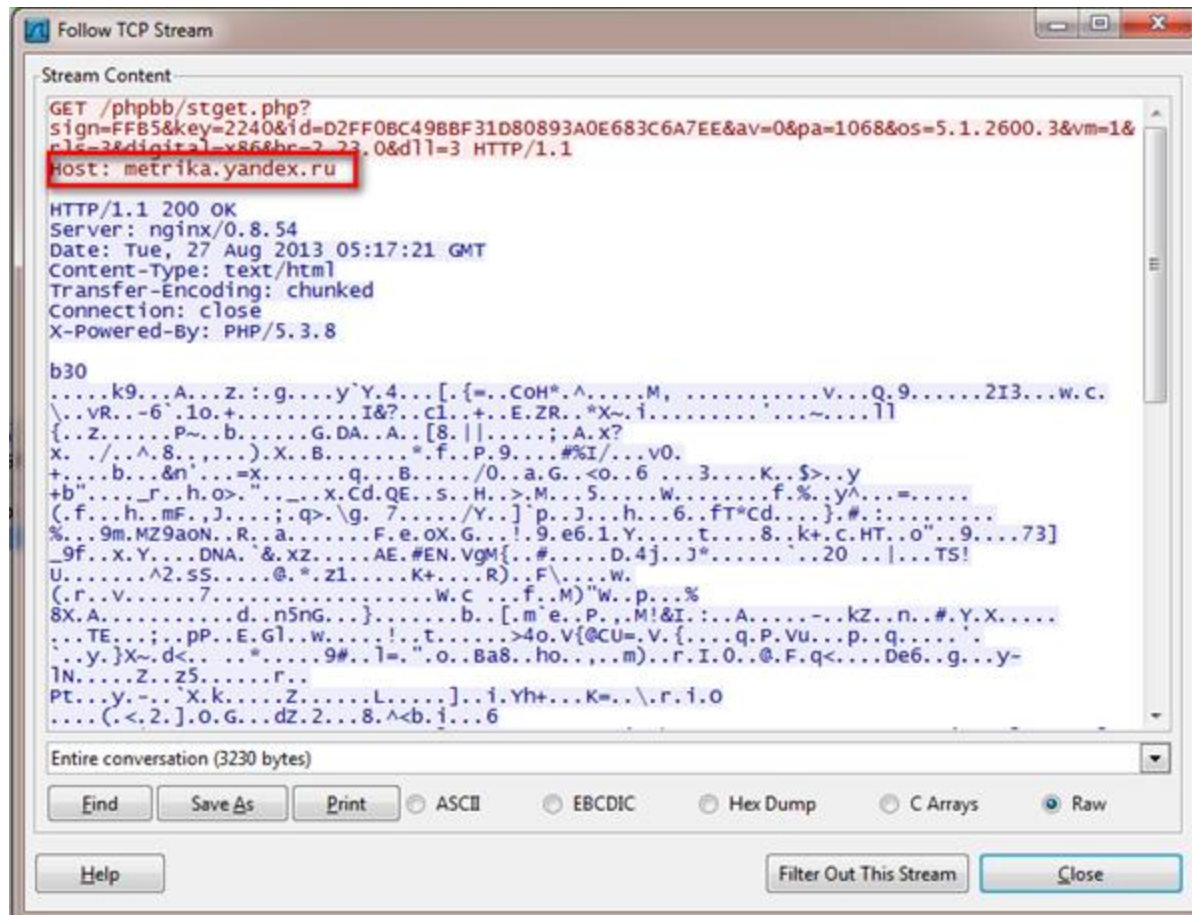
### GET vs. POST



# Header?

## ❖ HTTP 프로토콜의 일부

- Host 헤더: 서버를 식별 (가상호스팅과 연계하여 설명필요)



```
Stream Content
GET /phpbb/stget.php?
sign=FFB5&key=2240&id=D2FF0BC49BBF31D80893A0E683C6A7EE&av=0&pa=1068&os=5.1.2600.3&vm=1&
rlc=3&digital=x86&hc=2.23.0&d1l=3 HTTP/1.1
Host: metrika.yandex.ru

HTTP/1.1 200 OK
Server: nginx/0.8.54
Date: Tue, 27 Aug 2013 05:17:21 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: close
X-Powered-By: PHP/5.3.8

b30
.....k9...A...z...:g...y`Y.4...[.{=.COH*.^.....M,.....v...Q.9.....2I3...w.c.
\..vR...-6...1o...+.....I&?...c1...+.E.ZR...*X~.i.....'.....}}
{.z...P~..b...G.DA..A..[8..|].....;A.x?
x. /..^..8...).X..B.....*.f..P.9.....#%I/...v0.
+...b...&n'...=x.....q...B...../0..a.G...<o..6...3...K..$>..y
+b"..._r..h.o>..."_..x.Cd.QE..s..H..>..M...5...w.....f.%.y^...=....
(.f...h..mF..J...;..q>.\g. 7...../Y..]`p..J...h...6..fT*Cd....}.#.:.....
%...9m.MZ9aon..R..a.....F.e.oX.G...!.9.e6.1.Y....t...8..k+.c.HT..o"...9...73]
_9f..x.Y....DNA..&.xZ.....AE.#EN.VgM{..#.....D.4j..J*.....20...|...TS!
U.....^2.SS.....@.*.z1.....K+....R)..F\...w.
(.r..V.....7.....W.C...f..M)"w..p...%
8X.A.....d..n5nG...}.b...[.m'e..P...M!&I...A.....-kZ..n..#.Y.X....
...TE.....pP..E.Gl..w.....!..t.....>4o.V{@CU=.V.{...q.P.Vu...p..q....
...y..}X~.d<...*.9#.l=. ".o..Ba8..ho.....m)..r.I.O..@.F.q<...De6..g...y-
lN...Z..Z5.....r..
PT...y..-..X.k...Z.....L.....].i.Yh+...K=..\.r.i.o
....(<.2..].O.G...dZ..2...8..^<b.i...6

Entire conversation (3230 bytes)
Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw
Help Filter Out This Stream Close
```

# User Agent

## ❖ HTTP 프로토콜의 일부

- 브라우저의 종류를 식별

### Environment

Variable	Value
PATH	/usr/local/bin:/usr/bin:/bin
CTK_ERROR_DOCUMENT	404.html
DOCUMENT_ROOT	/www
HTTP_ACCEPT	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
HTTP_ACCEPT_CHARSET	ISO-8859-1,*,utf-8
HTTP_ACCEPT_ENCODING	gzip,deflate,bzip2
HTTP_ACCEPT_LANGUAGE	en-US,en
HTTP_REFERER	http://www.google.com/search?q=phpinfo+HTTP_USER_AGENT
HTTP_USER_AGENT	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.13 (KHTML, like Gecko) Chrome/0.2.149.27 Safari/525.13
REDIRECT_STATUS	200
REDIRECT_URL	/identifiant/info.php

# Referrer

## ❖ 상위 링크 추적을 위한 HTTP 헤더



# Cookie (중요)

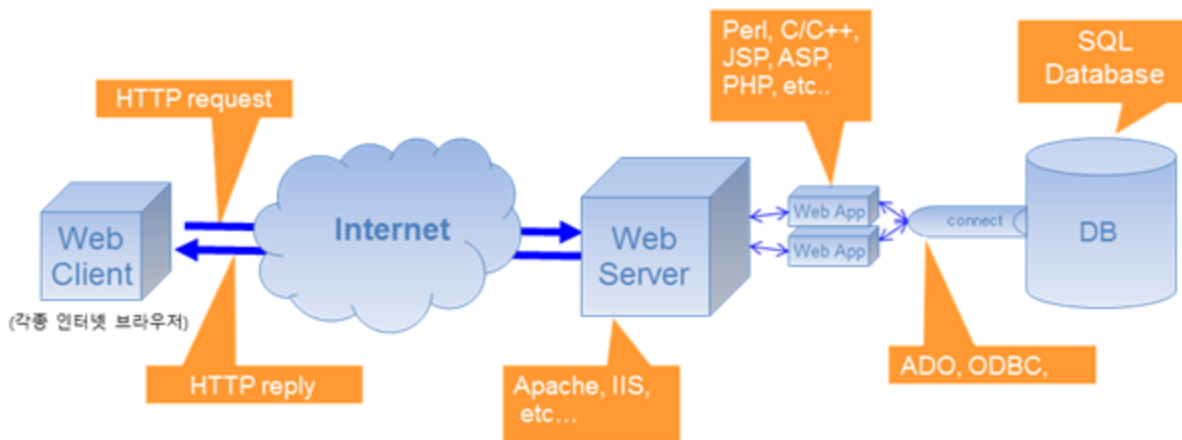
- ❖ 웹 브라우저를 식별
- ❖ 기타 여러가지 정보를 '브라우저'에 저장
- ❖ LocalStorage와의 차이?





# 웹 방화벽

## ❖ 웹 해킹에 집중된 방화벽



# Burp – 웹해킹 기본도구

The screenshot displays the Burp Suite application window. The top menu bar includes 'Burp', 'Intruder', 'Repeater', 'Window', and 'Help'. Below the menu is a toolbar with buttons for 'Target', 'Proxy', 'Spider', 'Scanner', 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Extender', 'Options', and 'Alerts'. A 'Filter: Showing all items' bar is present. The left sidebar shows a tree view of the target site 'http://www.google.com' with folders like 'advanced\_search', 'client\_204', 'history', 'images', 'imghp', 'intl', 'language\_tools', 'preferences', and 'search'. The main pane is divided into 'Contents' and 'Issues' tabs. The 'Contents' tab shows a list of HTTP requests with columns for Host, Method, URL, Params, and Status. The selected request is a GET request to 'http://www.google.com/search?q=pentestgeek&hl=en&gbv=1&oq=pentestgee...'. Below this, the 'Request' and 'Response' tabs are visible. The 'Request' tab shows the raw HTTP request details, including the method (GET), URL, host, user-agent, accept, accept-language, accept-encoding, dnt, referer, and cookie.

Host	Method	URL	Params	Status
http://www.google.c...	GET	/search?ie=ISO-8859-1&hl=en&source=hp&biw=&bih=...	✓	200
http://www.google.c...	GET	/search?q=pentestgeek&hl=en&gbv=1&oq=pentestgee...	✓	200
http://www.google.c...	GET	/xjs/_/js/k=xjs.hp.en_US.JrX4RoZaeBk.O/m=sb_he,d/r...		200
http://www.google.c...	GET	/client_204?&atyp=i&biw=1649&bih=742&ei=nzvhV9iy...	✓	204
http://www.google.c...	GET	/advanced_search		
http://www.google.c...	GET	/advanced_search?hl=en&authuser=0	✓	
http://www.google.c...	GET	/advanced_search?q=pentestgeek&hl=en&gbv=1&ie=U...	✓	

**Request Details:**

```
GET
/search?q=pentestgeek&hl=en&gbv=1&oq=pentestgeek&gs_l=heirloom-serp.3..0j0i30.56132.57;
heirloom-serp..1.10.373.28pXsfQweKk HTTP/1.1
Host: www.google.com
User-Agent: SNCAppSec2016
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Referer:
http://www.google.com/search?ie=ISO-8859-1&hl=en&source=hp&biw=&bih=&q=test&gbv=1&oq=t
.26166.0.26302.4.4.0.0.0.0.127.253.2j1.3.0....0...lac.1.34.heirloom-hp..2.2.126.3rCfcgJ
Cookie:
```

# URL 보는 법



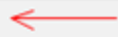
# urlencoding

## ❖URL 에 자주 사용되는 인코딩기법

### Submitted Form Data

Your input was received as:

text=Hello+G%C3%BCnter



URL Encoding result from plain text: "Hello Günter"

The server has processed your input and returned this answer.

# 아스키코드

## ❖컴퓨터가 글자를 메모리에 저장하는 방법

dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0	0	000	NULL	32	20	040	space	64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	a	012	LF	42	2a	052	*	74	4a	112	J	106	6a	152	j
11	b	013	VT	43	2b	053	+	75	4b	113	K	107	6b	153	k
12	c	014	FF	44	2c	054	,	76	4c	114	L	108	6c	154	l
13	d	015	CR	45	2d	055	-	77	4d	115	M	109	6d	155	m
14	e	016	SO	46	2e	056	.	78	4e	116	N	110	6e	156	n
15	f	017	SI	47	2f	057	/	79	4f	117	O	111	6f	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1a	032	SUB	58	3a	072	:	90	5a	132	Z	122	7a	172	z
27	1b	033	ESC	59	3b	073	;	91	5b	133	[	123	7b	173	{
28	1c	034	FS	60	3c	074	<	92	5c	134	\	124	7c	174	
29	1d	035	GS	61	3d	075	=	93	5d	135	]	125	7d	175	}
30	1e	036	RS	62	3e	076	>	94	5e	136	^	126	7e	176	~
31	1f	037	US	63	3f	077	?	95	5f	137	_	127	7f	177	DEL

www.alpharhms.com



# Base64

## ❖웹에서의 대표적인 인코딩기법

### Parameters Cont.

- ▶ [https://www.google.com/search?sxsrf=ALeKk00wo5D9cPJLi8ZkWCw\\_vUspW2m9Kw%3A1607543430005&ei=hSrRX\\_PcPOCr5NoPnsCpuAg&q=test&oq=test&gs\\_lcp=CgZwc3ktYWIQAziECCMQJziECCMQJzIFCAAQkQlyCwguELEDEMcBEK8BMgUIABCxAziOCC4QsQMqgwEQxwEQrwEyAggAMgIIADICCAyAggAOgQIABBDoggIABCxAxCDAToQCC4QsQMqgwEQxwEQowIQzoiCC4QsQMqgwE6CggAELEDEIMBEEM6CgguEMcBEKMCEEM6CAguEMcBEK8BOgsILhCxAxDHARCjAIDQyAFYjs0BYILSAWgAcAB4AIAbtAGIAakFkgEDMC40mAEAoAEBqgEHZ3dzLXdpesABAQ&sclient=psy-ab&ved=0ahUKEwizs-z41cHtAhXgFVvKfHR5gCocQ4dUDCA0&uact=5](https://www.google.com/search?sxsrf=ALeKk00wo5D9cPJLi8ZkWCw_vUspW2m9Kw%3A1607543430005&ei=hSrRX_PcPOCr5NoPnsCpuAg&q=test&oq=test&gs_lcp=CgZwc3ktYWIQAziECCMQJziECCMQJzIFCAAQkQlyCwguELEDEMcBEK8BMgUIABCxAziOCC4QsQMqgwEQxwEQrwEyAggAMgIIADICCAyAggAOgQIABBDoggIABCxAxCDAToQCC4QsQMqgwEQxwEQowIQzoiCC4QsQMqgwE6CggAELEDEIMBEEM6CgguEMcBEKMCEEM6CAguEMcBEK8BOgsILhCxAxDHARCjAIDQyAFYjs0BYILSAWgAcAB4AIAbtAGIAakFkgEDMC40mAEAoAEBqgEHZ3dzLXdpesABAQ&sclient=psy-ab&ved=0ahUKEwizs-z41cHtAhXgFVvKfHR5gCocQ4dUDCA0&uact=5)

Source	Text (ASCII)	M					a					n										
	Octets	77 (0x4d)					97 (0x61)					110 (0x6e)										
Bits		0	1	0	0	1	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0
Base64 encoded	Sextets	19					22					5					46					
	Character	T					W					F					u					
	Octets	84 (0x54)					87 (0x57)					70 (0x46)					117 (0x75)					

# 실습

## ❖ Base64 인코딩 / 디코딩 해보기

V1RJNWRWb3pTbWhrU0ZaeldWaFNjR0l5TlhwSlUwSXdZVWRzZWtsSGJlc  
EpTRkp2V2xOQ2VWcFhSbk5KUnpGc1l6Tk9hRm95VlVzSwo=

Value	Character	Value	Character	Value	Character	Value	Character
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

# 웹 어플리케이션의 보안이슈들

---

## ❖Directory Listing

## ❖Local File Inclusion (LFI)

## ❖SQL Injection

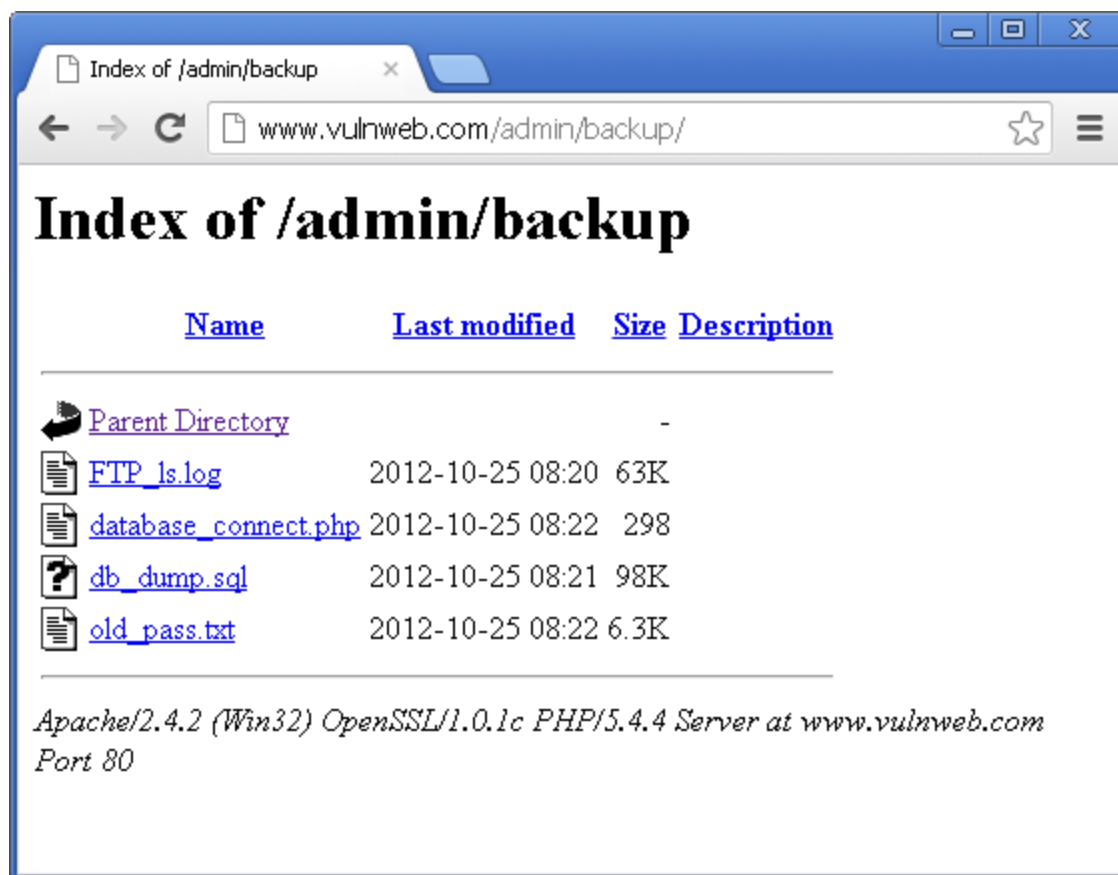
- 1=1 attack
- blind SQL injection
- prepared statement

## ❖Cross Site Scripting (XSS)

- reflected XSS
- cross site request forgery (CSRF)



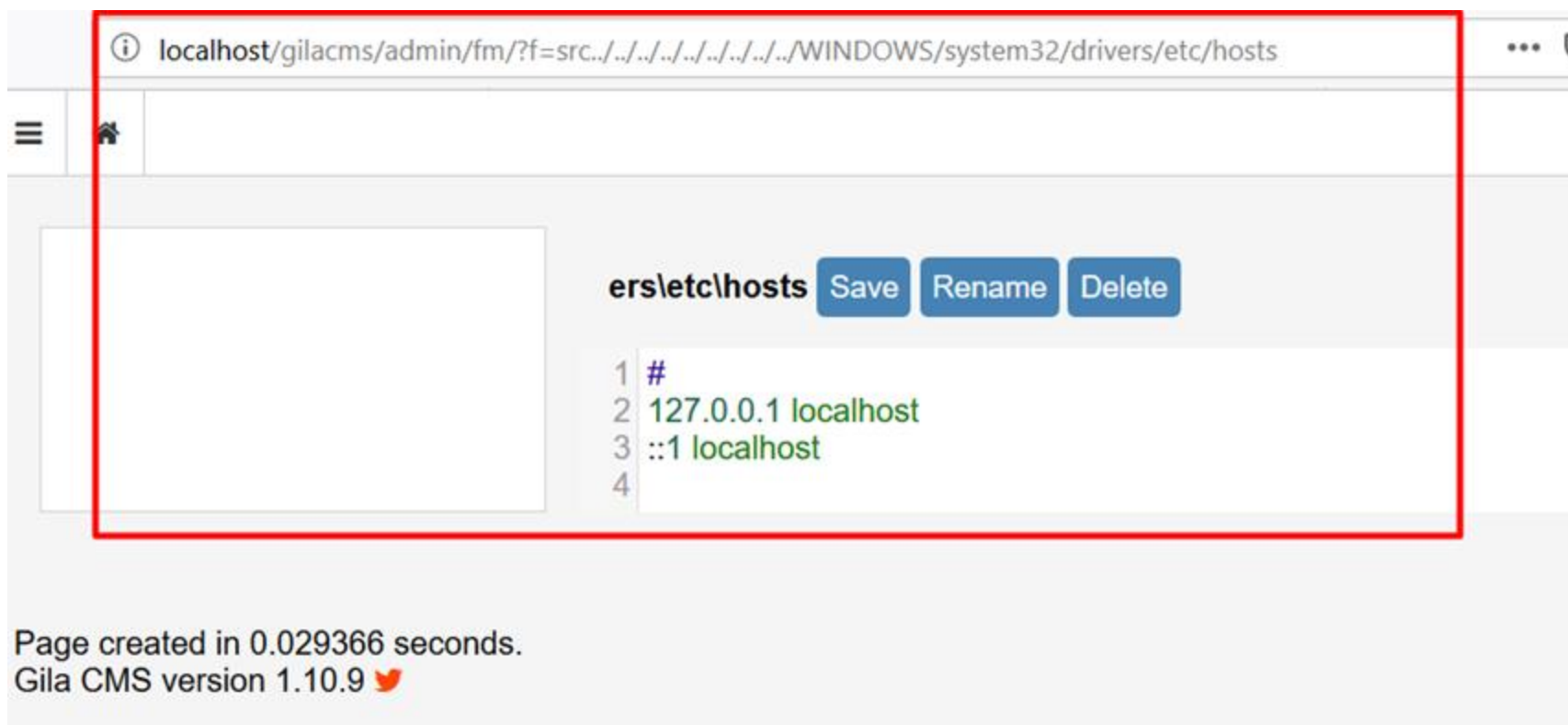
# Directory Listing 예



# Local File Inclusion

## ❖ 웹 어플리케이션 사용자가 비인가된 파일을 접근할 수 있음

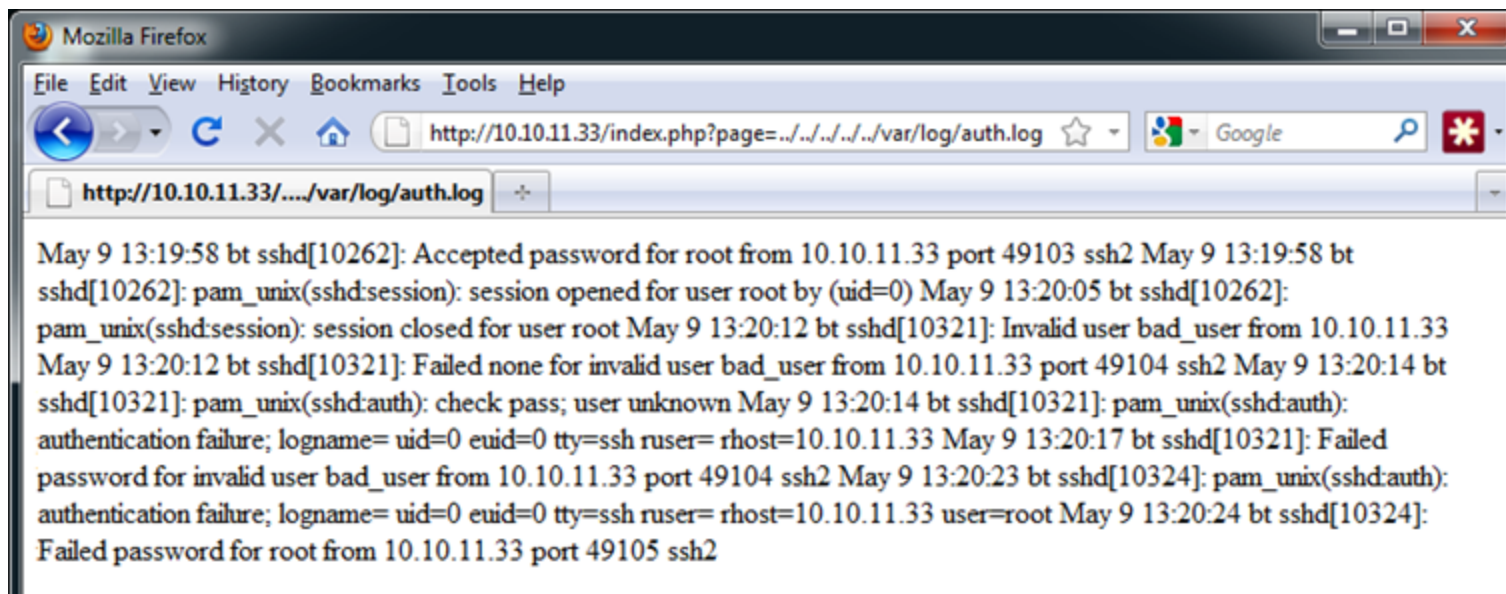
- 다운로드 취약점, 또는 파라미터 조작을 통해 다운로드되지 않아야 하는 파일이 다운로드됨



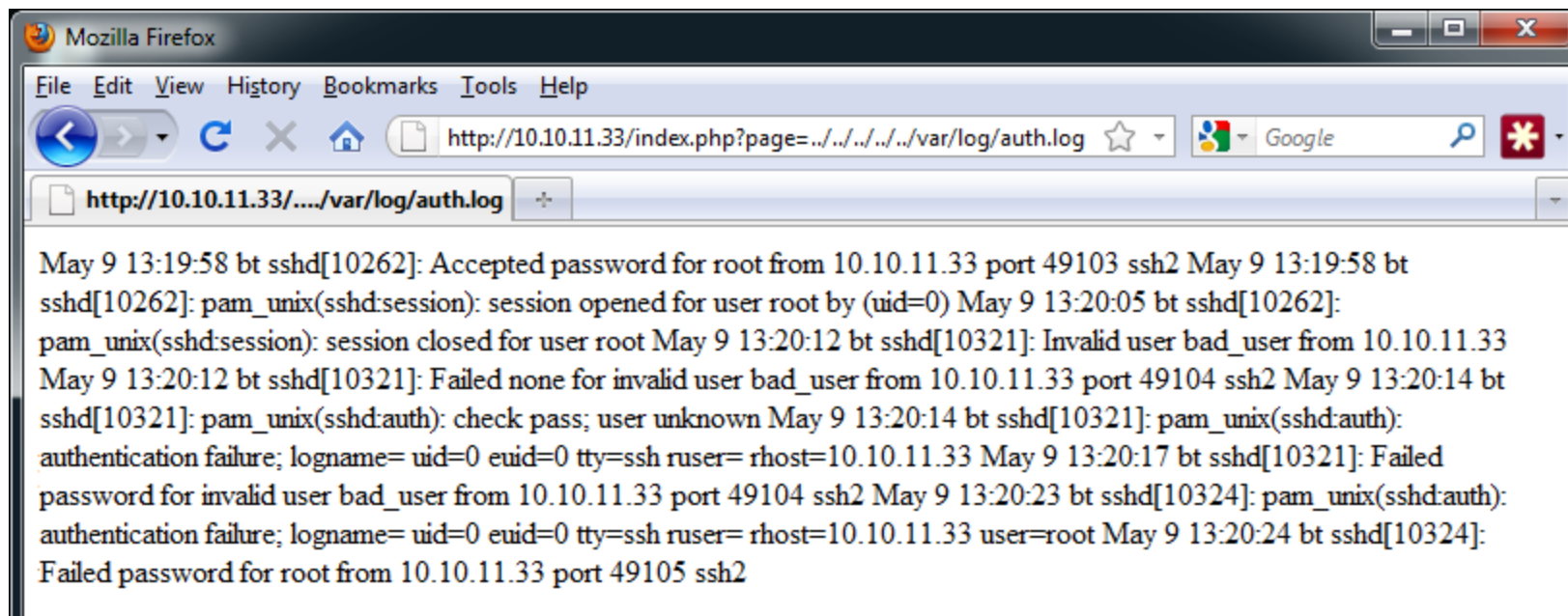
# LFI 취약코드 예시

## ❖ 웹 어플리케이션 파라미터 조작을 통해 비인가 파일 접근

- ?page=1.txt
- ?page=2.txt
- ?page=3.txt
- ✓ \$fp = fopen(\$\_GET[\$page], "r"); ....??
- ✓ /var/www/html



# LFI 결과 예시

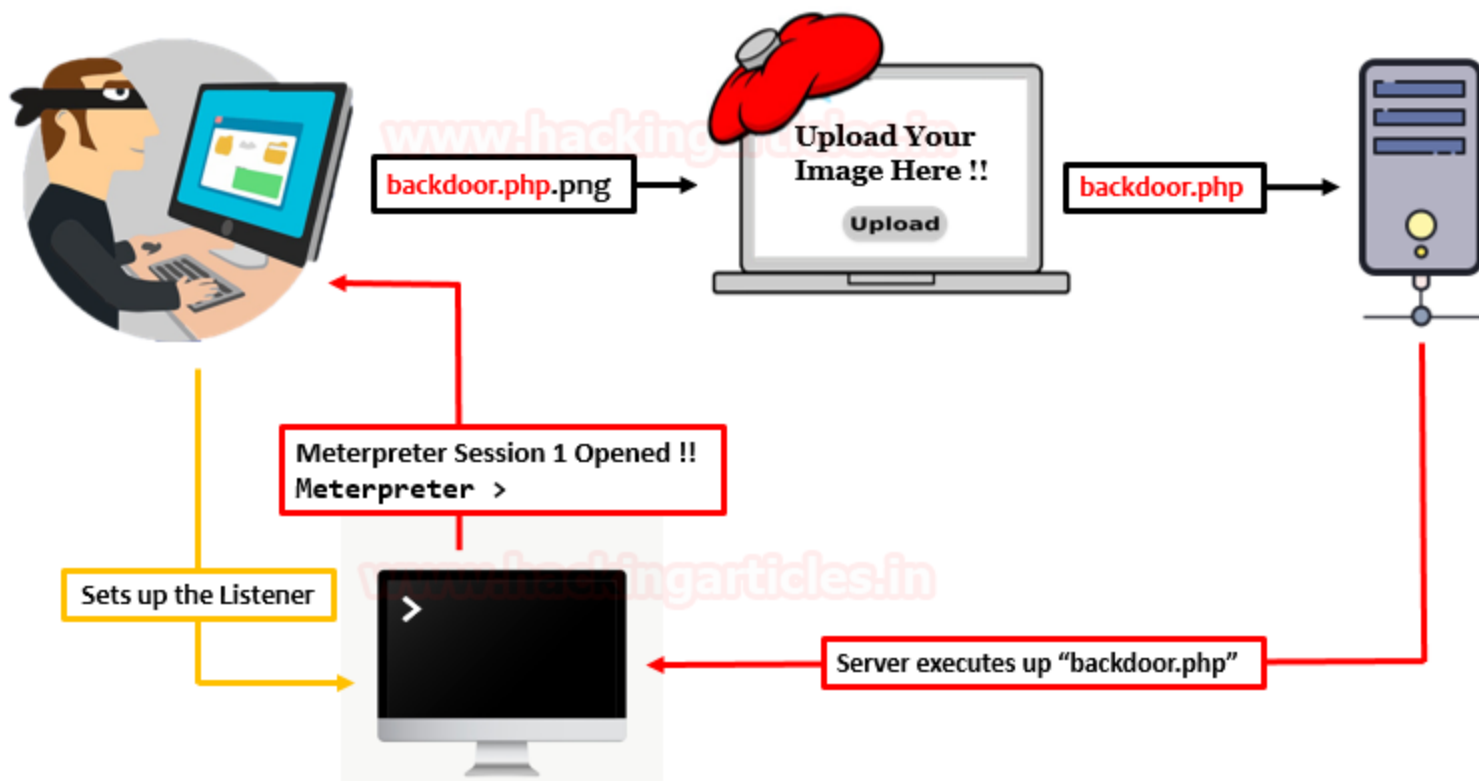


```
May 9 13:19:58 bt sshd[10262]: Accepted password for root from 10.10.11.33 port 49103 ssh2 May 9 13:19:58 bt  
sshd[10262]: pam_unix(sshd:session): session opened for user root by (uid=0) May 9 13:20:05 bt sshd[10262]:  
pam_unix(sshd:session): session closed for user root May 9 13:20:12 bt sshd[10321]: Invalid user bad_user from 10.10.11.33  
May 9 13:20:12 bt sshd[10321]: Failed none for invalid user bad_user from 10.10.11.33 port 49104 ssh2 May 9 13:20:14 bt  
sshd[10321]: pam_unix(sshd:auth): check pass; user unknown May 9 13:20:14 bt sshd[10321]: pam_unix(sshd:auth):  
authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.10.11.33 May 9 13:20:17 bt sshd[10321]: Failed  
password for invalid user bad_user from 10.10.11.33 port 49104 ssh2 May 9 13:20:23 bt sshd[10324]: pam_unix(sshd:auth):  
authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.10.11.33 user=root May 9 13:20:24 bt sshd[10324]:  
Failed password for root from 10.10.11.33 port 49105 ssh2
```

# 업로드 취약점

## ❖ 업로드 되지 않아야 하는 파일이 오류로 업로드 됨

- 예) 웹스크립트를 그림파일로 가장하여 업로드



# 웹셸

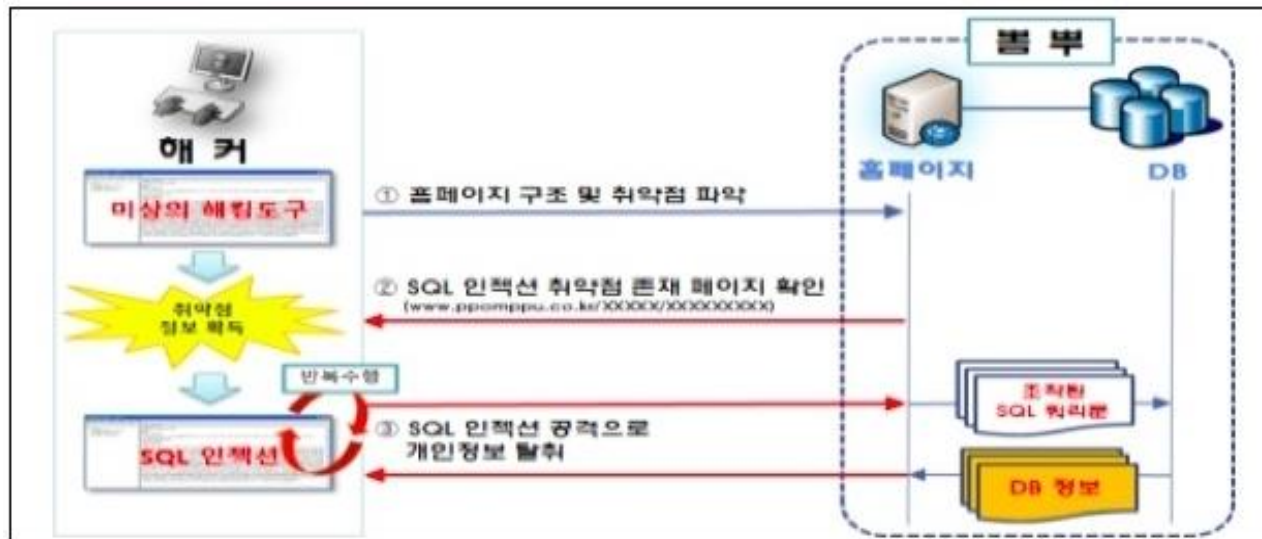
- ❖ 웹해킹에 성공한 공격자가 최종적으로 주입
- ❖ 웹 사이트에 자신의 제어프로그램을 업로드
- ❖ 주로 PHP 로 구성된 백도어 프로그램의 형태
  - PHP 웹셸이 가장 일반적
    - ✓ PHP -> 웹프로그래밍에서의 C 언어 와 유사

```
root@kali:~# cd /usr/share/webshells/php ↵
root@kali:/usr/share/webshells/php# ls -al
total 44
drwxr-xr-x 3 root root 4096 Jul 23 15:25 .
drwxr-xr-x 8 root root 4096 Jul 23 15:26 ..
drwxr-xr-x 2 root root 4096 Jul 23 15:25 findsocket
-rw-r--r-- 1 root root 2800 Jul 17 11:45 php-backdoor.php
-rwxr-xr-x 1 root root 5491 Jul 17 11:45 php-reverse-shell.php
-rw-r--r-- 1 root root 13585 Jul 17 11:45 qsd-php-backdoor.php
-rw-r--r-- 1 root root 328 Jul 17 11:45 simple-backdoor.php
root@kali:/usr/share/webshells/php#
```

# SQL Injection

## ❖ 데이터베이스 쿼리언어의 오류로 인한 정보탈취

- SQL 의 개념을 선행해서 이해해야함.

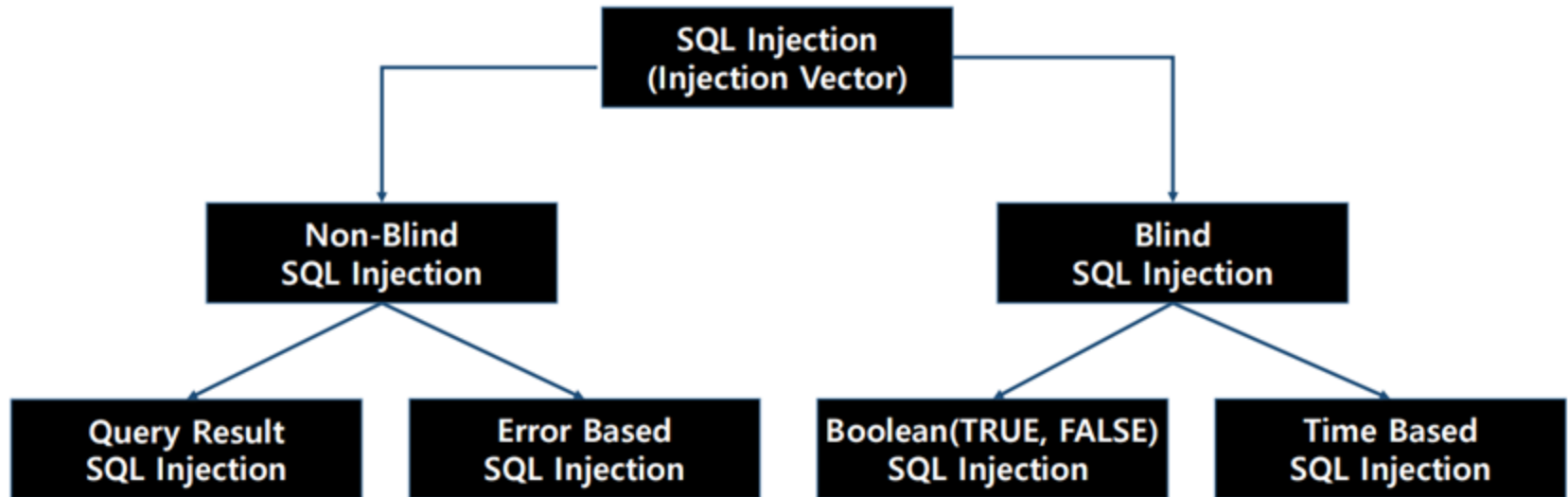


◇ 뽀뿌 해킹사고는 기초적인 해킹 수법인 'SQL 인젝션'이 쓰인 것으로 최종 확인됐다. [자료=미래부]

# SQL Injection 의 종류

❖ True/False 기반

❖ Timing 기반





# SQL Injection

## ❖ DBMS 의 SQL 쿼리문을 웹 어플리케이션 사용자가 조작

- SQL Injection
  - ✓ 로그인 우회
  - ✓ DB 정보 탈환
- Blind SQL Injection
  - ✓ DB 정보 탈환 (더 높은 난이도)
- 요약
  - ✓ 로그인 우회 가능
  - ✓ 데이터베이스 유출 가능

### SQL Injection.

User-Id:   
Password:

`select * from Users where user_id= 'srinivas' and password = 'mypassword'`

User-Id:   
Password:

`select * from Users where user_id= ' OR 1= 1; /*' and password = '*/--'`

swizardb.blogspot.com

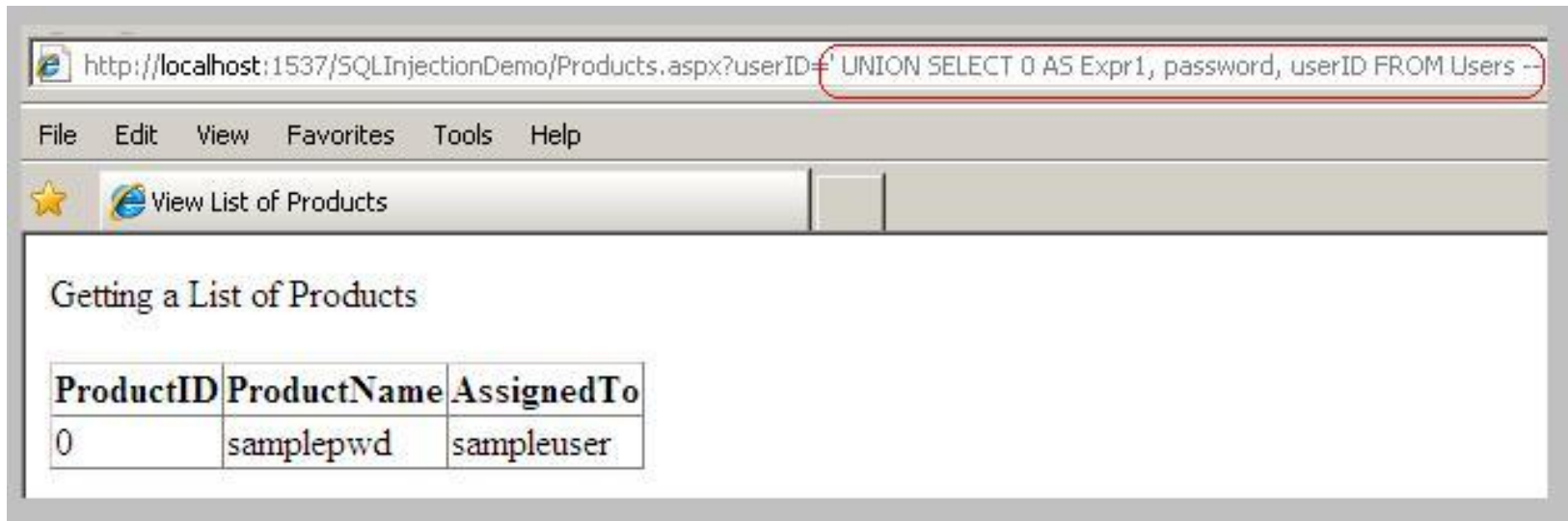
# 로그인 우회 코드예시 (ASP)

```
... <% function Login( connection ) {  
    var username = Request.form("username");  
    var password = Request.form("password");  
    var rso = Server.CreateObject("ADODB.Recordset");  
  
    var      sql      =      "select      *      from      pubs.guest.sa_table      \  
      where      username      =      "      +      username      +      "      and      \  
      password = " + password + "";  
  
    rso.open(sql, connection); //perform query  
  
    if (rso.EOF) //if record set empty, deny access  
    { rso.close();  
    %> <center>ACCESS DENIED</center> <%  
    } else { //else grant access  
    %> <center>ACCESS GRANTED</center> <%  
    // do stuff here ...
```

# SQL Injection

## ❖ DB 개인정보 유출

- Union 을 통한 추가 SELECT



# Prepared Statement

## ❖SQL 인젝션의 효과적인 방어

- Dynamic SQL Query Generation..?

### Query Execution Phases

Beauty of Prepare Statement .



1. Replaces ? with **user data**

After replacement, Query string formed is not compiled again.

If Query string is not compiled again, it means, if user data contains SQL then also it will not run and will be treated as pure data.

Query stored in Cache is pre-compiled, So it doesn't has to compile again, only replace placeholders and execute query.

# 다양한 SQL 인젝션 방법

---

## ❖ Blind SQL Injection

- 보통의 SQL 인젝션과의 차이는?
  - ✓ Side Channel Attack
  - ✓ 데이터를 간접적으로 탈취

## ❖ 쿼리 인젝션의 결과만 알수있음

- 데이터를 직접 볼수없음 (thus, blind)

# SELECT 문을 활용

## ❖참고: 키워드들에 대소문자 구분 없음

- SELECT == select == sElEcT

다음 예제는 Reservation 테이블에서 Name 필드와 RoomNum 필드만을 선택하는 예제입니다.

예제

```
SELECT Name, RoomNum  
FROM Reservation;
```

실행 결과

Name	RoomNum
홍길동	2014
임꺽정	918
장길산	1208
홍길동	504

쿼리의 결과를 통해 DATA 를 직접 확인

# SELECT 문을 활용

❖ `select * from 어찌고저찌고 where .....`

## Blind SQL Injection Attack



원하는 데이터 SELECT 의 결과를 직접 볼수없음  
(blind)

# Where 절에 대해서

## ❖ Select .... From [테이블] where [조건]

- 테이블의 레코드들중 조건에 부합되는 것들만 가져옴
  - ✓ 예) ID = 1234 -> ID=1234 의 조건이 True 를 만족하는 레코드들
  - ✓ 예) ID = 1234 or True -> ID=1234 or True 의 조건이 True 를 만족하는 레코드들

## ❖ 하나의 레코드만 사용하려는데 여러 개의 레코드를 가져오면?

- 예) select \* from user where id=??? and pw=???
  - ✓ 특정 ID/PW 를 만족하는 레코드를 가져와서 처리하려고 함
- 여러 개의 레코드중 가장 위에 있는 레코드 사용
  - ✓ 테이블에서 가장 먼저 찾게되는 레코드



# 블라인드 SQL 인젝션

## ❖ 쿼리의 True/False 여부만 알수있다

- 스무고개와 같은개념

## ❖ Where 절의 조건을 True/False 로 결정?

- False: 레코드를 안가져옴
- True: 레코드를 1개이상 (전체) 가져옴



# Blind SQL Injection

❖ Press releases are accessed with "pressRelease.jsp?id=5"

❖ A SQL query is created and sent to the database:

- "select title, description FROM pressReleases where id=5"

- ✓ 99999 or id=4 and substring(password, 1, 1)==CHR(0x41) --

no	title	description	password
2	hello	....	1234
3	bye	..	!@#\$asdf
4	apple	.....	zxcvQQ
5	banana	.....	77777

# 예시

## ❖게시판 글 열람시

- 예) <https://bbs.hello.com/board/viewpage.php?no=1324>  
✓ 1324번 게시물 열람요청

## ❖GET 파라미터를 통해 열람하고자하는 글 번호 전달

## ❖아래와같은 SQL 문을통해 DB 를 쿼리할것으로 추정

- `select * from BOARD where no=1324`  
✓ 만약 인젝션 필터링이 없다면?

## ❖아래와 같이 쿼리조작 가능

- `select * from BOARD where no=1324 and 1=2`  
✓ 결과는?
- `select * from BOARD where no=1324 or 1=2`  
✓ 결과는?

# sub 쿼리

## ❖ where 절의 조건속에서 다시 select 가 가능

- 예) select \* from bbs where id = (select id from user where name='haha')

## ❖ Select 내부에서 다시 Select

- outer query
- sub query

Outer Query

SELECT lastname, firstname  
FROM employees  
WHERE officeCode IN

Subquery or Inner Query

(SELECT officeCode  
FROM offices  
WHERE country = 'USA')

# Blind SQL 인젝션

## ❖ select 의 where 절을 제어할수있다?

- DB 내부의 정보에 대해 무엇이든 물어보고 True/False 형태로 답을 얻는다

## ❖ 무엇이 질문 가능한가?

- 회원번호 10번의 비밀번호의 길이는 8보다 큰가?  
✓ where no=1324 and length((select password from MEMBERS where uid=10)) > 8 --

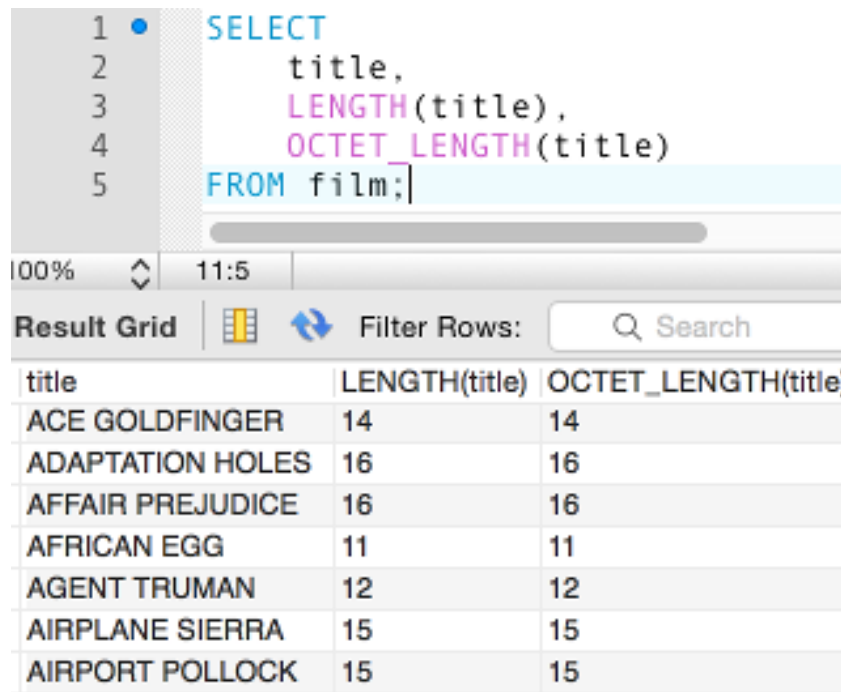
## ❖ And 와 Or 의 연산자 처리방식 (중요)

- and -> 곱셈처럼 생각
- or -> 덧셈처럼 생각
- $1+2*3 = 7$
- $\text{False or False and True} = \text{False or (False and True)} = \text{False or False} = \text{False}$
- $\text{True and True or False} = (\text{True and True}) \text{ or False} = \text{True or False} = \text{True}$
- $\text{id='admin' or '1'='2' and pw='1234'}$  -> 결과는?

# Length 함수

## ❖ 문자열의 바이트 길이를 가져옴

- `select length('hello') -> 5`



```
1 SELECT
2     title,
3     LENGTH(title),
4     OCTET_LENGTH(title)
5 FROM film;
```

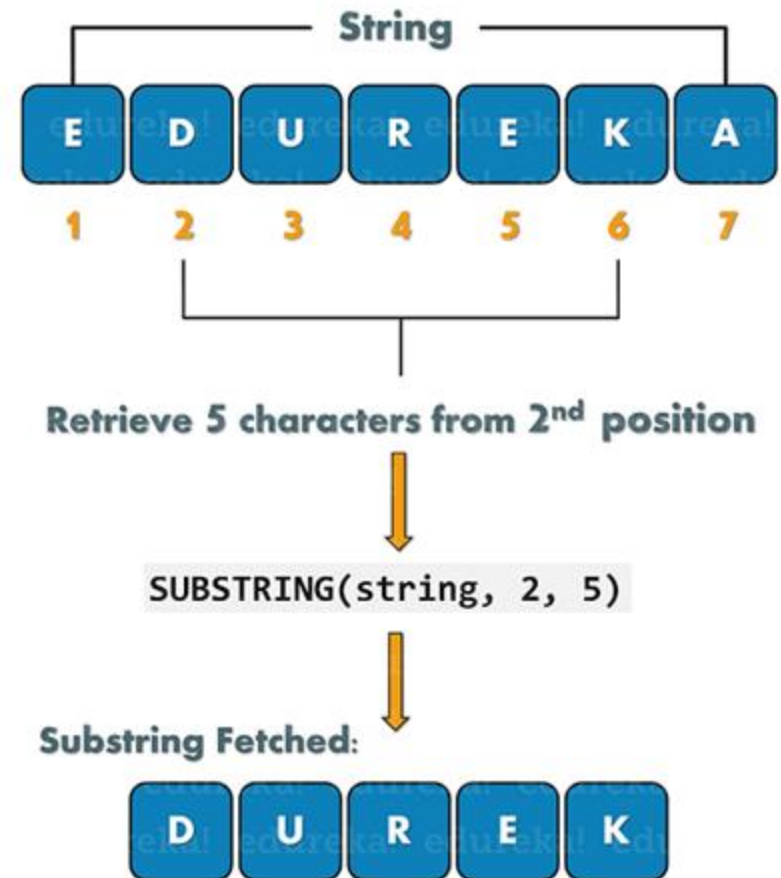
title	LENGTH(title)	OCTET_LENGTH(title)
ACE GOLDFINGER	14	14
ADAPTATION HOLES	16	16
AFFAIR PREJUDICE	16	16
AFRICAN EGG	11	11
AGENT TRUMAN	12	12
AIRPLANE SIERRA	15	15
AIRPORT POLLOCK	15	15

# substring

## ❖ 문자열의 부분문자열을 가져옴

## ❖ 문법

- substring(문자열, 시작위치, 길이)
  - ✓ select substring('hello', 2, 1) -> 'e'



# Time based SQL Injection

## ❖ True/False 결과도 알수없다면?

```
POST /waimai/index.php?m=public&a=checkemail HTTP/1.1
Host: 127.0.0.1
Content-Length: 49
Accept: application/json, text/javascript, */*; q=0.01
Origin: http://127.0.0.1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_2) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://127.0.0.1/waimai/index.php?m=public&a=register
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: 7eC9_2132_saltkey=f2CQcJKg; 7eC9_2132_lastvisit=1547529422;
__tins__16868462=%7B%22sid%22%3A%201547619737479%2C%20%22vd%22%3A
%201%2C%20%22expires%22%3A%201547621537479%7D; __51cke__=;
__51laig__=1; PHPSESSID=e922b0f3eed4e626b67add2710abd84f
Connection: close
```

param[0]=exp&param[1]=and sleep(5)&name=useremail

```
HTTP/1.1 200 OK
Date: Thu, 17 Jan 2019 07:14:48 GMT
Server: Apache
X-Powered-By: PHP/5.6.37
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 43
```

```
{
    "info": "",
    "status": "y"
}
```

357 bytes | 5,107 millis





# Cross Site Scripting

---

## ❖ XSS

- CSS 와의 혼동 유의
  - ✓ Cascading Style Sheet
- 자바스크립트 (JavaScript 와 밀접한 관계의 보안)

## ❖ 종류

- Reflected XSS
- Stored XSS
- Universal XSS
- ...

# JavaScript

## ❖웹사이트를 동적으로 만들어주는 요소

- WebSocket, AJAX 등 여러가지 웹 동적 구성 기법 존재
- Web Assembly 등



# Reflected XSS

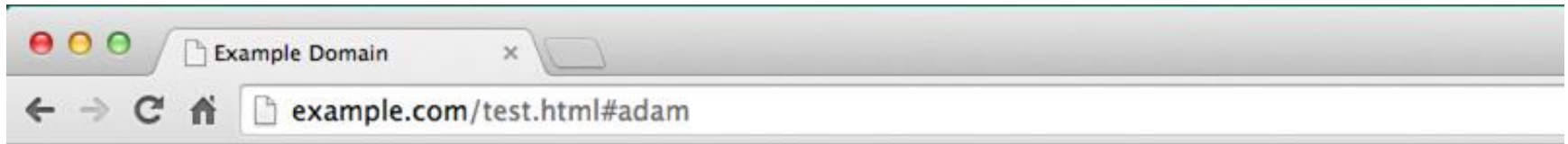
---

`http://example.com/test.html#adam`

```
<html>
  <body>
    <script>
      var name = location.hash;
      document.write("hello " + name);
    </script>
  </body>
</html>
```

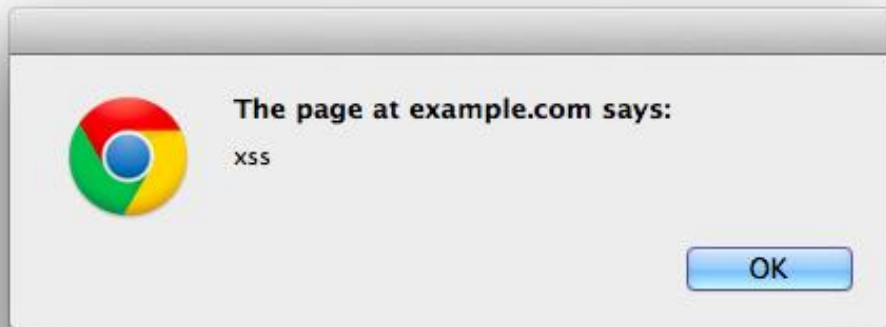
# Reflected XSS

---



Hello adam

# Reflected XSS



# XSS 가 위험한 이유

---

## ❖1. 쿠키 탈환

- 로그인 세션
- 기타 개인정보
- CSRF Token 도 탈취가능

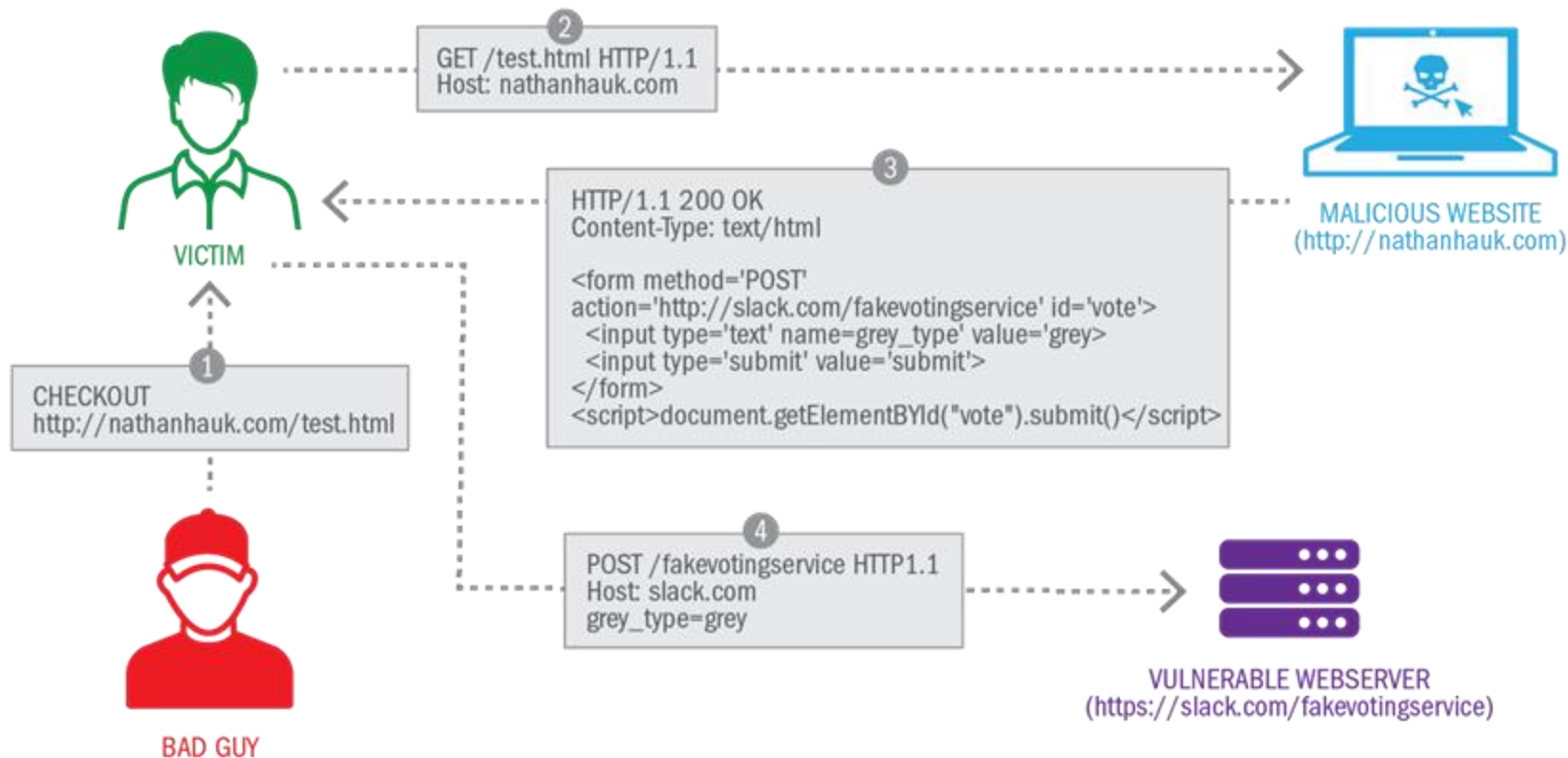
## ❖2. 키로깅

- 마우스/키보드 이벤트 후킹

## ❖3. 피싱유도

- 가짜 form 및 가짜 화면제작

# CSRF (Cross Site Request Forgery)



# CSRF Token (방어)

서버 측에서 Random 생성한 CSRF Token 을 삽입하여 방어.

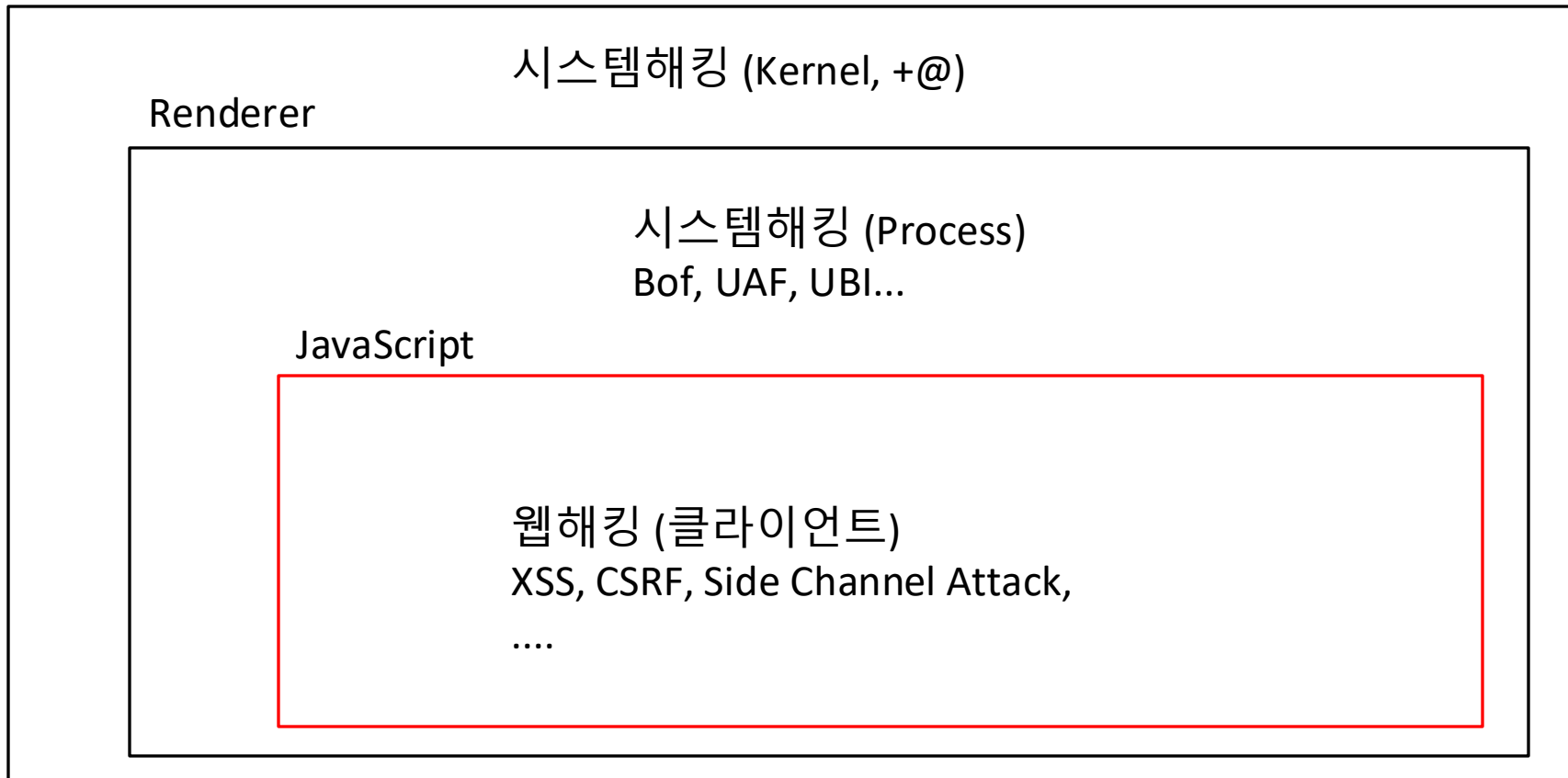
```
<input type="hidden" name="csrf-token" value="ClwNZNIR4XbisJF39l8y....." />
```

```
1 <!DOCTYPE html>
2 <!--[if lt IE 7]>
3 <html class='no-js lt-ie9 lt-ie8 lt-ie7' lang='en'></html>
4 <![endif]-->
5 <!--[if IE 7 ]>
6 <html class='no-js lt-ie9 lt-ie8' lang='en'></html>
7 <![endif]-->
8 <!--[if IE 8 ]>
9 <html class='no-js lt-ie9' lang='en'></html>
10 <![endif]-->
11 <!--[if (gt IE 8)]><!--> <html lang="en" class="no-js"> <!--<![endif]-->
12 <head prefix='og: http://ogp.me/ns# fb: http://ogp.me/ns/fb# mixlrcom: http://ogp.me/ns/fb/mixlrcom#>
13 <meta name="csrf-param" value="authenticity_token"/>
14 <meta id="csrf_token" name="csrf-token" value="NMQdn8yTMAFbdIFS0we6pyp5sl7J89Lnb6ixDL32g8I="/>
15 <meta content="Browse popular live broadcasts on Mixlr. Mixlr is a platform for social live audio." name="Description" />
16 <meta content="broadcast, audio, live, sound, tracks, music, crowd, mixlr" name="keywords" />
17 <meta content="NOARCHIVE" name="googlebot" />
18 <meta content="width=device-width,initial-scale=1,user-scalable=no" name="viewport" />
19 <meta name="apple-mobile-web-app-capable" content="yes" />
20 <meta name="mixlr-accept-language-value" content="en" />
21 <meta name="apple-itunes-app" content="app-id=583705714, affiliate-data=, app-argument=">
22 <title>
23 Popular - live broadcasts popular now | Mixlr
24 </title>
```



# 웹 브라우저 보안계층

Sandbox



---

# Q/A