

CHAPTER 13

웹 응용프로그램 취약점

목차

- 웹 응용프로그램 이해
- 주요 웹 응용프로그램 취약점



I

웹 응용프로그램 이해

웹 응용프로그램의 이해

응용프로그램

**애플리케이션(Application) 또는 앱(App)이라고도 하며
사용자가 요구하는 특정 기능을 수행 · 처리하기 위한 목적의 컴퓨터 프로그램**

웹 응용프로그램

웹 환경에서 동작되는 웹 기술 기반의 응용프로그램

➤ 웹 브라우저 또는 웹 기술이 적용된 응용프로그램을 통해 접근(접속)하여 사용하는 방식

웹 응용프로그램의 특성

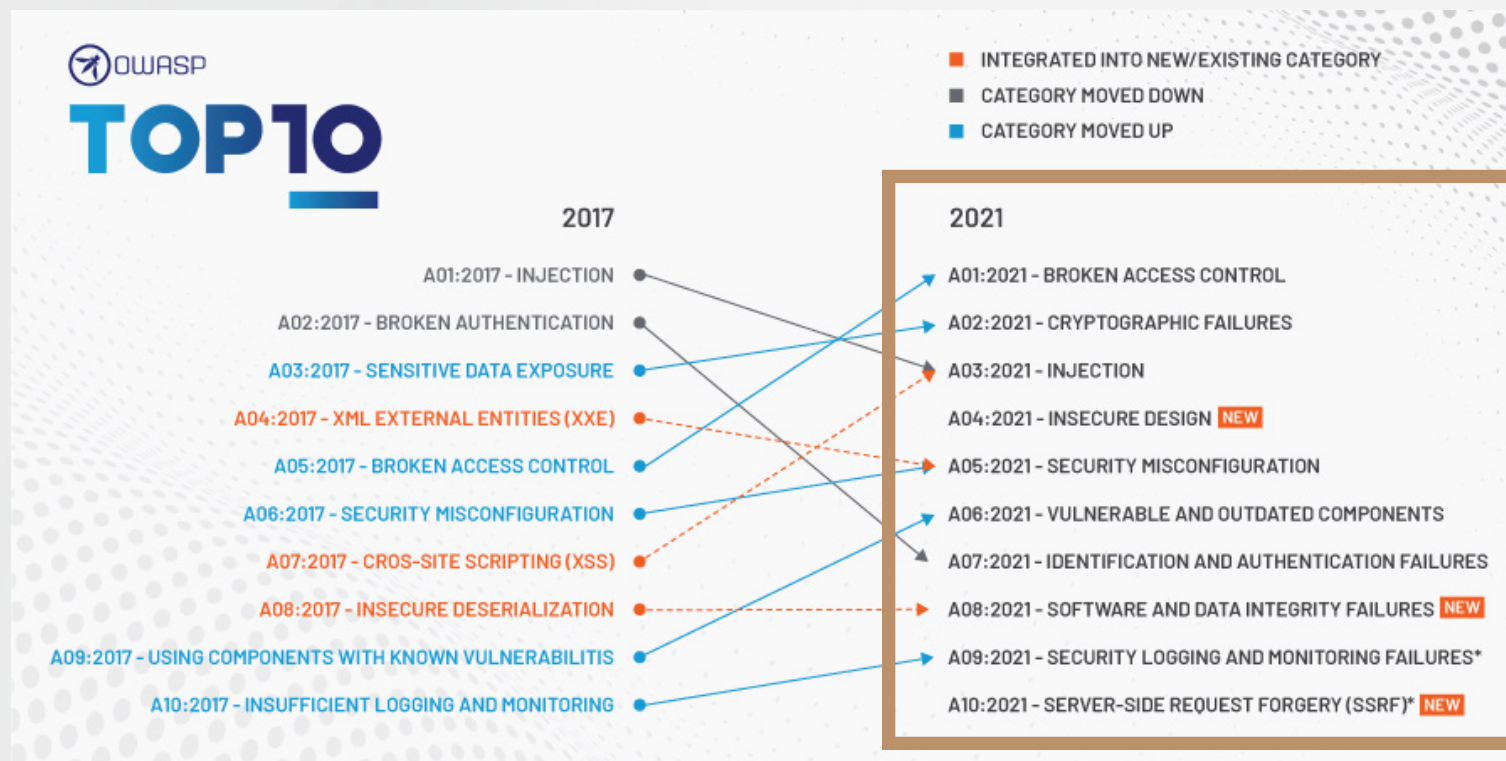
**웹 응용프로그램은 24시간 구동 · 서비스되며 정보통신망에 연결된 누구나 접근 가능한 반면,
눈에 보이지 않는 사이버공격을 정확하게 탐지하는 것에는 제한**

- 웹 응용프로그램은 HTML, 이동코드, 데이터베이스 등 다양한 구성요소들로 이루어져 있는 만큼 개발 · 관리에 소홀할 경우 공격자에 의해 악용될 수 있는 취약점이 포함될 수 있고, 알려지지 않은 취약점이 잠재되어 있을 가능성
- 웹 응용프로그램은 일방적으로 정보를 표시하는 정적 콘텐츠가 아닌, 사용자에게 의한 입력을 허용하는 동적 콘텐츠인만큼 악의적인 목적을 가진 사용자에게 의해 악성 자료가 입력될 여지 존재

OWASP TOP 10

OWASP에서 선정한 웹 응용프로그램 보안위험 목록

- > OWASP(Open Worldwide Application Security Project)는 소프트웨어 보안을 개선하기 위한 프로젝트이자 동시에 이를 주도하고 있는 비영리재단의 명칭



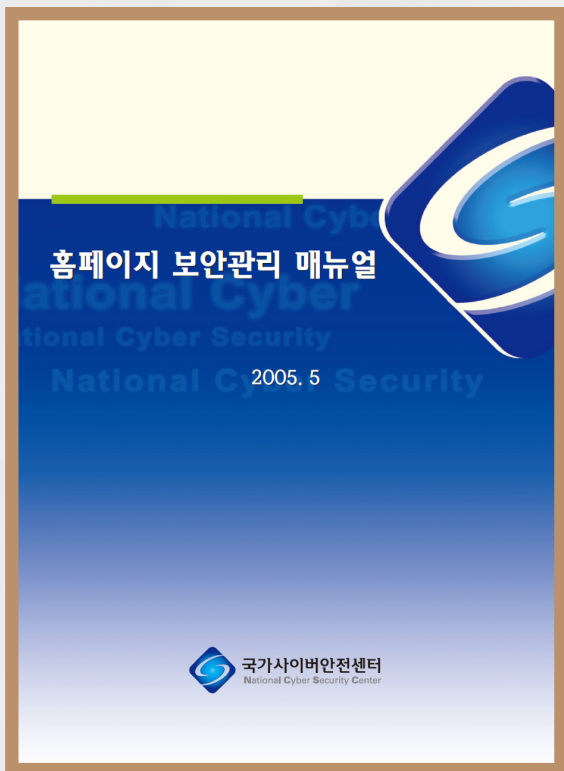
- 1 취약한 접근제어
- 2 실패한 암호화
- 3 주입(인젝션)
- 4 안전하지 않은 설계
- 5 잘못된 보안 환경구성
- 6 취약하며 오래된 구성요소
- 7 실패한 식별 및 인증
- 8 실패한 소프트웨어 및 자료 무결성
- 9 실패한 보안 기록(로깅) 및 모니터링
- 10 위조된 서버 사이드 요청 (SSRF)

웹 응용프로그램 보안 관련 가이드

2005년, 국가정보원 발간

홈페이지 보안관리 매뉴얼

- 국가정보원에서 웹 응용프로그램에 대한 8대 보안취약점(디렉토리 리스팅, 파일 다운로드, 파일 업로드, XSS, SQL 인젝션 등)을 선정하고 각 취약점별 점검 및 조치 방법을 수록한 가이드

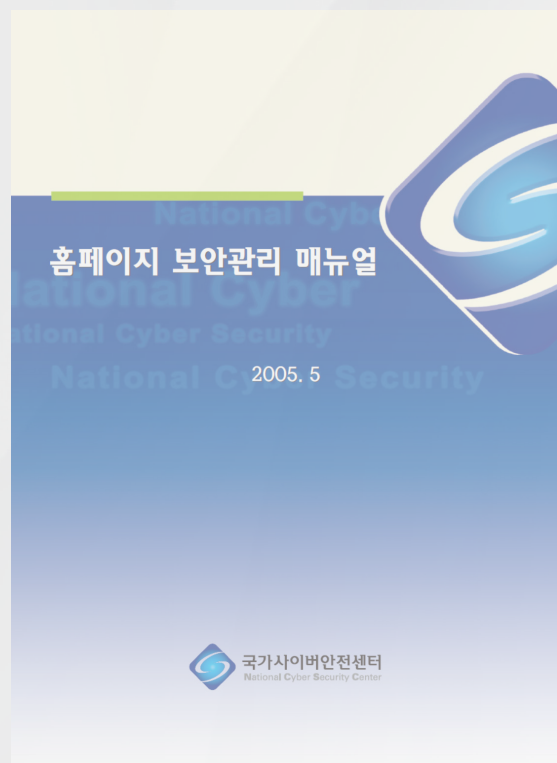


웹 응용프로그램 보안 관련 가이드

2021년, 한국인터넷진흥원 개정 발간

주요정보통신기반시설 기술적 취약점 분석·평가 방법 상세가이드

- 주요정보통신기반시설 취약점 분석·평가 기준(과학기술정보통신부 고시)에 따른 기술적 분야 취약점에 대해 10개 점검 분야별 세부 점검항목에 대한 취약점 개요, 점검대상 및 판단기준, 점검 및 조치 사례(방법)를 수록한 가이드

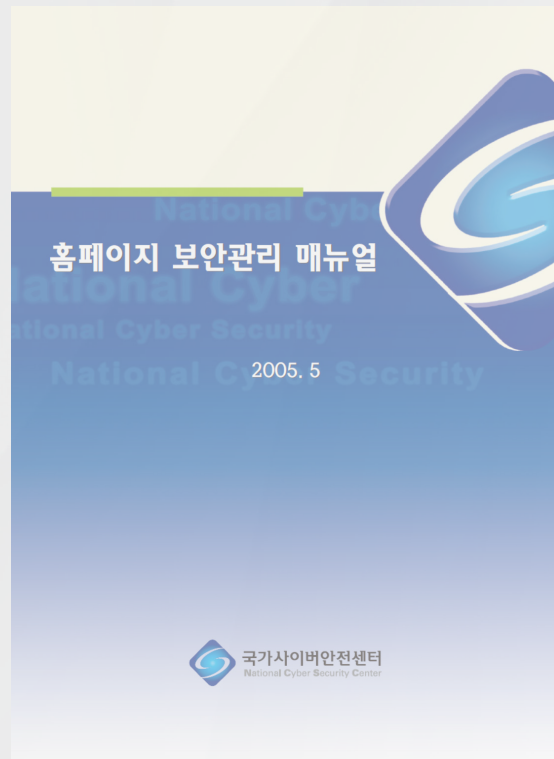


웹 응용프로그램 보안 관련 가이드

2021년, 한국인터넷진흥원 개정 발간

소프트웨어 개발보안 가이드

- ▶ 행정기관 및 공공기관 정보시스템 구축·운영 지침(행정안전부 고시)에 따라 소프트웨어 보안약점을 완화하기 위한 소프트웨어 개발보안과 단계별(분석·설계 및 구현) 보안강화 활동 및 시큐어코딩에 대해 수록한 가이드

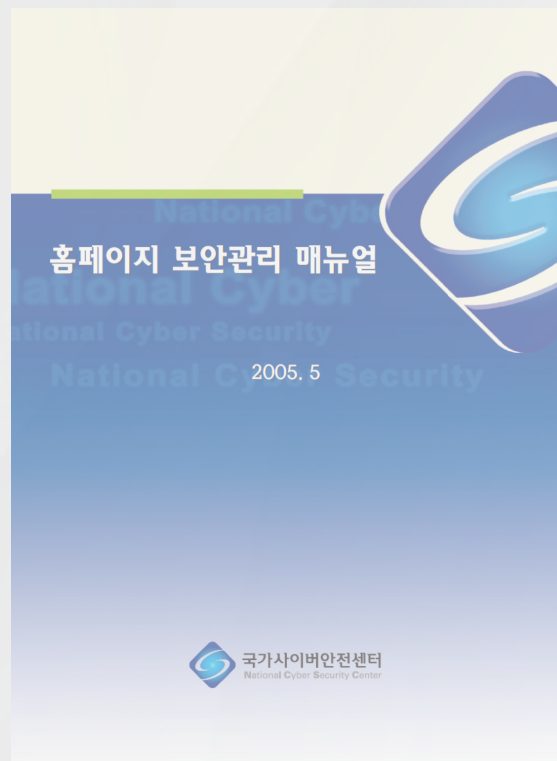


웹 응용프로그램 보안 관련 가이드

2021년, 한국인터넷진흥원 개정 발간

소프트웨어 보안약점 진단가이드

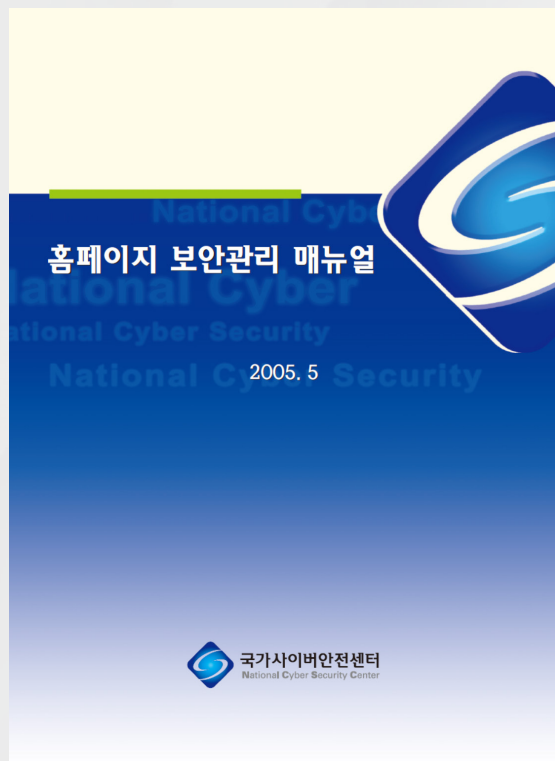
- ▶ 행정기관 및 공공기관 정보시스템 구축·운영 지침(행정안전부 고시)에 따라 소프트웨어 보안약점을 완화하기 위한 소프트웨어 개발 단계별(분석·설계 및 구현) 보안활동 및 보안약점에 대한 진단기준과 진단방법을 수록한 가이드



이외의 가이드는 한국인터넷진흥원 웹 사이트 및 국가사이버안보센터 웹 사이트 참고

> 한국인터넷진흥원(www.kisa.or.kr)

> 국가사이버안보센터(www.ncsc.go.kr)



실습

Practice

웹 응용프로그램의 구성요소 (간단한 웹 응용프로그램 작성)

실습 준비

Preparing Practices

웹 브라우저에서 다음 URL에 접속
`http://www.seculight.org:9088`

A dark blue world map is visible in the background, centered on the Atlantic Ocean. Overlaid on the map is a large cyan circle containing the Roman numeral 'II'.

II

주요 웹 응용프로그램 취약점

주요 취약점 :: 디렉토리 인덱싱 / 정보 누출 / 불충분한 인가

디렉토리 인덱싱

웹 응용프로그램 내 특정 경로에 초기 페이지에 대한 파일이 존재하지 않을 때 해당 경로 내 폴더 및 파일 목록을 표시하는 취약점

➤ 초기 페이지에 대한 파일명 예 : △index.html △index.htm △index.php △index.jsp △default.asp △default.aspx 등

정보 누출

개인정보, 계정정보, 금융정보 등 중요한 정보가 비인가자에게 노출되거나 웹 응용프로그램 오류 발생 시 웹 서버 및 응용프로그램에 대한 과도한 정보가 노출되는 취약점

➤ 웹 서버 및 응용프로그램에 대한 과도한 정보 예 : △웹 서버 구성 정보 △웹 응용프로그램 관련 정보 △데이터베이스관리시스템 정보 등

불충분한 인가

접근제어가 필요한 중요한 페이지에 대한 통제 수단이 미흡하여 비인가자가 접근할 수 있도록 허용하는 취약점

➤ 중요한 페이지 예 : △개인정보 변경 △합격자 조회 △비공개 게시물 △1:1 상담 △성적 조회 △관리자 기능 등

주요 취약점 :: 디렉토리 인덱싱 / 정보 누출 / 불충분한 인가

문제점

- 공격자는 **디렉토리 인덱싱**으로 인해 **노출된 폴더 및 파일 목록**을 토대로 웹 응용프로그램의 **구조를 파악**하여 **공격에 활용**하기 위한 **단서를 수집**할 가능성
- **디렉토리 인덱싱**으로 인해 **설정 파일 · 백업 파일 · 임시 파일**이 **노출**되거나 **정보 누출**로 인해 웹 서버 및 웹 응용프로그램에 대한 **과도한 정보**가 **노출**되면 공격자로 하여금 **공격에 활용**할 수 있는 **단서를 유추**하게 할 가능성
- **불충분한 인가**로 인해 **비인가자**가 **URL 매개변수(인자) 값 변경** 등과 같은 **단순한 방법**만으로 **중요한 페이지**에 **접근**하여 **민감한 정보**를 **열람**하거나 **변조**할 가능성
- **개인정보, 계정정보, 금융정보 등 중요한 정보**가 권한이 없는 **비인가자**에게 **노출**되는 그 자체가 곧 **정보유출 사고**






주요 취약점 :: 디렉토리 인덱싱 / 정보 누출 / 불충분한 인가

웹 브라우저 화면

주소

디렉토리 인덱싱

10.X.X.X - /inc

<u>Name</u>	<u>Last Modified</u>	<u>Size</u>	<u>Description</u>
 [Parent Directory]		-	
 config.conf	06-Feb-2020 09:31	2.7K	
 lib.bak	06-Feb-2020 09:31	108K	
 lib.php	06-Feb-2020 09:30	113K	
 temp.zip	06-Feb-2020 09:31	1.8K	
...			






Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.2.34 ... Python/2.7.5 Server at 10.X.X.X Port 80

주요 취약점 :: 디렉토리 인덱싱 / 정보 누출 / 불충분한 인가

웹 브라우저 화면

주소

10.X.X.X - /inc

<u>Name</u>	<u>Last Modified</u>	<u>Size</u>	<u>Description</u>
 [Parent Directory]		-	
 config.conf	06-Feb-2020 09:31	2.7K	
 lib.bak	06-Feb-2020 09:31	108K	
 lib.php	06-Feb-2020 09:30	113K	
 temp.zip	06-Feb-2020 09:31	1.8K	

정보 누출 (웹 서버 및 웹 응용프로그램에 대한 과도한 정보)

Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.2.34 ... Python/2.7.5 Server at 10.X.X.X Port 80

주요 취약점 :: 디렉토리 인덱싱 / 정보 누출 / 불충분한 인가

웹 브라우저 화면			
주 소	http://10.X.X.X/inform.ph?SEQ=1	불충분한 인가	GO
<div><p>귀하의 지원에 감사드립니다.</p><p>제한된 여건으로 인해 함께 할 수 없어 아쉽습니다.</p><p>다음에 다시 뵙기를 바랍니다.</p></div>			

주요 취약점 :: 디렉토리 인덱싱 / 정보 누출 / 불충분한 인가

웹 브라우저 화면

주 소	http://10.X.X.X/inform.php?SEQ=2	불충분한 인가	GO
-----	----------------------------------	---------	----

최종 합격을 축하드립니다!

이후의 계획에 대한 사항은 다음 내용을 참고하시기 바랍니다.

...

실습 1

Practice 1

주요 취약점 :: 파일 다운로드

파일 다운로드

웹 응용프로그램 내 **파일 다운로드 기능 호출 시 허용된 경로가 아닌 다른 경로에 존재하는 권한 없는 파일에 접근할 수 있는 취약점**

➤ 허용된 경로가 아닌 다른 경로 예: 웹 서버 내 시스템 디렉토리 등

문제점

공격자가 **매개변수(인자)를 조작하여 웹 서버 내 중요한 파일을 다운로드하면 정보 유출이나 2차 공격의 원인으로 작용될 가능성**

➤ 중요한 파일 예: △데이터베이스 파일 △웹 응용프로그램 설정값이 포함된 파일 △시스템 계정 및 비밀번호 파일 △시스템 설정 파일 등

주요 취약점 :: 파일 다운로드

웹 브라우저 화면

다운로드 기능 호출 시 매개변수 확인

주소	http://10.X.X.X/download.php?path=data/캡처%20화면.png			GO
----	--	--	--	----

질의응답 게시물 열람

HOME > 채용공고 > 질의응답

제목	채용 공고문이 열리지 않아요!		
작성자	이현호	작성일시	20XX-MM-DD HH:MM:SS
첨부파일	다운로드	열기	캡처 화면.png

안녕하세요? 이번에 귀 기관에 지원한 채용지원자입니다.
공지사항에 게시된 공고문을 다운로드하려고 하는데 오류가 뜨네요...
오류 화면을 첨부했으니, 확인 부탁드립니다요~~ ㅠ.ㅠ

목록보기

주요 취약점 :: 파일 다운로드

웹 브라우저 화면		매개변수(인자) 조작	
주 소	http://10.X.X.X/download.php?path=../../../../etc/passwd		GO
질의응답 게시물 열람		HOME > 채용공고 > 질의응답	
제 목	채용 공고문이 열리지 않아요!		
작 성 자	이현호	작성일시	20XX-MM-DD HH:MM:SS
첨부파일	다운로드 열기 캡처 화면.png		
안녕하세요? 이번에 귀 기관에 지원한 채용지원자입니다. 공지사항에 게시된 공고문을 다운로드하려고 하는데 오류가 뜨네요... 오류 화면을 첨부했으니, 확인 부탁드립니다요~~ π.π			
			목록보기

주요 취약점 :: 파일 다운로드

웹 브라우저 화면

주소

질의응답 게시물 열람 HOME > 채용공고 > 질의응답

다운로드 파일 내용 확인

메모장 화면

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
...
```

실습 2

Practice 2

주요 취약점 :: 파일 업로드

파일 업로드

웹 응용프로그램 내 **파일 업로드 기능을 활용하여 파일 업로드 시 웹 서버에서 실행 가능한 파일을 통제하지 못하는 취약점**

➤ 웹 서버에서 실행 가능한 파일 예: △웹 문서 △웹셸(WebShell) 등

문제점

공격자가 웹셸 등 웹 서버에서 실행 가능한 파일을 업로드하고 실행하여 시스템 관리자 권한을 획득하거나 연결된 인접 서버로 횡적 이동(Lateral Movement)할 가능성

➤ 연결된 인접 서버 예: △데이터베이스 서버 △파일저장 서버 △백업 서버 등

주요 취약점 :: 파일 업로드

웹 브라우저 화면

질의응답 게시물 작성 HOME > 채용공고 > 질의응답

제 목	테스트
첨부파일	<div>파일선택 ...</div>
점 검	테스트

저장하기 목록보기



contents.php

주요 취약점 :: 파일 업로드

웹 브라우저 화면

질의응답 게시물 작성

HOME > 채용공고 > 질의응답

제 목	테스트	
첨부파일	파일선택 ...	Z:\문서\contents.php
점검 테스트		
		저장하기
		목록보기

첨부파일 선택 후 저장

주요 취약점 :: 파일 업로드

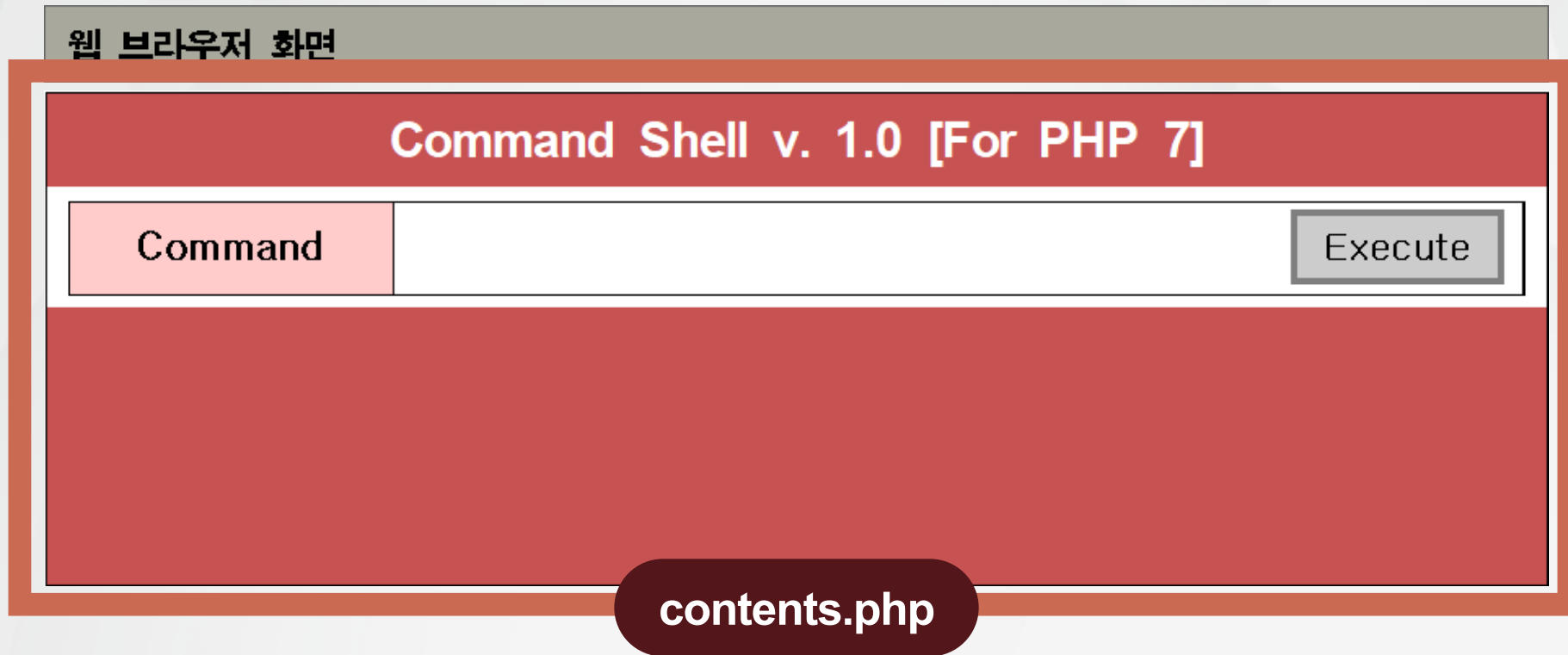
웹 브라우저 화면

질의응답 게시물 열람

HOME > 채용공고 > 질의응답

제 목	테스트		
작 성 자	점검계정1	작성일시	20XX-MM-DD HH:MM:SS
첨부파일	<div>다운로드</div> <div>열기</div> contents.php		
점검 테스트	저장된 첨부파일 확인		
			<div>수정하기</div> <div>목록보기</div>

주요 취약점 :: 파일 업로드



주요 취약점 :: 파일 업로드

웹 브라우저 화면

Command Shell v. 1.0 [For PHP 7]

Command

Execute

서버 운영체제 명령어 입력 시도

주요 취약점 :: 파일 업로드

웹 브라우저 화면

Command Shell v. 1.0 [For PHP 7]

Command

cat /etc/passwd

Execute

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

...

서버 운영체제 명령어 실행 결과

실습 3

Practice 3

주요 취약점 :: 크로스 사이트 스크립팅(XSS)

크로스 사이트 스크립팅(Cross Site Scripting, XSS)

**웹 문서 또는 콘텐츠에 이동코드를 삽입함으로써
해당 웹 문서 또는 콘텐츠에 노출된 다른 사용자를 공격할 수 있게 허용하는 취약점**

- 이동코드는 프로그램 코드를 다른 컴퓨터로 전송(이동)하여 실행하게 하는 기술로
대표적으로 JavaScript, VBScript 등이 있으며, 웹 브라우저의 스크립트 엔진을 통해 해석 · 실행

문제점

**공격자가 게시물 내 악의적인 기능을 수행하는 이동코드를 삽입하여
해당 게시물을 열람하는 다른 사용자의 로그인 세션(세션 쿠키)을 탈취하거나
위조된 가짜 사이트로 납치하거나 악성코드의 강제 실행 · 감염을 유도할 가능성**

- DBD(Drive by Download) 공격
오버플로우(Overflow)를 유발하거나 웹 브라우저 내 스크립트 엔진의 버그를 악용하는 방식으로 악성코드의 강제 실행 · 감염 유도

주요 취약점 :: 크로스 사이트 스크립팅(XSS)

웹 브라우저 화면

질의응답 게시물 작성 HOME > 채용공고 > 질의응답

제 목	테 스트
<div>게시물 본문에 이동코드(JavaScript) 작성</div> <div><script>alert(document.cookie);alert('까꿍!!');</script></div>	
<div>저장하기</div> <div>목록 보기</div>	

주요 취약점 :: 크로스 사이트 스크립팅(XSS)

웹 브라우저 화면

질의응답 게시물 열람 HOME > 채용공고 > 질의응답

제 목	테스트 2
<div>작성된 게시물 열람 시 이동코드 실행</div>	
<div>수정하기 목록보기</div>	

주요 취약점 :: 크로스 사이트 스크립팅(XSS)

웹 브라우저 화면

질의응답 게시물 열람

HOME > 채용공고 > 질의응답

제 목	테스트 2
	<div>이동코드가 실행된 결과</div> <div><div>알림</div><div>PHPSESSID=c72983d39d47646268c8a87fb6fc2f1c</div><div>확인</div></div>

수정하기

목록보기

주요 취약점 :: 크로스 사이트 스크립팅(XSS)

다양한 XSS 유형

• Stored XSS

- 사용자가 제공한 이동코드가 서버에 저장되어 다른 사용자가 해당 이동코드가 포함된 콘텐츠(게시물 등) 열람 시 실행되어 영향을 미치는 유형

* 예: `<script>alert('XSS');</script>`

• Reflected XSS

- 사용자가 제공한 이동코드가 즉시 반영되는 기능(검색결과 등)이 존재하는 경우
이를 토대로 이동코드가 포함된 URL을 구성하고 다른 사용자가 해당 URL에 접근 시 실행되어 영향을 미치는 유형

* 예: `http://example.com/search?q=<script>alert('XSS');</script>`

• DOM-based XSS

- 웹 브라우저의 렌더링 과정에서 동적으로 생성된 객체(DOM)에 이동코드가 포함되는 유형

* 예: `document.write('<script>location.replace("http://example.com");</script>');`

주요 취약점 :: 크로스 사이트 스크립팅(XSS)

XSS 대응방책

- **Input Validation (입력값 검증)**

- 사용자가 제공하는 값 중 XSS를 유발시킬 가능성이 있는 문자 등의 유무를 검증하는 기법으로, 문자열 필터링 등으로 구현
 - * 예 : 사용자가 입력한 내용 중 부등호(<, >)를 정규표현식을 사용하여 HTML Entities(< , >)로 치환

- **Output Encoding (출력값 부호화)**

- 출력 과정에서 XSS를 유발시킬 가능성이 있는 문자 등이 웹 브라우저의 렌더링 과정에서 실행되지 않도록 부호화하는 기법
 - * 예 : 저장되어 있는 내용을 불러오는 과정에서 부등호(<, >)를 정규표현식을 사용하여 HTML Entities(< , >)로 치환

- **Content Security Policy (콘텐츠 보안 정책)**

- HTTP 헤더 또는 메타 태그에 허용할 콘텐츠에 대한 보안 정책을 명시하여 허용되지 않은 출처의 콘텐츠를 제한하는 기법
 - * 예 : 사이트 자체의 출처(하위 도메인 제외) 및 신뢰할 수 있는 도메인(example.com)과 해당 도메인의 하위 도메인(*.example.com)의 콘텐츠를 허용

```
Content-Security-Policy: default-src 'self' example.com *.example.com
```

실습 4

Practice 4

주요 취약점 :: SQL 인젝션

SQL 인젝션(Injection)

사용자가 입력한 값에 의해 변형된 **SQL 구문**으로 인해
데이터베이스관리시스템(DBMS)에 대한 **부당한 접근**을 유발하는 취약점

➤ **SQL(Structured Query Language)**

데이터베이스의 형식 · 구조를 정의하거나 데이터를 조회 · 검색하거나 입력 · 변경 · 삭제 등의 조작을 위해 사용되는 구조적 질의 언어

문제점

공격자가 입력 필드에 **SQL 인젝션**을 유발시킬 수 있는 **SQL 구문**을 입력하여
DBMS를 비정상적으로 조작함으로써 권한이 없는 정보를 무단 열람하거나
새로운 자료를 무단 저장하거나 기존의 자료를 변조 또는 무단 삭제할 가능성

➤ SQL 인젝션을 유발시킬 수 있는 SQL 구문을 자동으로 생성 · 입력하는 자동화 도구 활용 가능

주요 취약점 :: SQL 인젝션

웹 브라우저 화면

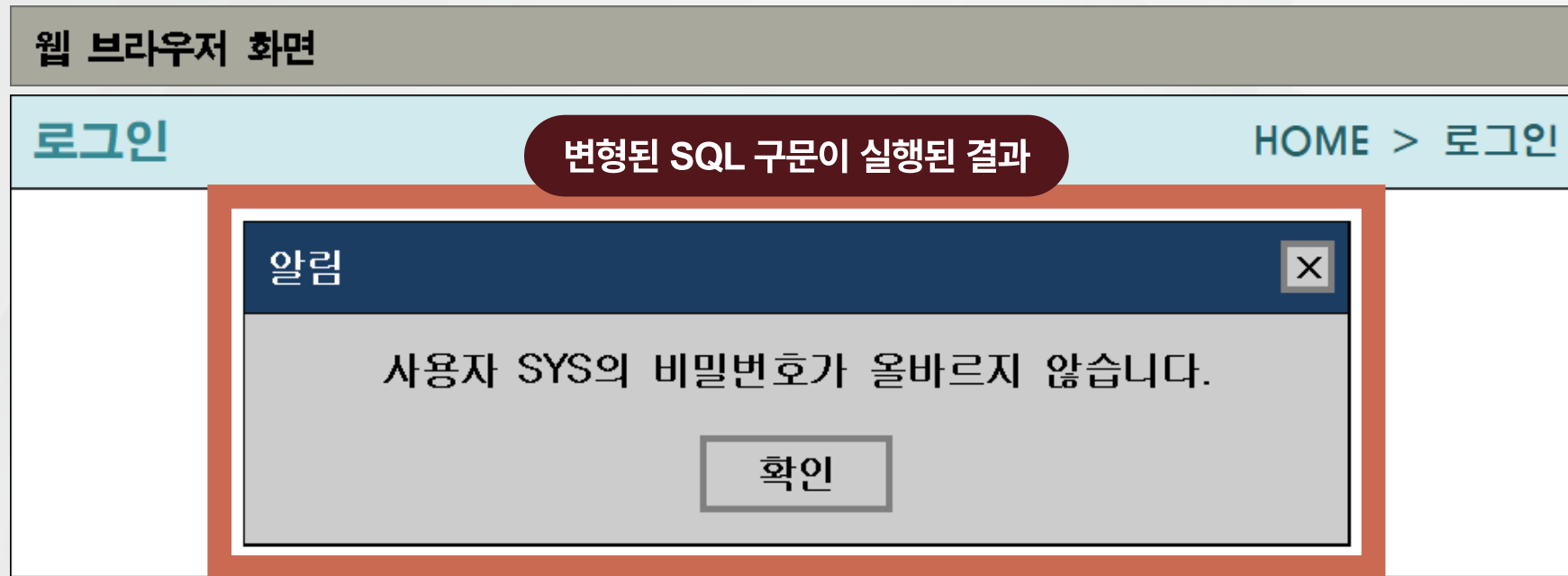
로그인 HOME > 로그인

SQL 구문 변형 유발 구문 삽입

아 이 디	' OR 1=1; #
비밀번호
로그인	계정 생성 비밀번호 찾기

로그인 시도

주요 취약점 :: SQL 인젝션



주요 취약점 :: SQL 인젝션

다양한 SQL 인젝션 유형

• Error-based SQL Injection

➤ 일반적으로 널리 알려진 SQL 인젝션 기법으로, 논리 오류 등 오류 발생을 유도하여 데이터베이스 구조 등 공격에 유용한 정보를 획득하는 유형

* 예:

```
SELECT * FROM users WHERE id = 1 AND password = ' ' OR 1=1; #' ;
```

• Union-based SQL Injection

➤ UNION(합집합) 연산자를 사용하여 같은 구조의 테이블(또는 같은 구조로 만들어서)을 이어 붙이는 방식으로 추가 정보를 획득하는 유형

* 예:

```
SELECT username, password FROM users WHERE username = 'admin'  
UNION SELECT credit_card_number, 1 FROM credit_cards;
```

• Blind SQL Injection

➤ 참/거짓 조건 등을 통해 조건에 부합되는 데이터의 존재 여부를 확인하는 방식(스무고개)으로 정보를 유추해가는 유형

* 예:

```
SELECT * FROM financial WHERE balance >= 12000;  
SELECT * FROM financial WHERE balance >= 11000;
```

주요 취약점 :: SQL 인젝션

SQL 인젝션 대응방책

• Prepared Statements (준비된 구문)

- 웹 응용프로그램의 질의문과 사용자가 제공하는 값을 분리하는 기법으로, 질의문을 먼저 준비해둔 상태에서 값을 바인딩하는 방식으로 구현
즉, 사용자가 값으로 SQL 구문을 입력하더라도 웹 응용프로그램 내 SQL 구문에 변형이 발생하는 것을 방지

* 예:

```
SELECT username FROM users WHERE username = ?
```

• Input Validation (입력값 검증)

- 사용자가 제공하는 값 중 SQL 인젝션을 유발시킬 가능성이 있는 문자 등의 유무를 검증하는 기법으로, 문자열 필터링 등으로 구현
단, Prepared Statements에 비해서는 불완전한 대책

* 예: 사용자가 입력한 값 중 작은따옴표(')를 정규표현식을 사용하여 HTML Entities(')로 치환

• Object-Relational Mapping (객체관계 매핑)

- 객체지향 프로그래밍 언어(Object Oriented Programming Languages)에서 객체 코드를 관계형 데이터베이스에 연결하는 방식으로 구현
단, 데이터베이스 계층을 추상화하기 때문에 데이터에 대한 사전 지식이 필요하며, 데이터베이스 성능에 영향을 미칠 가능성이 있는 대책

* 예: Spring Data JPA 등

실습 5

Practice 5