

2024-1 인터넷보안실무

# 인터넷 해킹과 보안 4판



## Chapter 01 인터넷과 웹의 이해

### 목차

- 01 인터넷의 탄생
- 02 인터넷 프로토콜
- 03 인터넷 거버넌스
- 04 웹의 탄생
- 05 HTTP의 기본 개념
- 06 웹 애플리케이션 기술

## 학습목표

- 인터넷과 웹의 탄생 배경과 발전 과정을 안다.
- 인터넷과 웹 서비스에 사용되는 주요 프로토콜에 대해 설명할 수 있다.
- 인터넷을 유지 및 관리하는 기구를 나열할 수 있다.
- 웹 애플리케이션에 사용되는 기술에 대한 기본 지식을 갖춘다.

# 01

## 인터넷의 탄생

01 인터넷의 탄생

2024-1  
인터넷보안실무

■ 인터넷의 탄생

- 인터넷은 멀리 떨어진 대학 연구소에 정보를 전달하는 용도로 시작
- 1969년 10월 29일, 미국 국방부 산하 고등연구국 ARPA의 연구용 네트워크인 ARPANET을 통해 UCLA의 레너드 클라인록 교수가 UCLA의 컴퓨터에서 스탠퍼드대학교 SRI 연구소의 컴퓨터로 메시지를 전송하는 데 성공
- 이후 ARPANET이 일반에 공개되어 TCP/IP 프로토콜로 연결되면서 인터넷이 본격적으로 발전하기 시작

■ 대한민국 인터넷의 탄생

- 1982년, 서울대학교와 KIET(전자통신연구소의 전신)가 TCP/IP로 SDN을 시작
- 1988년, 연구 전산망 기본 계획이 확정되어 교육망과 BITNET 연결
- 1994년, 한국통신이 KAIST와 연구소 등에 학술 및 교육 정보 교류용으로 제공한 하나망을 일반에 개방하여 코넷(KORNET)을 시작

Page 5

02

인터넷 프로토콜

02 인터넷 프로토콜

2024-1  
인터넷보안실무

■ 프로토콜(Protocol)

- 컴퓨터 간에 정보를 원활하게 교환하기 위해 상호 간에 정한 여러 가지 통신 규칙과 방법에 대한 약속 또는 규약

■ 프로토콜의 세 가지 요소

- 구문(Syntax): 데이터의 형식이나 신호로, 부호화 방법 정의
- 의미(Semantics): 정확한 정보 전송을 위한 전송 제어와 오류 제어 방법 정의
- 순서(Timing): 송신자와 수신자 간 혹은 양단(End-to-End)의 통신 시스템, 망 사이의 통신 속도나 순서 정의

Page 7

02 인터넷 프로토콜

2024-1  
인터넷보안실무

■ TCP/IP(Transmission Control Protocol/Internet Protocol)

- 가장 많이 사용되는 프로토콜
- 프로토콜에 대한 상세한 내용은 RFC라는 문서를 통해 공개

■ RFC(Request for Comments)

- 국제인터넷표준화기구(IETF)에서 만듦
- 인터넷에서 기술을 구현하는 데 필요한 상세 절차와 기본 틀을 제공하는 기술 관련 문서

Page 8

# 03

## 인터넷 거버넌스

- 국제인터넷주소관리기구(ICANN)
- 인터넷의 기술적인 문제를 관리
  - 인터넷의 유일한 식별자인 도메인 이름, 시스템(DNS)과 IP 주소, 프로토콜 번호와 매개변수 배정 등을 관리
  - DNS 루트 네임 서버 시스템의 개선 및 운영 담당



그림 1-8 국제인터넷주소관리기구 홈페이지

03 인터넷 거버넌스

2024-1  
인터넷보안실무

■ 인터넷할당번호관리기관(IANA)

- 인터넷이 생긴 초기에 사용자들에게 인터넷 주소를 할당하기 위해 국방정보청 (DISA)이 만듦
- 현재는 인터넷소사이어티(Internet Society, ISOC)의 산하 기관
- IANA의 가장 중요한 기능은 DNS Root Zone을 관리하는 것

Page 11

03 인터넷 거버넌스

2024-1  
인터넷보안실무

■ 인터넷소사이어티(ISOC)

- 인터넷의 이용과 기술에 관한 국제적인 협조와 협력을 촉진하기 위해 1992년 에 설립된 비영리 국제기구
- 세 개의 핵심 조직인 국제인터넷표준화기구(IETF), 인터넷기술관리그룹(IESG), 인터넷아키텍처위원회(IAB)를 지원

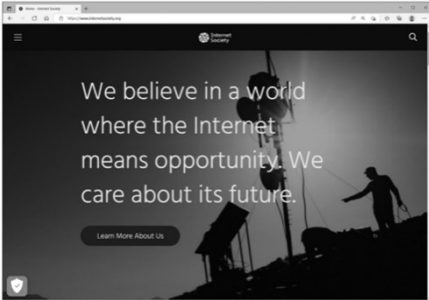


그림 1-9 인터넷소사이어티 홈페이지

Page 12

## 03 인터넷 거버넌스

2024-1  
인터넷보안실무

## ■ 인터넷소사이어티(ISOC)

- IETF : 인터넷의 운영, 관리, 개발에 대해 협의하고 프로토콜과 구조적인 사안을 분석하는 인터넷 표준화 기구

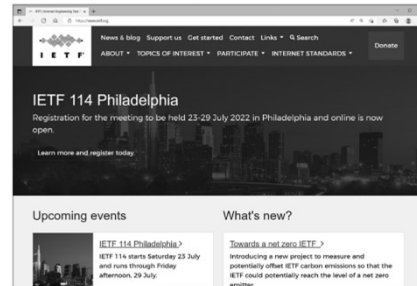


그림 1-10 국제인터넷표준화기구 홈페이지

- IESG : 인터넷의 기술적인 문제를 해결할 목적으로 설립된, 인터넷아키텍처 위원회 하부 조직
- IAB : 인터넷소사이어티의 감독 단체로, 인터넷의 방침이나 장기적인 기획 및 기술 정책 등을 심의하고 결정

Page 13

## 03 인터넷 거버넌스

2024-1  
인터넷보안실무

## ■ 월드와이드웹컨소시엄

- 월드와이드웹(이하 웹이라 칭함), 웹 브라우저, 웹 서버 기술의 표준화를 추진 하기 위해 교육 · 연구 기관 및 관련 회사들이 모여서 만든 단체
- 보통 WWW 컨소시엄 또는 줄여서 W3C라고도 부름
- W3C에서 정한 기술 표준은 국제적인 표준으로 인정받기 때문에 많은 기업과 연구가들이 활발한 활동을 펼치고 있음



그림 1-11 월드와이드웹컨소시엄 홈페이지

Page 14

03 인터넷 거버넌스

2024-1  
인터넷보안실무

■ 국제전기통신연합(ITU)

- 1865년에 설립되어 1947년부터는 UN산하 기관으로 활동
- ITU의 전기통신 표준화 부문 ITU-T은 인터넷에 영향을 미칠 수 있는 관세 문제를 다루고, 정보통신 네트워크에 대한 기술적인 표준과 운영 표준을 발행
- 2012년 12월 17일, ITU가 개최한 국제 전기통신 세계 회의WCIT에서 국제 통신 규약(ITR) 개정안 통과 → 인터넷에 대한 통제가 자발적인 기구에 의한 것에서 중앙 통제로 강화될 것임을 예고



그림 1-12 국제전기통신연합 홈페이지

04  
웹의 탄생



04 웹의 탄생

2024-1  
인터넷보안실무

■ 월드와이드웹

- 인터넷에 연결된 컴퓨터들이 하이퍼텍스트 형식으로 표현된 다양한 정보를 효과적으로 이용할 수 있도록 구성한 전 세계적인 시스템
- 간단히 웹이라고 부름
- 2000년 이전만 해도 웹을 통해 회사를 홍보하는 곳이 그리 많지 않았지만 지금은 대기업, 관공서, 개인 쇼핑몰에 이르기까지 수많은 웹 사이트가 존재
- 오늘날 웹이 없는 세상은 상상할 수도 없음

Page 17

04 웹의 탄생

2024-1  
인터넷보안실무

■ 웹의 탄생

- 1989년 3월 13일, 유럽입자물리연구소(CERN)에 근무하던 소프트웨어 공학자 팀 버너스 리가 과학자들 사이에 쉽게 정보를 주고받기 위한 목적으로 정보 관리 제안을 발표(최초의 인터넷 기반 하이퍼텍스트 프로젝트)
- 이후 1990년에 하이퍼텍스트 브라우저와 편집기가 개발되고 URL, HTTP, HTML이 차례대로 설계됨
- 1991년 8월 팀 버너스 리는 월드 와이드웹의 개념을 포함한 사이트를 일반인에게 최초로 공개하고, 로열티를 포기

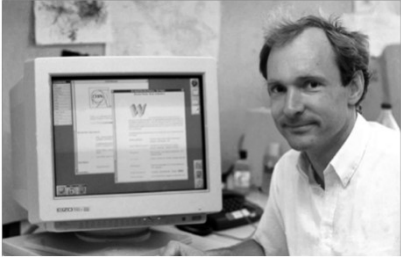


그림 1-13 팀 버너스 리

Page 18

04 웹의 탄생

2024-1  
인터넷보안실무

■ 초창기 웹

- 단순한 텍스트와 링크 위주
- 하이퍼텍스트 : 글자에 링크를 걸어놓고 클릭하면 다른 화면이 나타나는 것
- 하이퍼링크 : 한 페이지에서 다른 페이지의 문서로 쉽게 이동
- 웹 서핑, 웹 브라우징 : 하이퍼링크를 따라 이동하는 것

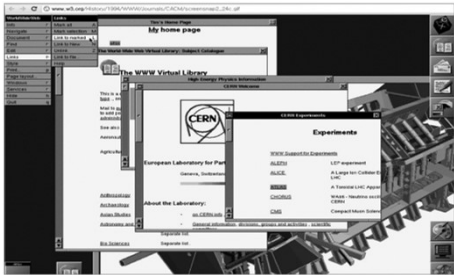


그림 1-14 팀 버너스 리가 만든 초창기 브라우저 화면

04 웹의 탄생

2024-1  
인터넷보안실무

■ 초기의 웹 브라우저

- 고급 기술이나 화려한 그래픽 없이 단순히 하이퍼링크로 여러 페이지를 묶어 놓은 정도

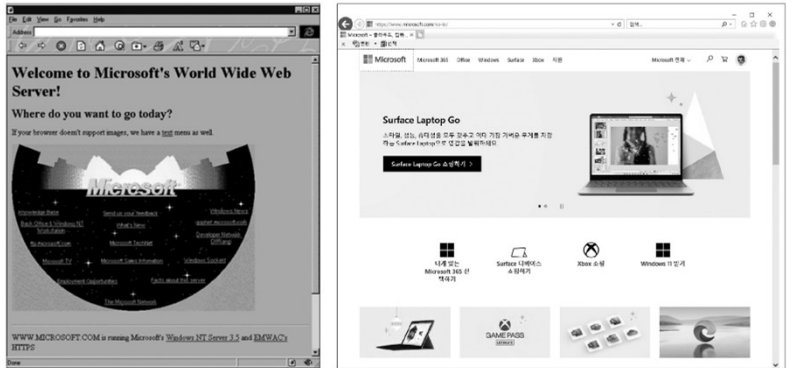


그림 1-15 초기의 마이크로소프트 웹 사이트와 최근의 마이크로소프트 웹 사이트

# 05

## HTTP의 기본 개념

### 05 HTTP의 기본 개념

2024-1  
인터넷보안실무

■ HTTP(hypertext transfer protocol)

- 인터넷에서 가장 많이 사용하는 프로토콜(팀 버너스 리가 웹을 만들면서 개발)
- 문서 간의 상호 연결을 통해 다양한 텍스트, 그래픽, 애니메이션을 화면에 보여 주고 사운드를 재생



그림 1-19 RFC-INDEX에서 HTTP 1.0 번호

05 HTTP의 기본 개념

2024-1  
인터넷보안실무

■ HTTP(HyperText Transfer Protocol)

- 0.9 버전의 HTTP는 서버에서 단순히 읽기 기능만 지원

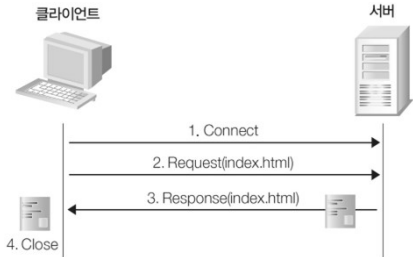


그림 1-20 HTTP 0.9를 이용하여 서버에 연결하기

- 현재 웹에서 주로 사용하는 HTTP는 1.0과 1.1 버전
  - HTTP 1.0 : 1996년 5월에 완성되었으며, 메소드는 GET, HEAD, POST 방식만 지원
  - HTTP 1.1 : 2001년에 공식 발표되어 메소드는 OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT 방식 지원

05 HTTP의 기본 개념 [1] Request

2024-1  
인터넷보안실무

■ Request

```
GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Encoding: gzip, deflate
Cookie: HSID=AaxlkKoV2snIEi6UQ; SSID=AAYVu_evC0Tiu3aVc;
APISID=JEWp0eojRTLfYKJ/ACgEWh_0mL8Li_-fl;
SAPISID=kabRBO-uT0ebDcfc/A2byDX--FwN649tHw
Host: www.google.comConnection: Keep-Alive
Accept-Language: ko-KR
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
```

- 첫 번째 줄: GET / HTTP /1.1
- HTTP 전송 방법: 웹 서버로부터 자료를 가져오는 기능을 하는 GET을 많이 사용(GET 메소드는 별도의 메시지 보디를 필요로 하지 않음)
- 요청된 URL: 웹 서버에 있는 자료를 요청할 때 사용되는 경로
- HTTP 버전: 인터넷에서 가장 일반적으로 사용되는 HTTP 버전은 1.0과 1.1(대부분의 브라우저는 초깃값으로 1.1 사용)

## 05 HTTP의 기본 개념 [1] Request

2024-1  
인터넷보안실무

## ■ Request

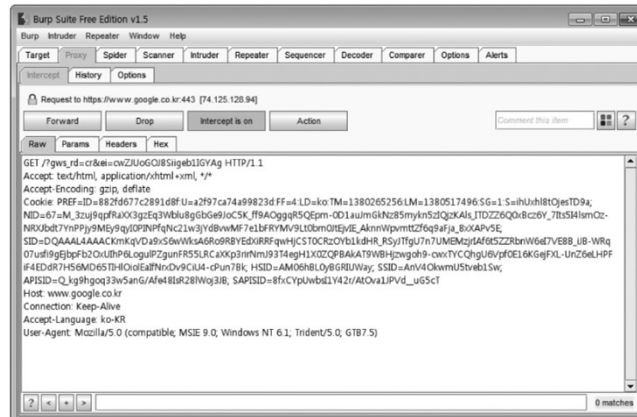


그림 1-21 GET 메서드 패킷 화면의 예

Page 25

## 05 HTTP의 기본 개념 [1] Request

2024-1  
인터넷보안실무

## ■ 웹 해킹과 관련된 요소

- 서버가 클라이언트에 전송한 인자값에 추가 정보를 보낼 때 사용

```

Cookie: HSID=AaxlkKoV2snIEi6UQ; SSID=AAYVu_evC0Tiu3aVc;
APISID= JEWp0eojRTLftYKJ/ACgEWh_0mL8Li_-fl;
SAPISID=kabRBO-uT0ebDcfc/A2byDX--FwN649tHw
  
```

- URL 주소에 나타난 호스트명을 자세하게 나타내기 위해 사용

```
Host: www.google.com
```

- 브라우저나 기타 클라이언트의 소프트웨어 정보를 보여줌

```
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
```

- GET 방식은 요청 데이터에 대한 인수를 URL을 통해 웹 브라우저로 전송(링크 주소만 알아도 연결된 페이지의 내용 확인 가능)

```
http://www.hanb.co.kr/edu/view_detail.html?hi_id=363
```

Page 26

## 05 HTTP의 기본 개념 [1] Request

2024-1  
인터넷보안실무

## ■ POST

- HTTP의 보디 영역에 소켓을 이용하여 데이터를 전송
- URL을 통해 인수값을 전송하지 않기 때문에 다른 사람이 링크를 통해 해당 페이지를 볼 수 없음
- 보내려는 인자값이 URL을 통해 노출되지 않아 보안 측면에서 GET 방식보다 안전

## ■ 일반적인 게시판에서의 GET 방식과 POST 방식의 사용

- 목록이나 글을 보는 화면에는 접근 자유도를 부여하기 위해 GET 방식 사용
- 글을 저장 · 수정 · 삭제하는 작업을 할 때는 보안을 위해 POST 방식 사용

Page 27

## 05 HTTP의 기본 개념 [1] Request

2024-1  
인터넷보안실무

## ■ POST 메소드의 예

```
POST / HTTP/1.0
Accept: */*X-Cl: 126323033
X-AT: OVERNET
X-GO: 1;KR;842;9530
X-DM: www.google.co.kr
X-SP: 762
Host: dr1.webhancer.com
Content-Length: 0
Pragma: no-cache
Connection: Close
```

Page 28

05 HTTP의 기본 개념 [1] Request

2024-1  
인터넷보안실무

■ Request 패킷의 메소드

- HEAD: 서버 쪽 데이터를 검색하고 요청하는 데 사용
- OPTIONS: 자원에 대한 요구- 응답 관계에서 관련된 선택 사항에 대한 정보를 요청할 때 사용
- PUT: 메시지에 포함되어 있는 데이터를 지정한 URI 장소에 지정된 이름으로 저장
- DELETE: URI에 지정되어 있는 자원을 서버에서 지울 수 있게 함
- TRACE: 요구 메시지의 최종 수신처까지 루프백 검사용으로 사용

05 HTTP의 기본 개념 [2] Response

2024-1  
인터넷보안실무

■ Response

- 클라이언트가 보낸 Request의 응답 패킷으로 형식이 간단함
- Response 패킷에 담긴 주요 내용은 서버에서 쓰이는 프로토콜 버전, HTTP 상태 코드(200 OK) 등이며, 전달할 데이터의 형식, 데이터 길이 등과 같은 추가 정보가 포함되어 있음



그림 1-22 Response 응답 화면

05 HTTP의 기본 개념 [2] Response

2024-1  
인터넷보안실무

■ HTTP 상태 코드

표 1-1 일반적인 상태 코드

상태 코드	의미	설명
100번대	정보 전송	임시 응답을 나타내는 것은 Status-Line과 선택적인 헤더로 이루어져 있고 빈 줄로 끝을 맺는다. HTTP 1.0까지는 계열에 대한 어떤 정의도 이루어지지 않았기 때문에 시험용 외에는 서버 쪽의 추가 응답이 없다.
200번대	성공	클라이언트의 요청이 성공적으로 수신되어 처리되었음을 의미한다.
300번대	리다이렉션	클라이언트의 요구 사항을 처리하려면 다른 곳에 있는 자원이 필요하다는 것을 의미한다.
400번대	클라이언트 측 에러	클라이언트가 서버에 보내는 요구 메시지를 완전히 처리하지 못한 경우처럼 클라이언트 측에서 오류가 발생한 것을 의미한다.
500번대	서버 측 에러	서버 자체에서 생긴 오류 상황이나 클라이언트의 요구 사항을 제대로 처리할 수 없을 때 발생한다.

Page 31

05 HTTP의 기본 개념 [2] Response

2024-1  
인터넷보안실무

■ HTTP 세부적인 상태 코드

표 1-2 상세한 상태 코드

상태 코드	의미	상태 코드	의미
100	Continue	404	Not Found
101	Switching Protocols	405	Method Not Allowed
200	OK	406	Not Acceptable
201	Created	407	Proxy Authentication Required
202	Accepted	408	Request Time-Out
203	Not-Authorized Information	409	Conflict
204	No Content	410	Gone
205	Reset Content	411	Length Required
206	Partial Content	412	Precondition Failed
300	Multiple Choices	413	Request Entity Too Large
301	Moved Permanently	414	Request URI Too Large
302	Moved Temporarily	415	Unsupported Media Type
303	See Other	500	Internal Server Error
304	Not Modified	501	Not Implemented
305	Use Proxy	502	Bad Gateway
400	Bad Request	503	Service Unavailable
401	Unauthorized	504	Gateway Time-Out
402	Payment Required	505	HTTP Version not Supported
403	Forbidden		

Page 32



05 HTTP의 기본 개념 [2] Response

2024-1  
인터넷보안실무

■ HTTP 세부적인 상태 코드 설명

- 200 OK : 클라이언트의 요청이 성공했다는 것을 나타냄
- 201 Created : 클라이언트의 PUT 요청이 성공적이라는 것을 나타냄
- 301 Moved Permanently : 브라우저의 요청을 다른 URL로 항상 전달
- 302 Moved Temporarily : 브라우저의 요청을 임시 URL로 바꾸고 Location 헤더에 임시로 변경한 URL의 정보를 적음(클라이언트가 다음에 같은 요청을 하면 기존 URL로 돌아감)
- 304 Not Modified : 브라우저가 서버에 요청한 자료에 대해 서버는 클라이언트 내에 복사된 캐시를 사용하면 된다는 것을 의미
- 400 Bad Request : 클라이언트가 서버에 잘못된 요청을 했다는 것을 나타냄
- 401 Unauthorized : 서버가 클라이언트의 요청에 대해 HTTP 인증 확인을 요구
- 403 Forbidden : 클라이언트의 요청에 대해 접근을 차단
- 404 Not Found : 클라이언트가 서버에 요청한 자료가 존재하지 않음
- 405 Method Not Allowed : 클라이언트가 요청에 이용한 메소드는 해당 URL에 지원이 불가능함
- 413 Request Entity Too Large : 클라이언트가 요청한 보디가 서버에서 처리하기에는 너무 큼
- 500 Internal Server Error : 서버가 클라이언트의 요청을 실행할 수 없을 때 500 상태 코드가 발생 (SQL 인젝션 취약점이 존재하는지 확인할 때 유용)

Page 33

05 HTTP의 기본 개념 [2] Response

2024-1  
인터넷보안실무

■ HTTP 1.0

- 문서에 몇 개의 그림이 있는 상관없이 텍스트가 저장된 HTML 문서를 먼저 전송받은 후 연결을 끊고 다시 연결하여 그림을 전송받음

```
sequenceDiagram
    participant Client as 클라이언트
    participant Server as 서버
    Note over Client, Server: 1. Connect
    Client->>Server: 2. Request(index.html)
    Server-->>Client: 3. Response(index.html)
    Note over Client, Server: 4. Close
    Note over Client, Server: -----
    Client->>Server: 5. Connect
    Client->>Server: 6. Request(index.jpg)
    Server-->>Client: 7. Response(index.jpg)
    Note over Client, Server: 8. Close
```

그림 1-23 HTTP 1.0을 이용하여 index.html 읽기

Page 34

05 HTTP의 기본 개념 [2] Response

2024-1  
인터넷보안실무

■ HTTP 1.1

- 연결 요청이 계속 들어오면 HTML 문서를 받은 후 바로 그림 파일을 요청

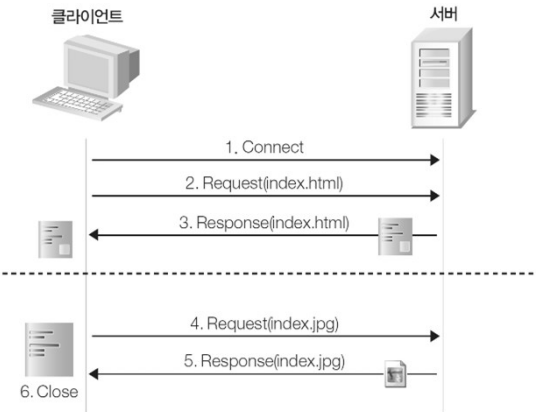


그림 1-24 HTTP 1.1을 이용하여 index.html 읽기

06

웹 애플리케이션 기술

06 웹 애플리케이션 기술

2024-1  
인터넷보안실무

- 웹 애플리케이션에는 클라이언트와 서버 사이에 메시지를 전달하기 위한 핵심 통신 프로토콜뿐만 아니라 웹 애플리케이션의 각 기능을 활용하기 위한 다양한 기술이 사용됨
- 웹 애플리케이션을 공격하거나 보호하려면 웹 애플리케이션에 어떤 기술이 사용되는지, 어떻게 동작하는지, 취약점은 무엇인지 알아야 함

Page 37

06 웹 애플리케이션 기술 [1] 서버 측 기능

2024-1  
인터넷보안실무

■ 서버 측 기능

- 초기의 웹 서버는 단순한 정적 페이지 제공
- 현재는 입력한 값에 따라 다양한 결과를 화면에 보여주는 동적 기능 제공

■ 웹 애플리케이션이 이용하는 서버 측 기능

- PHP, Node.js, Python, ASP, ASP.NET, JSP, Java, VBScript와 같은 서버 측 스크립트 언어
- Nginx, Cloudflare Server, Apache, IIS, Sun Java System, Node.js와 같은 웹 서버
- Microsoft SQL 서버, 오라클 데이터베이스, MySQL, PostgreSQL, IBM DB2와 같은 데이터베이스

Page 38

06 웹 애플리케이션 기술 [1] 서버 측 기능

2024-1  
인터넷보안실무

■ 서버 측 스크립트 언어

- 클라이언트가 요청한 데이터를 서버 측에서 처리하여 원하는 결과를 돌려주기 위해 사용하는 언어
- 윈도우 계열 기반은 주로 ASP, 웹 애플리케이션 플랫폼에는 주로 JSP 사용
- 최근에는 Node.js처럼 운영체제와 상관없이 구현 가능한 스크립트 언어도 사용
- 웹 애플리케이션 소스코드를 분석해서 취약점을 찾으려면 기본적인 서버 측 스크립트 언어에 대해 이해하고 있어야 함

Page 39

06 웹 애플리케이션 기술 [1] 서버 측 기능

2024-1  
인터넷보안실무

■ 웹 서버

- 일반적으로 많이 사용하는 웹 서버는 아파치와 IIS
- 현재는 Nginx와 같은 새로운 웹 서버도 많이 사용됨
- 2022년 4월을 기준으로 Nginx 점유율이 가장 높고 전통적으로 많이 사용되  
는 아파치 웹 서버가 그 다음
- 마이크로소프트의 IIS 웹 서버는 2018년에 점유율이 50%까지 올랐으나 지  
금은 상당히 하락

표 1-3 개발사별 웹 서버 점유율

개발사	웹 서버 수(2022년 4월 기준)	점유율(%)
Nginx	361,438,143	31.13
아파치	268,005,916	23.08
OpenResty	92,950,864	8.01
Cloudflare	63,701,232	5.49

Page 40

06 웹 애플리케이션 기술 [1] 서버 측 기능

2024-1  
인터넷보안실무

■ 데이터베이스

- 데이터베이스 관리 시스템(DBMS) : 데이터베이스를 관리하는 소프트웨어
- DBMS에는 Microsoft SQL 서버, 오라클 데이터베이스, MySQL, PostgreSQL, IBM DB2 등이 있음
- DBMS를 사용하면 데이터베이스를 만들고 데이터를 입력 · 변경 · 검색할 수 있음

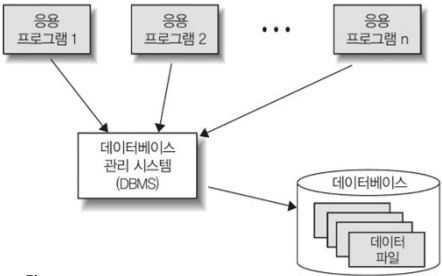


그림 1-34 DBMS의 역할

06 웹 애플리케이션 기술 [2] 클라이언트 측 기능

2024-1  
인터넷보안실무

■ HTML

- 1980년 유럽입자물리연구소(CERN)의 팀 버너스 리가 HTML의 원형인 인콰이어를 제안
- 1991년 말 팀 버너스 리가 인터넷에서 문서를 'HTML 태그'라고 부르면서 시작
- HTML은 2000년부터 국제표준(ISO/IEC 15445:2000)이 됨
- 2010년 8월 HTML5 Working Draft가 공개
- 2014년 10월 28일 HTML5가 표준으로 확정
- 2017년에 HTML 5.2가 권고안으로 공개
- HTML5의 대표적인 특징은 멀티미디어 요소로, 오디오와 비디오 요소를 이용하면 플러그인 없이 멀티미디어를 재생할 수 있음

■ 자바스크립트

- 객체 기반의 스크립트 프로그래밍 언어
- 넷스케이프커뮤니케이션의 운영자인 브렌던 아이크가 모카라는 이름으로 처음 개발
- 라이브스크립트(LiveScript)라는 이름을 거쳐 자바스크립트가 됨
- 성능 문제로 인해 서버 측에서 처리하지 않는 부분을 클라이언트 측에서 처리할 수 있도록 할 때 주로 사용
- 자바스크립트로 작성된 입력값 검증 부분은 주요 공격 대상이 됨



감사합니다.

인터넷  
해킹과 보안 4판