

웹 어플리케이션 취약성을 제거하라

안전한 웹을 위한 코딩 한줄의 정석

Hiroshi Tokumaru 지음
박건태, 신대호 옮김



안전한 웹을 위한 코딩 한줄의 정석

지은이 Hiroshi Tokumaru

옮긴이 박건태, 신대호

1판 1쇄 발행일 2012년 11월 2일

펴낸이 장미경

펴낸곳 로드북

편집 임성준

디자인 이호용(표지), 박진희(본문)

주소 서울시 관악구 신림동 1451-15 101호

출판 등록 제 2011-21호(2011년 3월 22일)

전화 02)874-7883

팩스 02)843-6901

정가 28,000원

ISBN 978-89-97924-01-1 93560

TAIKEITEKINI MANABU ANZENNA WEB APPLICATION NO TSUKURIKATA

Copyright © 2011 Hiroshi Tokumaru

All rights reserved.

Original Japanese edition published by SOFTBANK Creative Corp.

Korean translation rights © 2012 by RoadBook

Korean translation rights arranged with SOFTBANK Creative Corp. Tokyo

through EntersKorea Co., Ltd. Seoul, Korea

이 책의 한국어판 저작권은 (주)엔터스코리아를 통한 일본의 SOFTBANK Creative Corp.

와의 독점 계약으로 로드북이 소유합니다.

신 저작권법에 의하여 한국 내에서 보호를 받는 저작물이므로 무단전재와 무단복제를 금합니다.

이메일 chief@roadbook.co.kr

블로그 www.roadbook.co.kr

Q&A roadbook.zerois.net/qna

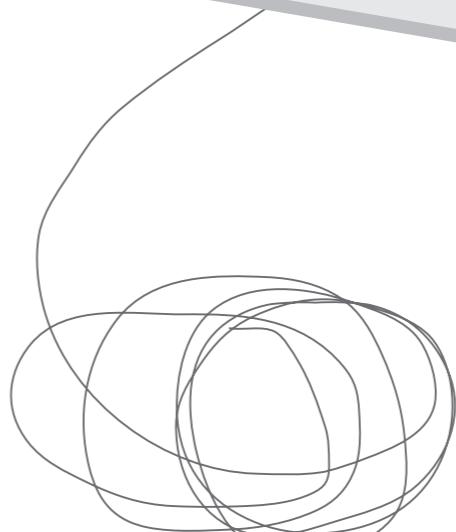
예제소스 다운로드

예제소스는 아래 URL에서 다운로드 해주세요.

<http://www.roadbook.co.kr/89>

제공되는 자료는 다음과 같습니다.

- 실습 환경을 갖춘 가상 머신 이미지 파일(자세한 내용은 2장 참고)
- 본문 예제 소스



지은이 머리말

최근 웹 어플리케이션 취약성을 노린 공격이 빈번하게 발생하고 있으며 그에 따른 피해가 속출하고 있습니다. 공격에 대처하기 위해서는 취약성을 없애면 되겠지만, 그러기 위해서는 웹 어플리케이션 개발 엔지니어가 보안에 대한 올바른 지식을 가지고 있을 필요가 있습니다.

이미 인터넷 상에는 보안에 관한 정보가 넘치고 있지만 그 중 대부분은 표면적인 내용이라 개발 엔지니어가 궁금해 하는 필요한 정보를 얻기에는 부족한 면이 있습니다. 구체적으로는 다음의 정보를 예로 들 수 있습니다.

왜 취약성이 발생하는 것인가?

취약성이 있으면 어떠한 영향이 있는가?

취약성을 없애기 위해서는 어떻게 프로그래밍을 해야 하는가?

왜 그런 방법으로 취약성이 없어지는가?

이 책은 이런 의문에 대한 해답을 제시할 목적으로 기획되었습니다. 따라서 취약성이 생기는 원리부터 구체적인 대처 방법과 근거에 대한 내용까지 가능한 자세히 설명하고 있습니다. 대상 독자는 프로그래머, 설계자, 프로젝트 관리자, 품질 관리 담당자 등 웹 어플리케이션 개발과 관련된 모든 사람을 대상으로 합니다. 또한 웹 어플리케이션을 발주하는 입장에 있는 분들에게도 가능한 유용한 정보를 설명하기 위해 최선을 다했습니다.

주로 개발자를 위해 썼지만, 공격 방법에 대해서도 구체적으로 설명하고 있습니다. 그 이유는 취약성에 따른 영향을 절실히 알아주셨으면 하기 때문입니다. 하지만 공격을 웹사이트 관리자의 허가 없이 테스트하면 관련 법률에 따라 처벌을 받을 수 있습니다. 공격 수법을 실제 서비스 중인 사이트에 대해 허가 없이 테스트하지 않도록 주의하시길 바랍니다. 이 책에서는 독자가 안심하고 공격 방법 등을 테스트하기 위해 VMware Player 가상 머신에서 취약성 샘플을 테스트할 수 있도록 준비했습니다. 실습에 필요한 소프트웨어는 무상으로 제공하고 있는 틀입니다. 스스로 동작을 시켜보는 등 직접 테스트하여 취약성에 대한 이해를 완전히 본인 것으로 만들 수 있기를 기대합니다.

또한 이 책의 프로그램 샘플은 주로 PHP를 활용했지만 설명 내용은 PHP에만 국한되지 않도록 주의를 기울였습니다.

옮긴이 머리말

요사이 해킹에 의한 대규모 고객 정보(개인정보) 유출 사고가 끊임 없이 뉴스의 한 부분을 장식하고 있습니다. 굴지의 대기업 및 공공 기관까지 공격을 받아 관련 피해가 확대되어 감에 따라 사태의 심각성이 더욱 커지고 있습니다.

이는 기업이나 기관 및 단체뿐만 아니라 고객에게 2차 3차 피해로 확대되어 피해자의 삶이 파괴될 수 있는 심각한 문제를 일으킬 수 있다는 점에서 경각심을 가져야 할 필요가 있습니다. 이는 온라인 전쟁이자 테러입니다. 그 피해는 오프라인까지 미치고 심지어 피해가 막대하다는 것을 우리는 익히 알고 있습니다.

우리는 선한 방어자로서 최대한 강력한 시스템을 구축해야 합니다. 테러리스트(크래커)들은 방어가 소홀한 곳을 끊임 없이 공격해 올 것입니다. 우리는 담대하게 적들의 공격을 무력화 시켜야 합니다. “죄송합니다 2차 피해가 없도록 최선을 다하겠습니다.”라는 팝업 창 하나만 띠운 채, 원인에 대해서는 신종 해킹이라 대처가 어려웠다는 변명은 더 이상은 안 됩니다.

사회적으로 엔지니어들 사이에서도 관심이 높아진 이 시점에서 본서와 같은 책이 나온 것을 참으로 감사하고 다행스럽게 생각합니다.

이 책은 보안에 관해 단순하게 방어적인 프로그래밍에 대한 설명만을 소개하는 책이 아닙니다. 공격 방법에 대해서도 자세히 설명하고 해결책을 제시하고 있습니다. 또한 엔지니어의 지식에 대한 부족함에 경각심을 주고 저자의 오랜 관심과 연구 결과를 반영시킨 매우 깊이 있는 책이라고 확신합니다. 부디 이 책을 통해 기술적으로 한 단계 진보할 수 있는 기회가 되길 바랍니다.

마지막으로 있을 만한 질문에 대해 간단히 말씀 드립니다.

- URL을 원문 그대로 example.jp로 하고 있습니다. vmware를 사용하여 내부 네트워크를 구축하여 테스트하므로 변경해야 할 특별한 이유가 없었습니다.
- 한국에서는 주로 ‘취약점’으로 표현되지만 원문 그대로 ‘취약성’으로 번역했습니다. 특별히 혼란을 줄 수 있는 용어가 아니기 때문입니다.
- OS 이미지는 직접 (마음껏) 테스트하고, 필요하면 언제든 지우고 새로 이미지를 받아 사용하면 됩니다.

대표 역자 박건태

지은이 토쿠마루 히로시

지은이 머리말 3
옮긴이 머리말 4

CHAPTER 1 웹 어플리케이션 취약성이란?

- 1.1 취약성이란, '악용할 수 있는 버그' 14
- 1.2 취약성이 있으면 안 되는 이유 16
- 1.3 취약성 발생의 원인 20
- 1.4 보안 버그와 보안 기능 21
- 1.5 책의 구성 22

CHAPTER 2 실습 환경 구축

- 2.1 실습 환경의 개요 24
- 2.2 VMware Player 설치 26
- 2.3 가상 머신 설치 및 동작 확인 29
- 2.4 Fiddler 인스톨 38

CHAPTER 3 웹 보안의 기초: HTTP, 세션 관리, SOP

- 3.1 HTTP와 세션 관리 46
 - COLUMN 쿠키 몬스터 문제
- 3.2 수동적 공격과 Same Origin Policy 78
 - COLUMN 제삼자의 JavaScript를 허가하는 경우
 - COLUMN X-FRAME-OPTIONS

CHAPTER 4 안전성을 위협하는 웹 어플리케이션 버그

- 4.1 웹 어플리케이션 기능과 취약성의 관계 94
- 4.2 입력 처리와 보안 98
 - COLUMN 문자 인코딩의 자동 변환과 보안
 - COLUMN 입력값과 프레임워크
- 4.3 표시 처리에 관한 취약성 문제 116
 - 4.3.1 크로스 사이트 스크립팅(기본편) 116
 - 4.3.2 크로스 사이트 스크립팅(발전편) 137
 - 4.3.3 에러 메시지에서 정보 유출 151
- 4.4 SQL 호출에 따른 취약성 152
 - 4.4.1 SQL 인젝션 152
 - COLUMN DB 안의 테이블명, 컬럼명의 조사 방법
 - COLUMN MDB2를 채용한 이유
- 4.5 중요한 처리를 할 때에 생기는 취약성 177
 - 4.5.1 Cross-Site Request Forgeries, CSRF 177
 - COLUMN 내부 네트워크에 대한 CSRF 공격
 - COLUMN 토큰과 원타임 토큰

4.6 세션 관리의 취약성 196	<small>COLUMN</small> 업로드한 파일로 서버측 스크립트의 실행
4.6.1 세션 하이재킹의 원인과 영향 196	<small>COLUMN</small> 파일명에 의한 XSS 주의
4.6.2 추측 가능한 세션 ID 199	<small>COLUMN</small> 확장자를 체크할 때의 주의사항
4.6.3 URL 포함 세션 ID 204	4.12.3 파일 다운로드에 의한 크로스 사이트 스크립팅 325
4.6.4 세션 ID의 고정화 212	<small>COLUMN</small> BMP 형식에 대한 주의와 MS07-057
4.7 리다이렉트 처리 관련 취약성 226	<small>COLUMN</small> 이미지를 별도의 도메인에서 제공하는 경우
4.7.1 오픈 리다이렉터 취약성 227	4.13 인클루드에서 발생하는 문제 343
<small>COLUMN</small> 쿠션 페이지	4.13.1 파일 인클루드 공격 343
4.7.2 HTTP 헤더 인젝션 235	<small>COLUMN</small> RFI 공격의 다양성
<small>COLUMN</small> HTTP Response 분할 공격	4.14 eval에서 발생하는 문제 353
<small>COLUMN</small> HTTP 헤더와 개행	4.14.1 eval 인젝션 353
<small>COLUMN</small> PHP header 함수는 개행을 어디까지 체크하는가	4.15 공유 자원에 관한 문제 362
4.7.3 리다이렉트 처리에 관한 취약성 정리 250	4.15.1 경합 상태 취약성 362
4.8 쿠키 출력에 관한 취약성 251	
4.8.1 쿠키의 부적절한 이용 252	
<small>COLUMN</small> 패딩 오라클 공격과 MS10-070	
4.8.2 쿠키의 Secure 속성에 관한 취약성 255	
4.9 메일 송신 문제 267	CHAPTER 5 대표적인 보안 기능
4.9.1 메일 송신 문제의 개요 267	5.1 인증 370
4.9.2 메일 헤더 인젝션 취약성 269	5.1.1 로그인 기능 370
4.10 파일 접근에 발생하는 문제 281	5.1.2 무작위 공격에 대한 대책 379
4.10.1 디렉토리 트레버설 취약성 282	5.1.3 패스워드 저장 방법 383
<small>COLUMN</small> 스크립트의 스스로부터 줄줄이 정보가 누설된다	<small>COLUMN</small> 데이터베이스 암호화와 패스워드 보호
<small>COLUMN</small> basename 함수와 널 바이트	<small>COLUMN</small> 암호학적 해시 함수가 만족되는 요건
4.10.2 의도하지 않은 파일 공개 290	<small>COLUMN</small> 패스워드 누출 경로
4.11 OS 커맨드를 호출할 때 발생하는 취약성 294	5.1.4 자동 로그인 395
4.11.1 OS 커맨드 인젝션 294	5.1.5 로그인 폼 404
4.12 파일 업로드에서 발생하는 문제 312	<small>COLUMN</small> 패스워드는 꼭 마스크 표시 해야 하는가
4.12.1 파일 업로드 문제의 개요 312	5.1.6 에러 메시지의 요건 406
<small>COLUMN</small> 메모리 사용량이나 CPU 사용 시간 등 다른 리소스에도 주의	5.1.7 로그아웃 기능 407
	5.1.8 인증 기능 정리 409

5.2 계정 관리 413
 5.2.1 유저 등록 413
 5.2.2 패스워드 변경 419
 5.2.3 메일주소의 변경 420
 5.2.4 패스워드 리마인더 422
 5.2.5 계정의 정지 426
 5.2.6 계정의 삭제 427
 5.2.7 계정 관리 정리 427

5.3 인가 428
 5.3.1 인가란? 428
 5.3.2 인가가 불충분한 전형적인 예 429
 COLUMN 비밀정보를 포함한 URL에 의한 인가 처리
 5.3.3 인가 제어의 요건 정의 433
 COLUMN 룰이란?
 5.3.4 인가 제어의 올바른 구현 434
 5.3.5 정리 435

5.4 로그 출력 436
 5.4.1 로그 출력의 목적 436
 5.4.2 로그의 종류 437
 5.4.3 로그 출력의 요건 438
 5.4.4 로그 출력의 구현 441
 5.4.5 정리 442

6.2 속임수 대책 455
 6.2.1 네트워크적인 속임수 455
 COLUMN VISA 도메인 문제
 6.2.2 피싱 457
 6.2.3 웹사이트 속임수 대책 457
 COLUMN 무료 서버 증명서

6.3 감청, 변조 대책 462
 6.3.1 감청, 변조의 경로 462
 6.3.2 중간자 공격 463
 COLUMN 루트 증명서를 도입하지 않고 시키지도 않는 일
 COLUMN SSL의 확인 아이콘
 6.3.3 대책 467

6.4 맬웨어 대책 470
 6.4.1 웹사이트의 맬웨어 대책이란 470
 6.4.2 맬웨어 감염 경로 471
 6.4.3 웹 서버 맬웨어 대책의 개요 472
 6.4.4 웹 서버에 맬웨어가 감염되지 않게 하는 대책 472
 COLUMN 웹사이트의 바이러스 대책과 검블러의 관계

6.5 정리 476

CHAPTER 6 웹사이트의 안전성을 높이기 위해

6.1 서버에 대한 공격 경로와 대책 445
 6.1.1 기반 소프트웨어의 취약성을 이용한 공격 445
 6.1.2 부적절한 로그인 445
 6.1.3 대책 446

7.1 개발 관리에서 보안 정책의 전체 이미지 478
7.2 개발 체제 480
7.3 개발 프로세스 483
 7.3.1 기획 단계의 유의점 483
 7.3.2 발주시 유의점 483

COLUMN 취약성에 대한 책임

- 7.3.3 요건 정의시의 주의점 484
 - 7.3.4 기본 설계의 진행 방법 486
 - 7.3.5 상세 설계, 프로그래밍시의 유의점 486
 - 7.3.6 보안 테스트의 중용성과 방법 486
 - 7.3.8 수주자 측 테스트 487
 - 7.3.9 발주자 측 테스트(검수) 488
 - 7.3.10 운용 단계의 유의점 488
- 7.4 정리 490

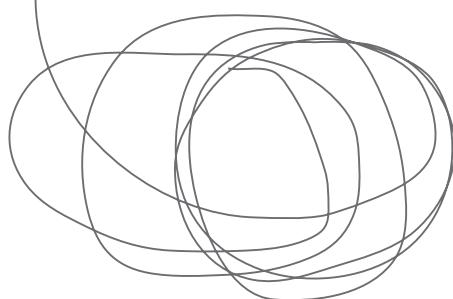
Chapter 1

웹 어플리케이션 취약성이란?

이번 장에서는 책 전체의 주제에 해당하는 취약성에 대해 설명합니다.

취약성이란 무엇이며, 왜 문제가 되는지 그리고 취약성이 생기는 원인은 무엇인지 등에 대해 알아보겠습니다.

이 장의 후반부에서는 이 책의 구성 및 학습 방법을 정리해두었습니다.



취약성이란, ‘악용할 수 있는 버그’

모든 어플리케이션에는 버그가 있기 마련이고, 개발자는 항상 버그와 함께 있다고 말할 수 있을 정도로 개발자인 우리는 새로운 기능을 개발하면서도, 기존 시스템의 버그를 수정하고 패치하는 작업을 하고 있습니다. 어플리케이션 버그는 예를 들어 잘못된 결과를 표시하거나, 시스템 행에 걸리거나, 화면이 깨져서 표시되거나, 속도가 너무 느리거나 하는 결과를 초래합니다. 또한 버그 중에는 악용할 수 있는 것도 존재합니다. 이런 악용 가능한 버그를 취약성 vulnerability 또는 보안 security 버그라고 합니다.

악용이란 어떤 것이 가능할까요? 악용에 대한 예를 보도록 하겠습니다.

- 타인의 개인 정보를 자유롭게 열람할 수 있다.
- 웹사이트의 내용을 바꿀 수 있다.
- 사이트에 접속한 이용자의 PC에 바이러스를 감염시킬 수 있다.
- 실제 이용자로 위장하여 개인 정보의 열람, 글을 작성하여 등록, 온라인 쇼핑, 은행 송금 등을 할 수 있다.
- 웹사이트를 이용 불가 상태로 만들 수 있다.
- 온라인 게임 등에서 무적의 캐릭터가 될 수 있는 아이템을 마음껏 얻을 수 있다.
- 이용자는 자신의 개인 정보를 확인하려고 했더니, 다른 사람의 개인 정보가 보인다.¹

보통 버그라는 것이 개발자에게 (안타깝게도) 늘 가까운 곳에 존재하듯이, 웹 어플리케이션 개발자에게 취약성 역시 늘 가까운 곳에서 존재하고 있습니다. 취약성에 관해 의식하지 않은 채, 웹 어플리케이션을 개발하면 위에서 나열한 것과 같이 악용 가능한 웹 사이트가 만들어지고 마는 것입니다. 이 책은 취약성이 없는 안전한 웹 어플리케이션의 개발 방법에 대한 원리부터 대책까지 다루고 있습니다.

¹ 다른 이용자 정보가 보이는 버그는 의도적으로 악용한 것은 아니지만, 우연한 사고로 인해 보안 문제가 생긴 것을 말합니다. 실제로 몇 년 전 메일 서비스를 제공하는 한 포털 사이트에 개인 계정으로 로그인하였더니, 다른 사람 계정의 메일을 볼 수 있었던 경우가 있었습니다.

1.2

취약성이 있으면 안 되는 이유

취약성이 있으면 안 되는 이유를 몇 가지 측면에서 검토해 보도록 하겠습니다.

경제적 손실

취약성이 있어서는 안 될 첫 번째 이유는 웹사이트의 경제적 손실입니다. 전형적으로 발생하는 손실로는 다음과 같은 것이 있을 수 있습니다.

- 이용자의 금전적 손실에 대한 보상
- 변상 및 위자료 등의 비용
- 웹사이트를 당분간 운영할 수 없는 기회 손실
- 신용 실추에 따른 매출의 감소

이와 같은 경제적 손실은 적게는 수억 원에서 많게는 수백 억에 이르는 경우도 있습니다.

하지만 다음과 같은 의문이 생길지도 모르겠습니다. 매출 규모가 그다지 크지 않은 웹사이트라면 위에 나열한 경제적 손실은 상대적으로 작으므로 “만약 무슨 일이 생겼을 경우, 이용자의 손실에 대해서는 충분히 보상 가능한 범위이므로 굳이 사전 대책이 필요할까?”라고 생각하는 웹사이트 운영자가 있을지 모르겠습니다. 그럼 과연 경제적 손실뿐일까요? 경제적 손실 이외의 측면에 대해서도 알아보도록 하겠습니다.

법적인 요구

웹사이트 안전 대책에 대한 법률로서 개인정보에 관한 법률(정보통신망법) 개정이 2011년 12월 29일 국회 본회의를 통과하여 2012년 2월 17일 공포 후 6개월이 경과한 8월 18일부터 시행되고 있습니다. 이 법안은 개인정보를 수집하고 저장하는 사업자는 개인정보를 취급하는 사업자로서 안전 조치에 대한 의무를 명시하고 있습니다.

최근 해킹 등으로 인한 대규모 개인정보 누출 사고가 연이어 발생하면서 기업의 개인정보 보호 체계를 전면적으로 강화할 필요가 있다는 사회적 공감대가 형성되었습니다. 인터넷 상의 개인정보 보호 강화 방안은 개인정보의 과도한 수집 제한 및 기업의 개인정보 관리 강화, 이용자의 자기 정보 통제 강화를 주요 골자로 하는 20개 세부 실천 과제를 포함하고 있으며, 이 가운데 주민번호 수집 이용 제한, 개인 정보 유효 기간제, 개인정보 이용 내역 통지제, 개인정보 누출 통지 신고제 등의 신규 제도가 정보통신망법 개정을 통해 신설되었습니다.

주요 사항을 살펴 보면, 개인정보 누출 통지 및 신고에 대한 제도, 개인정보 유효기간 제도, 개인정보 이용 내역 통지 제도, 개인정보 처리 시스템 망 분리로 크게 나눌 수 있으며, 개인정보를 위한 도입 취지 및 근거 법령 그리고 사업자 조치 사항 등을 자세히 안내하고 있습니다. 행정 처벌은 각각에 대해 약간의 차이가 있으나, 대체적으로 2년 이하의 징역 또는 3천만 원 이하의 과태료를 부과하고 있다는 것을 확인할 수 있었습니다. 즉, 웹사이트에서 개인정보를 다루는 사업자는 개인정보 보호법 및 관련 가이드라인으로부터 웹 어플리케이션의 취약성에 대한 대책으로 안전 조치 의무가 요구되고 있는 것입니다.

자세한 내용은 <http://www.doitnow2012.kr>의 <개정 정보통신망법 개인정보보호 신규 제도 안내서>를 참고하기 바랍니다.

이용자가 회복하기 힘든 피해를 입는 경우가 많다

취약성이 원인이 되는 사건에는 이용자의 피해가 회복하기 힘든 것이 많다는 사실도 고려해야 합니다. 일단 유출된 개인 정보를 막는 것은 불가능합니다. 이용자의 명예를 훼손시킨 경우, 원래의 상태로 돌이키는 것은 불가능하다는 것은 너무나도 자명한 일입니다.

또한 신용카드 번호가 유출된 경우, 이용자의 금전적인 손실은 보상해줄 수 있어도 이용자가 입은 상처와 불안 그리고 그에 따른 고통은 보상해 줄 수는 없을 것입니다.

즉, 사건이 발생한 이후에 돈으로 해결하는 것이 사실상 불가능하다고 할 수 있습니다.

웹사이트 이용자들에게 거짓말을 하는 것이다

많은 웹사이트가 자신의 사이트는 안전하다고 주장하고 있습니다. “이 사이트는 보안을 전혀 고려하고 있지 않으므로 이용자가 스스로 책임지고 이용해주세요.”라고 하는 웹사이트는 없을 것입니다.

만약 사이트의 안전성을 주장한다면 취약성을 없애야 할 필요가 있습니다. 취약성이 존재한다면 웹사이트 안전성에 큰 영향을 미칠 것이기 때문입니다.

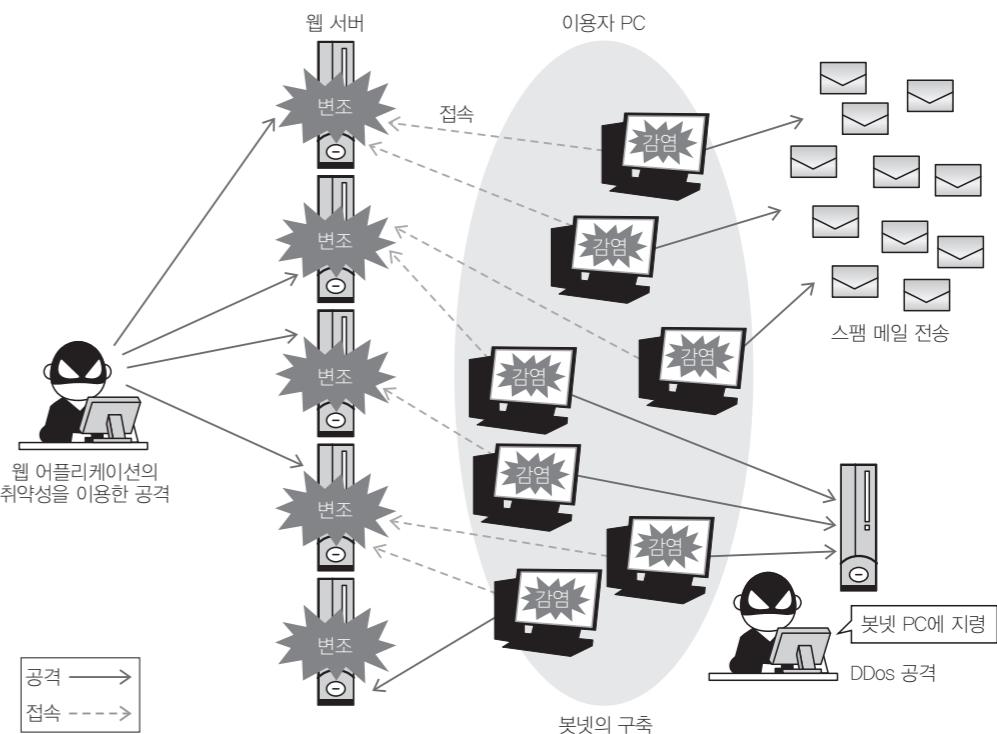
봇넷^{Botnet} 구축에 가담

인터넷 안전성을 위협하는 요인 중 하나가 봇넷^{Botnet}입니다. 봇넷이란 Malware의 일종으로, PC에 바이러스가 감염되어 외부로부터의 지령을 받아 스팸 메일 전송 또는 DDoS 공격(분산형 서비스 방해 공격) 등에 가담하는 좀비 PC들로 구성된 네트워크를 말합니다. 2010년 큰 문제가 되었던 Gumbelar 역시 봇넷 구축이 목적 중 하나였다고 합니다.

웹 어플리케이션의 취약성이 봇넷 구축에 악용되고 있습니다. [그림 1-1]은 웹 어플리케이션 취약성에 의해 봇넷이 구축되는 상황을 이미지로 정리한 것입니다.

공격자는 취약성이 있는 웹사이트의 내용을 변경하여 사이트에 접속한 이용자의 PC가 봇에 감염되도록 합니다. 그 사이트에 접속한 이용자의 PC가 봇^{Bot}에 감염되어 공격자의 지령을 받아 자신도 모르는 사이에 공격에 가담되는 상태가 됩니다. 봇넷의 일원이 된 좀비 PC는 스팸 메일 전송 또는 DDoS 공격에 이용됩니다.

웹 어플리케이션의 취약성을 이용한 공격



[그림 1-1] 웹 어플리케이션 취약성을 악용한 봇넷 구축

봇넷은 네트워크 범죄자의 큰 수입원이 되고 있다고 합니다. 즉 인터넷에 취약한 웹사이트를 개발하여 일반인에게 공개하는 것은 반사회적인 세력에 가담하는 것과 다를 바 없다고 할 수 있습니다.

1.3

취약성 발생의 원인

다음으로 취약성이 생기는 원인에 대해서 설명합니다.

우선 취약성의 발생 원인은 다음의 2종류로 나누어 생각할 수 있습니다.

(A) 버그에 의한 것

(B) 체크 기능 부족에 의한 것

(A)에는 SQL 인젝션^{Injection}이나 Cross-Site Scripting(XSS: 크로스-사이트 스크립팅)과 같은 유명한 취약성이 포함되어 있습니다. 이들 취약성은 원래 보안과는 관계가 없는 곳에서 발생하여 어플리케이션 전체에 영향을 미치는 특성이 있습니다. 때문에 어플리케이션 개발팀 전원에게 안전한 어플리케이션 작성을 유도하고 권고해야 할 필요가 있지만, 그에 대한 대책 및 공고 그리고 개발 규칙이 미비한 개발팀이 아직 많다는 것이 현실입니다.

한편 (B)의 예로는 디렉토리 접근 공격^{Directory Traversal} 취약성이 있습니다. 디렉토리 접근 공격이란 웹 서버 설정상의 오류 및 위치 오류를 이용하여 해당 디렉토리에 접근해 자료를 유출하는 공격으로서, 이런 종류의 취약성은 보안에 관한 개발자의 의식이 부족한 경우에 생기는 함정으로 (A)와 같이 취약성의 영향이 어플리케이션 전체에 미칩니다.

이렇듯 웹 어플리케이션 취약성은 생각지도 못한 곳에서 큰 함정이 생길 수 있습니다. 과거부터 취약성은 늘 강조되어 왔습니다. 단 우리가 만드는 어플리케이션에서의 함정은 학습을 통해 극복할 수 있는 부분이라고 할 수 있습니다.

1.4

보안 버그와 보안 기능

이번 장의 서두에서 취약성은 버그의 일종이라고 설명했지만, 어플리케이션의 보안을 위해 단지 버그를 없애는 것만으로는 충분하지 않은 경우가 있습니다. 예를 들어 통신 내용을 HTTPS로 암호화하지 않은 상태는 버그도 아니고 취약성도 아니지만 통신 내용이 도청당할 가능성이 있습니다.

HTTPS를 이용해서 통신로를 암호화하는 것과 같이, 적극적으로 안전성을 강화하는 기능을 이 책에서는 ‘보안 기능’이라고 하겠습니다. 보안 기능은 어플리케이션 요건의 하나라고 생각할 수 있으므로 ‘보안 요건’이라고 하는 경우도 있습니다.

어플리케이션의 보안을 요건과 버그로 정리하는 것은 개발 관리상 중요합니다. 버그를 없애는 것이 당연한 것처럼, 취약성을 없애는 것 역시 당연한 작업입니다. 한편 보안 기능을 요건으로 포함할 지의 여부는 비용을 고려하여 어플리케이션 발주자가 정해야 할 문제입니다.

이 책에서는 독자에게 보안 버그와 보안 기능의 다른 점을 의식하도록 하기 위해 각각의 장을 나눠서 설명합니다.

이 책의 구성은 아래와 같습니다.

1장은 이 책의 도입부로서 취약성이란 무엇인가라는 설명으로 시작하여 취약성이 생기는 원인 및 보안 버그와 보안 기능의 차이점을 설명합니다.

2장에서는 이 책의 실습 환경을 준비합니다. 이 책은 취약성 테스트를 위해 VMware를 이용한 가상 머신을 사용하여 제공하고 있습니다. 이 가상 머신 환경과 진단에 사용하는 툴들에 관해 설명합니다.

3장에서는 웹 어플리케이션 보안의 기초가 되는 HTTP와 쿠키, 세션 관리 등의 지식, SOPSame Origin Policy에 대해 설명합니다.

4장은 이 책의 중심이 되는 장입니다. 이 장에서는 웹 어플리케이션을 기능별로 구분하여 발생하기 쉬운 취약성 패턴에 대하여 원인부터 대책까지를 설명합니다.

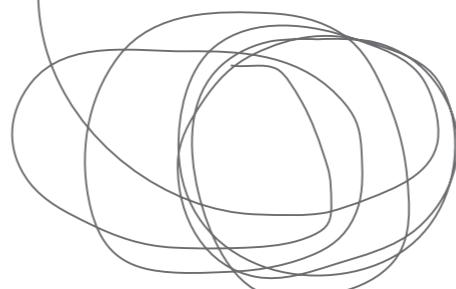
5장은 대표적인 보안 기능으로서 인증, 계정 관리, 인가, 로그 출력에 대해 설명합니다.

6장은 웹 어플리케이션 이외의 측면에서 웹사이트의 안전성을 높이기 위한 정책의 전체 그림을 설명합니다.

7장은 안전한 웹 어플리케이션 개발을 위한 개발 프로세스에 대해 설명합니다.

이 장에서는 테스트를 위해 필요한 환경을 구축합니다.
설명용 화면은 Windows7 기준이지만 Windows XP나 Windows Vista에서도 같은 방법으로 구축이 가능합니다.

해당 자료의 다운로드는 이 책의 앞부분에 정리되어 있습니다.



2.1

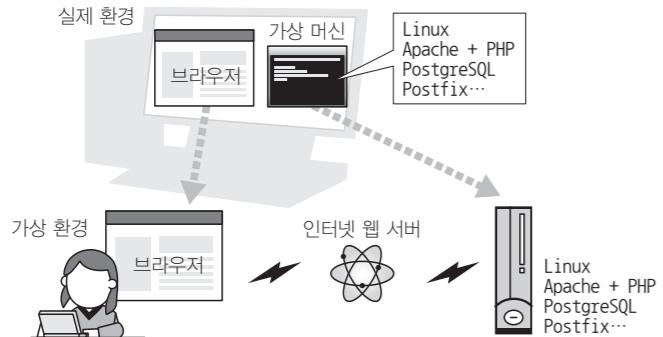
실습 환경의 개요

이 책의 테스트는 다음 환경에서 동작하는 것으로 가정하겠습니다.

- Linux(Ubuntu10.04)
- Apache2.2
- PHP5.3
- PostgreSQL8.4
- Postfix 등 Sendmail 메일 서버

윈도우를 사용하는 독자는 VMware 가상 머신을 이용하여 환경을 구축하면 됩니다.

VMware에서 리눅스가 동작하는 구조는 [그림 2-1]과 같습니다.



[그림 2-1] 이 책에서 제공하는 실습 환경

가상 머신 위의 리눅스 서버는 실제로는 자신의 로컬에서 동작하고 있지만 이것을 인터넷상의 서버라고 간주할 수 있습니다. 가상 머신의 이용으로 실제 서버와 거의 비슷한 환경을 자신의 로컬상에 구축할 수 있습니다.

이 책에서 설치할 프로그램은 다음과 같습니다.

- VMware Player(VMware 실행 환경)
- Fiddler(진단용 툴)
- 가상 머신

VMware Player와 Fiddler는 무상으로 제공되는 툴입니다. 가상 머신은 이 책에서 VMWare Player상에 동작하도록 구축한 리눅스 환경을 말합니다.

다음 절에서는 각각의 설치 방법에 대해 설명합니다.

VMware Player 설치

VMware Player란

VMware Player란 VMware사가 무상으로 제공하고 있는 가상화 소프트웨어입니다. 앞에서도 설명했듯이 이 책에서는 VMware Player를 이용하여 리눅스 서버가 동작하는 환경을 만들고 그것을 웹 서버로 간주하고 실습을 할 것입니다.

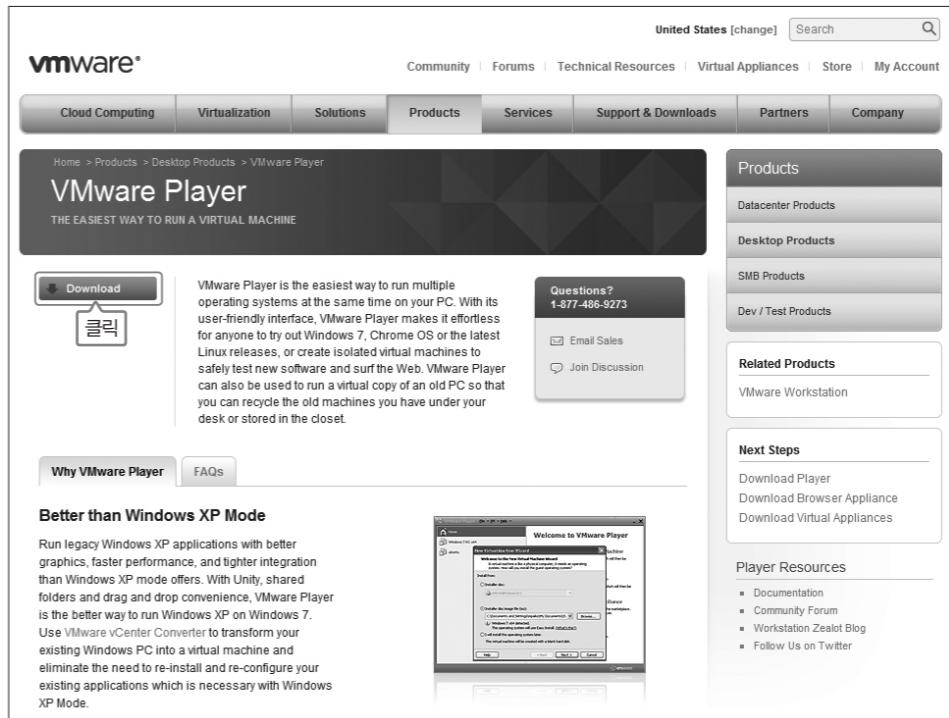
이 책의 집필 시점에서는 VMware Player의 최신 버전은 3.1.4로 동작에 필요한スペ은 다음과 같습니다.

- CPU : 표준적인 x86 호환 또는 x86-64로 Intel VT 또는 AMD-V 호환 제품 (PAE를 지원하지 않는 Pentium M 등에는 설치할 수 없습니다)
- OS : Windows XP 또는 Windows Vista, Windows7
- 메모리 : 1GB 이상
- 하드디스크 : 1GB 이상의 여분이 있는 용량

또한 하드디스크의 빈 용량이 충분하지 않은 경우는 가상 머신을 외부 디스크(USB 메모리 또는 SD 메모리 가능)에 저장할 수도 있습니다. VMware Player 자체만 설치하는 데 필요한 용량은 150MByte 정도입니다.

VMware Player 다운로드

<http://www.vmware.com>로부터 다운로드합니다. 화면 좌측 상단의 Download 버튼을 클릭하고 다운로드를 위한 등록 절차(메일 어드레스 및 이름 입력)를 마치면 다운로드 할 수 있습니다.



[그림 2-2] VMware 홈페이지로부터 최신 버전의 인스톨러 다운로드

VMware Player 셋업

홈페이지로부터 다운로드하거나 이 책에서 제공하는 VMware Player를 더블클릭하면 설치가 진행됩니다. Windows7 또는 Windows Vista에서는 사용자 계정 컨트롤 디아ログ가 표시되어 프로그램 변경 허용에 대한 여부를 묻습니다. [예]를 클릭하여 다음을 진행합니다.

가상 머신 설치 및 동작 확인

VMware Player Setup 화면이 나오면 Next를 클릭하여 인스톨을 시작합니다.



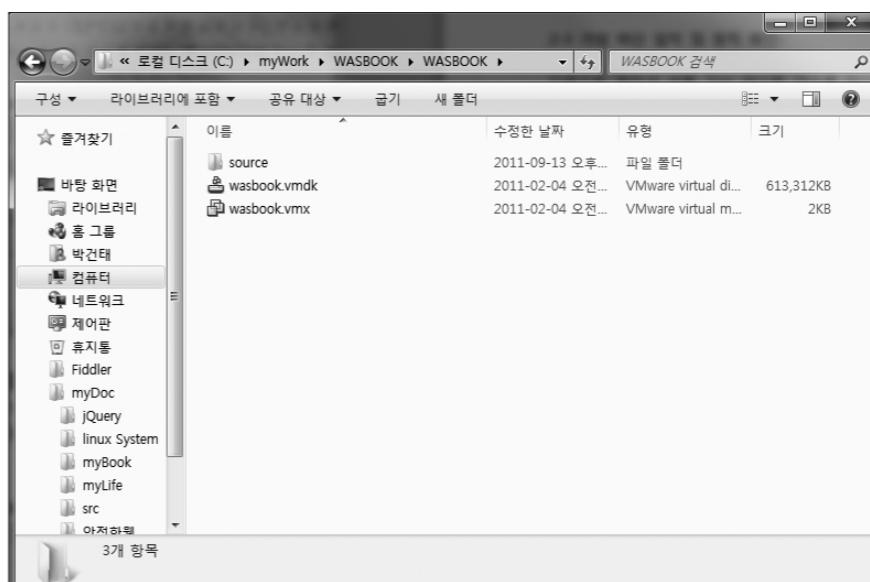
[그림 2-3] 셋업 화면

이후부터는 모든 설정을 디폴트로 하고 설치합니다. 설치되는 장소는 필요에 따라 변경 가능합니다. 다음 화면이 표시되면 인스톨에 성공한 것을 확인할 수 있습니다. 화면 지시에 따라 Windows를 재시작할 필요가 있습니다.



[그림 2-4] 셋업 완료 화면

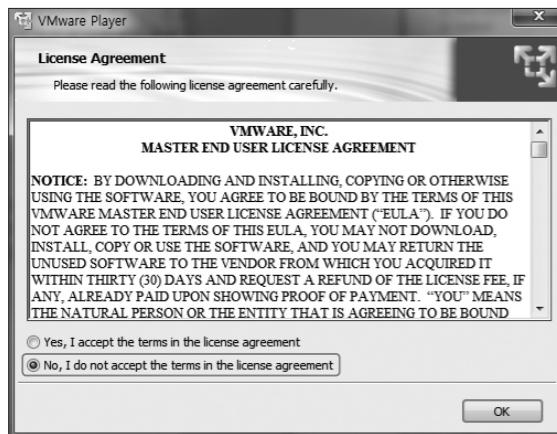
다음으로 취약성 샘플 가상 머신을 설치합니다. 가상 머신 인스톨은 WASBOOK.ZIP 파일을 풀기만 하면 준비가 끝납니다(이 책의 앞부분에 설치 가이드를 참고하세요). 압축을 푼 이후의 사이즈는 약 600M바이트이므로 여유분을 생각해서 800M 정도의 빈 드라이브에 설치하도록 합시다. USB 메모리 또는 SD 메모리에 설치하는 것도 가능합니다. 이후에 이어지는 설명에서는 C:\myWork\에 압축을 풀었다고 가정하고 설명하도록 하겠습니다.



[그림 2-5] WASBOOK 폴더 내용

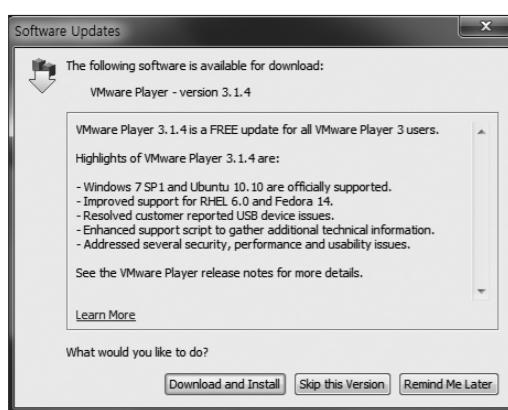
가상 머신 시작하기

WASBOOK 폴더 안의 wasbook.vmx를 더블클릭하면 VMware Player가 시작됩니다. VMware Player를 처음으로 시작할 때는 [그림 2-6]과 같이 사용 허가 동의를 요구하므로 내용을 확인한 후 동의하도록 합니다.



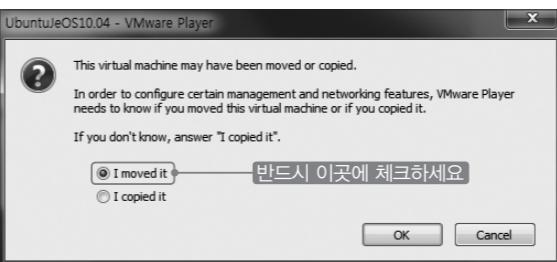
[그림 2-6] VMware Player의 사용 허가 계약

[그림 2-7]과 같이 소프트웨어 업데이트에 대한ダイ얼로그가 표시되면(버전 번호와 소프트웨어 설명은 다른 내용입니다), 업데이트를 하고자 하는 경우는 [Download and Install]을 클릭하고 업데이트를 하지 않는 경우에는 [Remind me Later]를 클릭합니다.



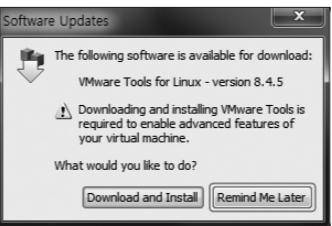
[그림 2-7] 소프트웨어 업데이트 화면

다음으로 [그림 2-8]과 같은ダイ얼로그가 표시되면 반드시 “I moved it”을 클릭하도록 합니다.



[그림 2-8] 반드시 “I moved it”을 선택

다음으로 [그림 2-9]와 같이 VMware Tools 다운로드 화면이 표시되는 경우, VMware Tools는 필요하지 않으므로 “Remind Me Later”를 클릭하면 리눅스가 부팅되기 시작합니다.



[그림 2-9] VMware Tools 업데이트 여부를 묻는ダイ얼로그

[wasbook login:] 프롬프트가 표시되면 부트가 완료된 것입니다. 여기서부터는 우선 Ctrl+G 키를 눌러 가상 머신으로 전환하여 유저 ID에는 root, 패스워드에 wasbook을 입력하고 로그인합니다. 다음 셸 프롬프트에서 ifconfig eth0를 입력하면 [그림 2-10]과 같은 화면이 표시됩니다.

```

Ubuntu 10.04.1 LTS wasbook tty1
root login: root
Password:
Last login: Fri Feb 4 00:41:48 JST 2011 on tty1
Linux wasbook 2.6.32-24-generic-pae #39-Ubuntu SMP Wed Jul 28 07:39:26 UTC 2010
i686 GNU/Linux
Ubuntu 10.04.1 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/
root@wasbook:~# ifconfig eth0
eth0      Link encap:Ethernet HWaddr 00:0c:29:3c:1f:b7
          inet addr:192.168.231.128  Bcast:192.168.231.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3c:1fb7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2498 (2.4 KB)  TX bytes:1934 (1.9 KB)
          Interrupt:17  Base address:0x1080

root@wasbook:~# 

```

[그림 2-10] 가상 머신에 로그인하여 ifconfig 커맨드 실행

여기서 inet addr:의 오른쪽에 표시되어 있는 IP 어드레스를 메모해 두도록 합시다. 이 IP 어드레스는 나중에 hosts 파일을 설정할 때 필요합니다.

가상 머신의 사용법

처음 사용해 보는 독자를 위해 가상 머신의 사용법에 대해 간단히 설명하도록 하겠습니다.

키 입력 전환

가상 머신의 화면에서 키 입력을 할 때에는 VMware Player의 윈도우를 액티브 상태로 해서 Ctrl+G 키를 누릅니다. 또는 VMware의 안쪽에 검은 부분 어딘가를 마우스로 클릭합니다. 가상 머신 키 입력을 끝내고 다른 윈도우로 키 입력을 전환하는 경우에는 Ctrl+Alt 키를 누릅니다. 관련 설명이 [그림 2-11]과 같이 VMware Player 좌측 하단에 표시됩니다.



[그림 2-11] 키 입력 전환 방법 설명

종료 방법

가상 머신을 종료하는 방법을 설명하겠습니다.

root로 로그인한 상태라면 다음의 커맨드로 종료할 수 있습니다.

```
# shutdown -h now
```

또는 로그인 프롬프트에서 유저 ID를 down이라고 입력합니다. 패스워드는 필요 없습니다. 자동으로 shutdown이 시작됩니다. 어떤 경우든 Linux 셋다운이 종료하면 자동으로 VMware Player도 종료합니다.

리눅스 조작

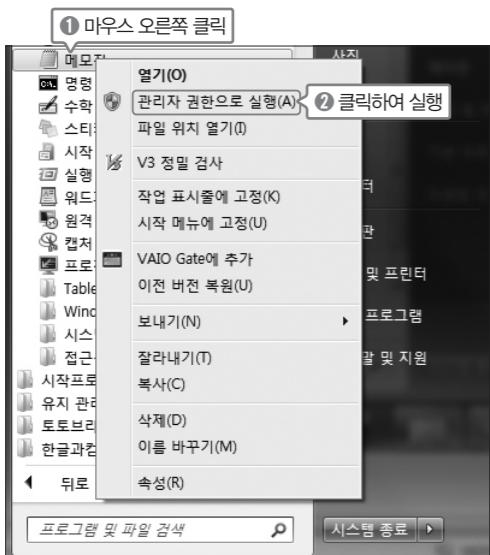
리눅스 조작에 대해서는 이 책에서는 설명하지 않으므로 Linux(Ubuntu)에 관한 해설서나 웹사이트를 참고하도록 합시다.

Hosts 파일 편집

앞으로 실습을 편하게 하기 위해 Windows의 hosts 파일에 다음 호스트명을 추가합니다.

- example.jp 취약성 사이트
- trap.example.com ... 공격자가 관리하는 어둠의 사이트

hosts(보통 C:\Windows\System32\drivers\etc\hosts) 파일은 관리자 권한이 아니면 변경할 수 없으므로(Windows Vista 또는 Windows7의 경우) 다음 페이지의 [그림 2-12]와 같이 시작 메뉴에서 메모장 메뉴를 표시하고 마우스 오른쪽을 클릭하여 [관리자 권한으로 실행]을 클릭합니다. 메모장에서 hosts 파일을 열 때에는 열기 다이얼로그에서 파일 종류를 모든 파일로 하지 않으면 hosts 파일이 표시되지 않으므로 주의합니다.



[그림 2-12] 메모장을 관리자 권한으로 실행

메모장을 사용해서 다음의 내용을 추가합니다. IP 어드레스 부분은 조금 전에 확인했던 가상 머신 IP 어드레스로 변경해 주세요.

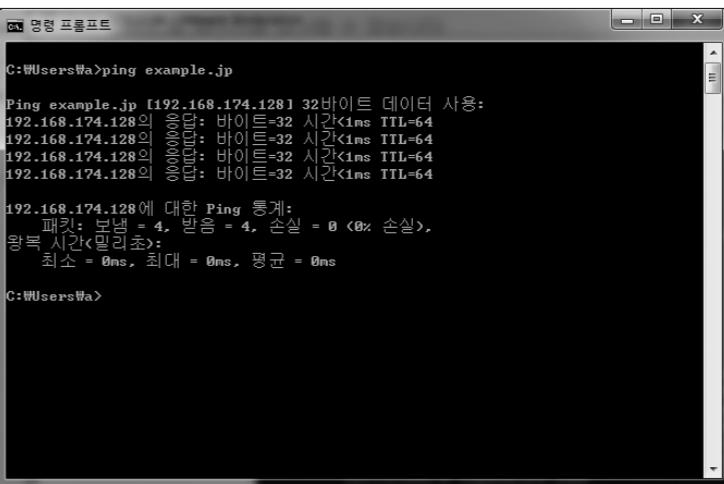
hosts 파일 편집 예

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1      localhost
#       ::1            localhost
127.0.0.1      localhost
192.168.71.128  example.jp      trap.example.com
```

이 설정으로 example.jp와 trap.example.com에는 가상 머신 IP 어드레스가 할당되었습니다. 또한 바이러스 감시 프로그램 등이 hosts 파일 변경을 감지해서 블록하는 경우가 있습니다. 이런 경우 해당 프로그램에서 블록을 해제해 주세요.

ping에 의한 통신 확인

hosts 파일의 설정이 끝났다면 Windows 커맨드 프롬프트에서 ping example.jp를 입력해서 ping 커맨드에 의해 통신이 가능한지 확인합니다.



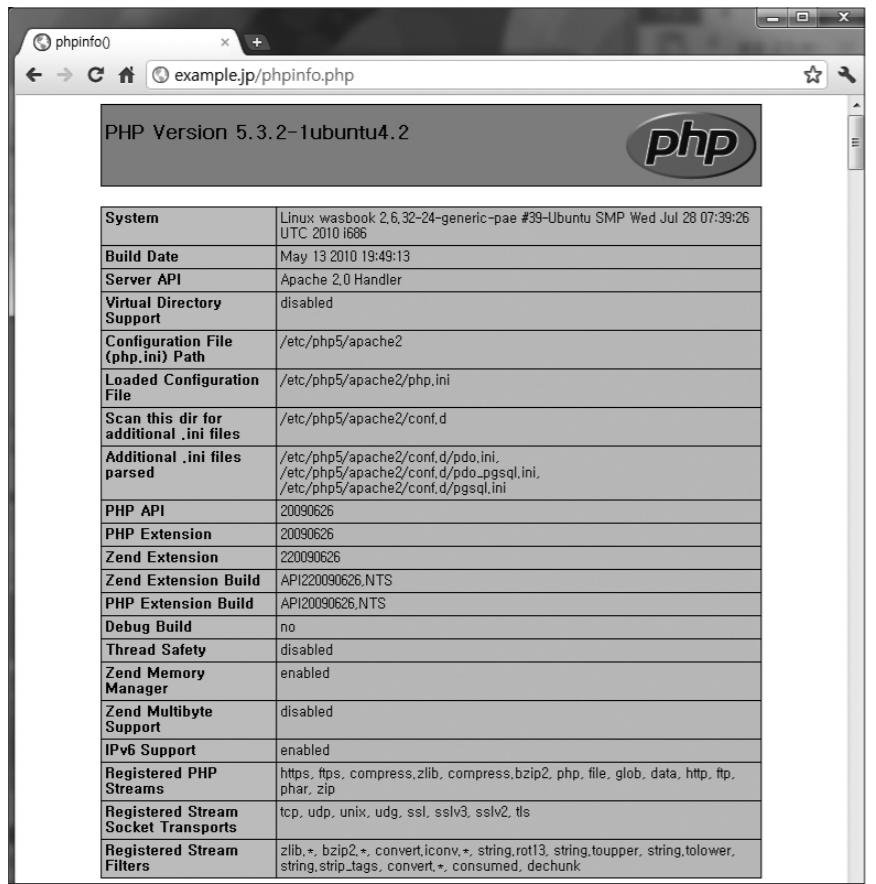
[그림 2-13] ping 커맨드를 이용한 통신 확인

(가상 머신은 동작하는 상태) 통신 상태가 확인되지 않는 경우에는 원인으로 다음과 같은 것을 생각할 수 있습니다.

- 가상 머신 시작시 “I copied it”을 선택했다.
- IP 어드레스를 잘못 적었다.
- hosts 파일의 호스트명이 잘못됐다.
- hosts 파일의 편집시 관리자 권한이 아니었다.

Apache와 PHP 동작 확인

ping으로 동작을 확인했다면 Internet Explorer(IE)를 시작하고 어드레스 바에 http://example.jp/phpinfo.php를 입력합니다. 다음 화면이 표시되는 것을 확인하도록 합시다. 설정에 문제가 있는데도 화면이 표시되지 않으면, 브라우저를 다시 시작해보세요.



[그림 2-14] 가상 머신의 웹 서버에 접속한 화면

메일 어드레스 설정과 확인

다음으로 메일 송신시의 취약성을 실습하기 위해 메일 어드레스를 설정합니다. 이 설정은 4.9절과 4.11절에서 사용하므로 지금은 설정하지 않아도 괜찮습니다.

사용하고 있는 메일 클라이언트에서 다음 계정을 설정하도록 합니다. 계정을 2종류로 설정하고 있는 이유는 취약성 연습에 2종류의 수신자를 가정으로 하고 있기 때문입니다.

[표 2-1] 실습용 메일 계정

유저	패스워드	메일 어드레스	POP3 서버	SMTP 서버
wasbook	wasbook	wasbook@example.jp	example.jp	example.jp
bob	wasbook	bob@example.jp	example.jp	example.jp

셋업 체크를 위해 wasbook에서 bob으로 메일을 송신해 보도록 합니다. bob이 수신된다면 설정이 정상적으로 완료된 것입니다.

2.4

Fiddler 인스톨

본서에서는 HTTP를 이해하기 위해 Fiddler라는 무상 툴로 HTTP 패킷을 감시하고 변경하는 것을 학습합니다. 이 절에서는 Fiddler 셋업 방법에 대해 설명합니다.

Fiddler란?

Fiddler는 Eric Lawrence가 개발한 웹 어플리케이션 디버그용 툴로서 무상으로 공개되어 있습니다. Fiddler는 Windows PC상에서 프록시로 동작하고 HTTP 통신을 감시하거나 HTTP 통신을 변경하는 것이 가능합니다. 같은 종류의 툴에는 Burp suite나 Paros 등이 있습니다.

Fiddler 셋업

Fiddler의 최신판은 <http://fiddler2.com/fiddler2/version.asp>에서 다운로드할 수 있습니다. 홈페이지에 접속하면 다음 화면을 볼 수 있습니다. Install Fiddler2에서 다운로드하여 설치할 수 있습니다. 유저 등록을 하지 않고서도 바로 다운로드가 시작됩니다.



[그림 2-15] Fiddler 다운로드 페이지

첨부 파일을 사용해서 인스톨하는 것에 대해 설명하도록 하겠습니다. 이 책에서 제공하는 Fiddler2Setup.exe를 클릭하면 Windows7 또는 Windows Vista에서는 사용자 계정 컨트롤 다이얼로그가 표시되어 프로그램 변경 허용에 대한 여부를 묻습니다. 계시자명에 Eric Lawrence라고 표시되어 있는 것을 확인하고 “예”를 클릭하여 다음으로 진행하도록 합니다.

[그림 2-16]과 같이 License Agreement에 관한 동의 화면이 나오면 “동의합니다”를 클릭한 후 화면의 디폴트 설정으로 인스톨을 진행하면 됩니다.



[그림 2-16] Fiddler2 setup License Agreement에 관한 동의 화면

만약 Windows XP에서 인스톨하는 경우라면 .NET Framework 인스톨이 필요한 경우 있습니다. Windows Vista 및 Windows7에서는 기본으로 인스톨되어 있습니다.

Fiddler 동작 확인과 간단한 사용법

Fiddler를 실행하면 [그림 2-17]과 같은 화면이 표시됩니다(Fiddler는 시작 메뉴에서도 실행할 수 있습니다). 여기서 “AutoDecode”를 클릭해 둡니다.

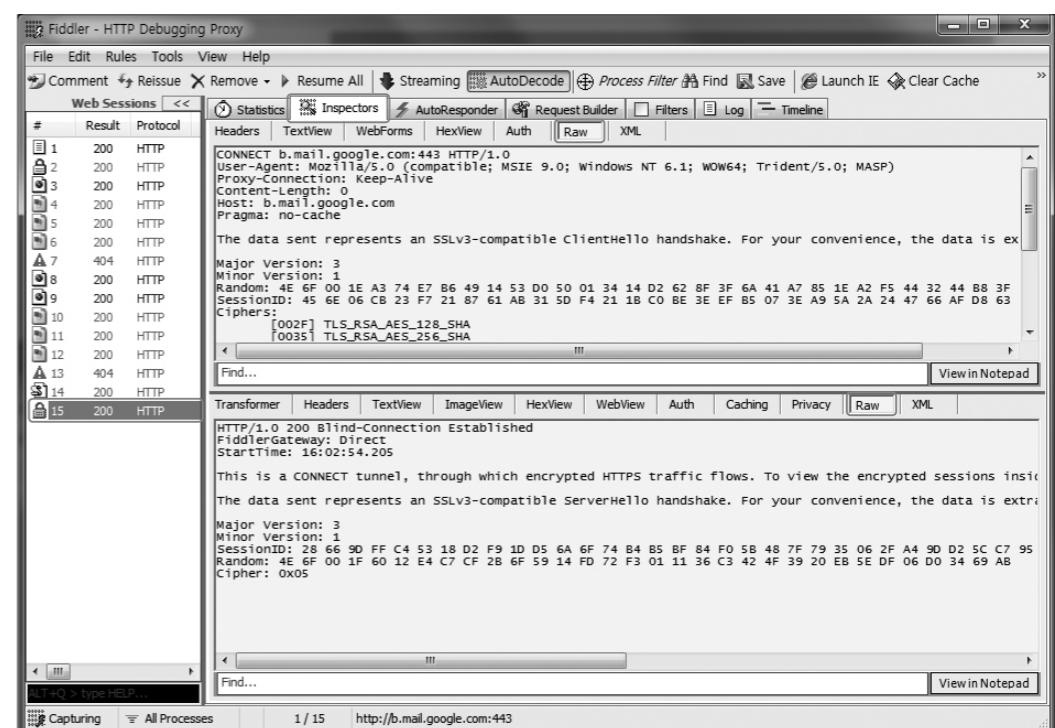
또한 [Update Announcement] 화면이 표시되는 경우는 바로 업데이트를 할 경우는 “Yes”를 클릭하고 다음에 업데이트를 할 경우는 “No”를 클릭합니다.



[그림 2-17] Fiddler 시작 화면

Fiddler는 시작시 Internet Explorer(IE)의 프록시 설정을 변경합니다. 이 때 보안 프로그램이 프록시 설정 변경을 하지 못하도록 하는 경우가 있습니다. 이 때에는 프록시를 해제해 두도록 합시다.

Fiddler의 시작을 확인한 후 IE에서 <http://example.jp/phpinfo.php>(가상 머신상의 웹 페이지)로 접속합니다. 이미 phpinfo.php를 표시하고 있는 경우는 F5를 눌러 새로고침을 하도록 합니다. [그림 2-18]과 같은 화면이 표시되는 것을 확인할 수 있습니다.



[그림 2-18] Fiddler에 의한 HTTP 통신의 감시

화면 좌측의 [Web Sessions]라는 뷰어에서 /phpinfo.php를 선택합니다. 또한 화면 상단에 있는 Tab 중에 [Inspectors] 또는 [Raw] 탭과 화면 중간 부분의 [Raw]를 선택합니다. 이 지정을 하는 이유는 HTTP를 있는 그대로 표시하도록 하기 위해서입니다.

Fiddler는 앞에서도 설명했듯이 HTTP 메시지 표시 이외에도 메시지를 변경하는 것도 가능합니다. 자세한 설명은 다음 장에서 소개하도록 하겠습니다.

이상으로 실습 환경 설정이 마무리되었습니다.

한글이 깨지는 경우 PuTTY를 이용하여 접속하기

소스를 직접 수정 및 테스트하고 싶은 독자들은 VMware로 서버에 직접 접속하면 한글이 깨지는 현상이 있습니다. 따라서 VMware로 해당 OS를 시작한 후, PuTTY 프로그램을 이용하여 접속하면 됩니다.

PuTTY 설치 후, Host Name에 IP를 넣고 접속하되, 여전히 한글이 깨지는 경우는 PuTTY Configuration에서 Category → Windows → Translation에서 Remote character set이 UTF-8로 설정되어 있는지 확인 바랍니다.

PuTTY 다운로드 페이지입니다.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

서비스	소프트웨어	버전
PHP	PHP	5.3.2
데이터베이스	PostgreSQL	8.4.4
메일 서버	Postfix	2.7.0
POP3 서버	Dovecot	1.2.9
SSH 서버	OpenSSH	5.3

Apache 루트 디렉토리

/var/www

참고 : 가상 머신 데이터 리스트

사용될 계정 목록

ID	패스워드	목적
root	wasbook	Linux 루트 계정
wasbook	wasbook	어플리케이션 관리자
alice	wasbook	메일 송신자
bob	wasbook	메일 수신자
carol	wasbook	그 외
down	없음	셧다운 용

설치한 소프트웨어

서비스	소프트웨어	버전
OS(Linux)	Ubuntu	10.04.1 LTS
웹 서버	Apache	2.2.14

Chapter 3

웹 보안의 기초

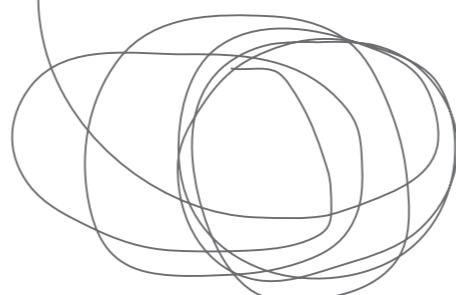
: HTTP, 세션 관리, SOP

이번 장에서는 웹 보안에 관한 기초 지식에 대해 설명합니다.

우선 이번 장의 전반부에서 HTTP와 세션 관리에 대해 설명한 후, 후반부에서는

브라우저 보안 기능 중 하나인 SOPSame Origin Policy에 대해 설명합니다.

SOP는 크로스 사이트 스크립팅 등 주요 취약성의 원리를 이해하기 위해 필요한
지식입니다.



3.1

HTTP와 세션 관리

3.1 HTTP와 세션 관리

- 31-001:현재시각
- 31-002:인력-확인-등록
- 31-010:Basic 인증 실험
- 31-020:쿠키에 의한 세션 관리

홀으로

[그림 3-1] /31/메뉴

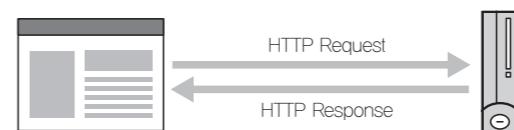
실행 결과는 [그림 3-2]와 같이 됩니다.



[그림 3-2] 현재 시간을 표시하는 스크립트

이 때 배후에서는 [그림 3-3]에서 볼 수 있듯이 다음과 같은 처리가 일어납니다.

- 브라우저에서 서버로 HTTP Request를 전송
- 서버에서 브라우저로 HTTP Response를 리턴



[그림 3-3] HTTP Request와 Response

Fiddler로 HTTP Message 확인

HTTP Request/Response Message를 Fiddler를 이용하여 볼 수 있습니다. Fiddler를 실행시킨 후, 웹 브라우저에서 조금 전 페이지를 리로드 *Reload* 합니다. 이번에는 Fiddler를 경유하여 통신하므로 Fiddler를 통해 HTTP 통신을 확인할 수 있습니다.

왜 HTTP를 배우는가?

웹 어플리케이션의 취약성에는 웹 고유의 특성에서 유래되는 것이 있습니다. 웹 어플리케이션에서는 어떤 정보가 유출되기 쉬우며, 어떤 정보가 변경될 수 있는지, 그리고 안전하게 정보를 저장하기 위해서는 어떠한 방법이 있는가와 같은 배경 지식이 부족하여 취약성이 존재하는 어플리케이션을 만들게 됩니다. 이렇듯 웹의 특성에서 유래하는 취약성을 이해하기 위해서는 HTTP 및 세션 관리에 대한 이해가 필요합니다.

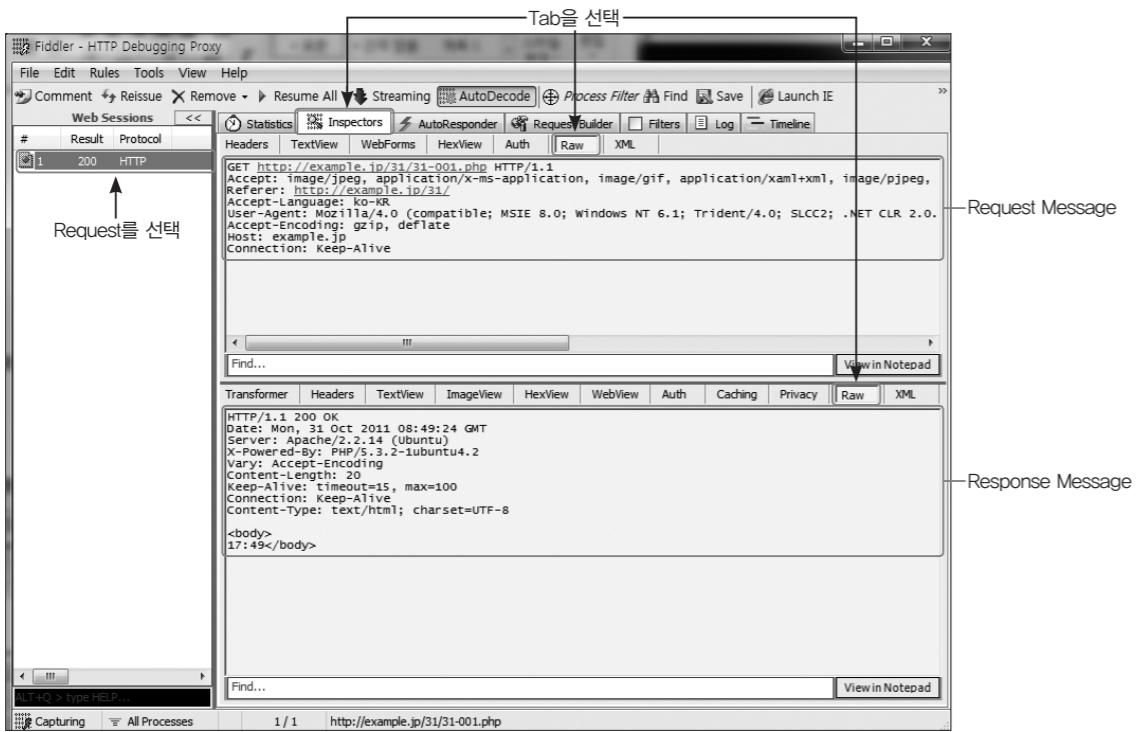
간단한 HTTP

우선 가장 간단한 HTTP 체험부터 시작해 보도록 하겠습니다. 리스트 31-001.php는 현재 시간을 표시하는 스크립트입니다.

[리스트] /31/31-001.php

```
<body>
<?php echo htmlspecialchars(date('G:i')) ; ?>
</body>
```

이를 실행하기 위해서는 <http://example.jp/31/>에서 [31-001:현재시각]을 클릭합니다 (그림 3-1).



[그림 3-4] Fiddler를 사용하여 HTTP 통신 확인

Fiddler로 HTTP 통신 과정을 보기 위해서는 [그림 3-4]와 같이 화면 윗부분의 [Inspectors] → [Raw] 탭과 화면 중간에 있는 [Raw]를 선택하고 좌측 화면에서 31-001.php의 Request를 선택합니다. 우측 화면에 표시되는 것이 브라우저와 웹 서버가 주고 받는 메시지 내용입니다.

Request Message

[그림 3-4]의 Fiddler의 Request Message(우측 화면 윗부분)에 표시되는 내용은 브라우저에서 웹 서버로 보내는 Message입니다.

Request Message의 첫 번째 행은 Request Line이라고 하는 웹 서버에 대한 명령을 나타냅니다. Request Line은 메소드, URL(URI), 프로토콜 버전 Protocol Version을 각각 공백으로 연결하여 [그림 3-5]와 같이 표현합니다. Fiddler에서는 Scheme(프로토콜)과 호스

트명 FQDN, Full Qualified Domain Name을 포함한 전체 URL이 표시되어 보이지만, 이것은 프록시 Fiddler를 경유하고 있기 때문이며 보통은 PATH 이후의 부분만 표시됩니다.

GET	/31/31-001.php	HTTP/1.1
메소드	Request URL	프로토콜 버전

[그림 3-5] Request Line

HTTP 메소드에는 GET(리소스 취득) 이외에도 POST, HEAD 등이 있습니다. GET과 POST는 HTML의 form 태그의 method 속성에 지정하는 것과 같습니다. POST에 대해서는 나중에 설명하겠습니다.

Request Message의 2행 이후는 Header라고 하며, 이름과 값을 콜론(:)으로 구분하는 형태로 되어 있습니다. [그림 3-4]에는 다양한 Header가 있지만 그 중 필수가 되는 부분은 Host¹뿐입니다. Host는 메시지를 보낼 곳의 호스트명(FQDN)과 포트 번호(80의 경우는 생략 가능)를 나타냅니다.

Response Message

한편 [그림 3-4]의 우측 아래 화면에 표시되는 내용은 웹 서버에서 리턴된 내용으로서 Response Message라고 합니다. Response Message는 [그림 3-6]과 같이 Status Line, Header, Body로 구성됩니다.

Status Line	HTTP/1.1 200 OK		
Header	Date: Mon, 31 Oct 2011 08:49:24 GMT	X-Powered-By: PHP/5.3.2-1ubuntu4.2	Server: Apache/2.2.14 (Ubuntu)
Vary: Accept-Encoding			Content-Length: 20
Keep-Alive: timeout=15, max=100			Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8			<body>
17:49</body>			
Blank Line			
Body	<body>		
	14:34</body>		

[그림 3-6] Response Message의 구조

¹ HTTP/1.0 사양에서는 Host Header 역시 생략 가능합니다.

Status Line

Status Line은 Request Message를 처리한 결과의 상태를 반환합니다(그림 3-7).

HTTP/1.1	200	OK
프로토콜 버전	Status Code	Status phrase

[그림 3-7] Status Line의 구조

[표 3-1]에서 볼 수 있듯이 Status Code는 100 단위로 증가하는 의미 있는 상태 코드로 분류됩니다. 자주 사용되는 Status Code에는 200(정상 종료), 301 및 302(리다이렉트), 404(파일이 없음), 500(내부 서버 에러) 등이 있습니다.

[표 3-1] Status Code 설명

Status Code	개요
1xx	처리가 진행되고 있다.
2xx	정상 종료
3xx	리다이렉트

Status Code	개요
4xx	클라이언트 에러
5xx	서버 에러

Response Header

Response Message의 2행 이후는 Header입니다(그림 3-6 참조). Blank Line(개행만 있는 행)은 Header의 끝을 나타냅니다. 대표적인 Header는 다음과 같습니다.

- Content_Length
Body의 바이트 수를 나타냄.
- Content_Type
MIME 타입이라고 하는 리소스 종류를 지정합니다. HTML의 경우는 text/html입니다. [표 3-2]에 주요 MIME 타입과 의미를 정리하였습니다.

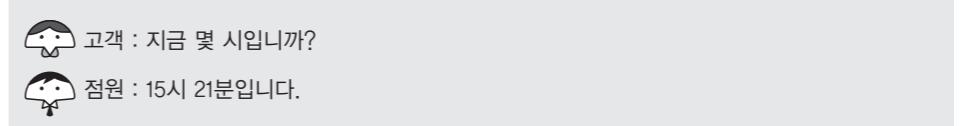
[표 3-2] 주요 MIME 타입

MIME 타입	의미
text/plain	텍스트 파일
text/html	HTML
application/xml	XML 문서
text/css	CSS
image/gif	GIF 이미지 파일
image/jpeg	JPEG 이미지 파일
image/png	PNG 이미지 파일
application/pdf	PDF 파일

세미콜론(;) 뒤에 지정된 charset=UTF-8은 HTTP Response에 대한 문자 인코딩(Encoding) 지정입니다. 문자 인코딩은 정확하게 지정해야만 합니다.

HTTP를 대화로 예를 들면

HTTP는 Request와 Response를 계속해서 주고 받으므로 우리 인간의 대화를 예로 들면 이해하기 쉽습니다. 시간 표시 스크립트를 예로 한 간단한 HTTP Message를 대화로 표현하면 다음과 같은 형식이 됩니다.



입력-확인-등록 형식의 Form의 경우를 예로 한 약간 복잡해진 HTTP Message를 보도록 하겠습니다.