

SECURE CODING

고언약 기윤호

이상훈(18살) 한국디지털미디어고

「
한 두 세 네 다섯
여섯 일곱 여덟
아홉 열 한 두 시
자 이 삼 사 오 십
정 일 이 삼 사 육
오 오 칠 팔 구 분
」

상훈아 언젠가 너는 꼭 보
석처럼 빛날거야:)

양은민(18학번) 앱프로그래밍 과제

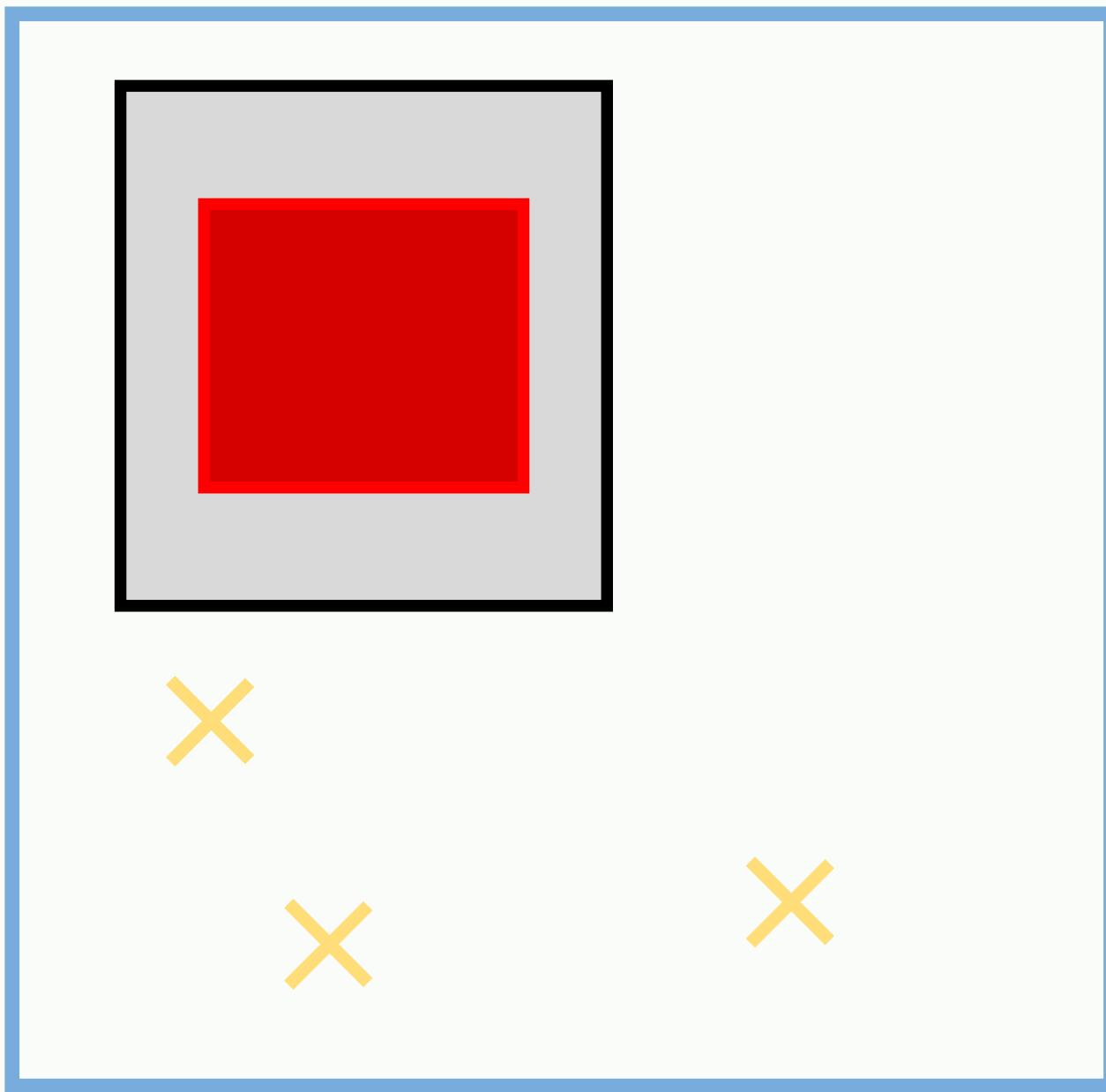


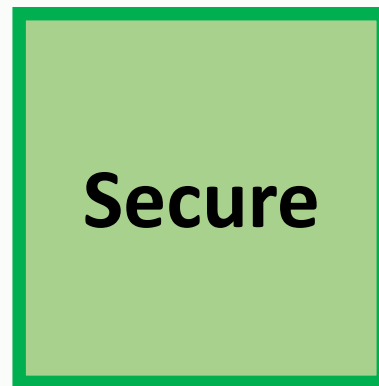
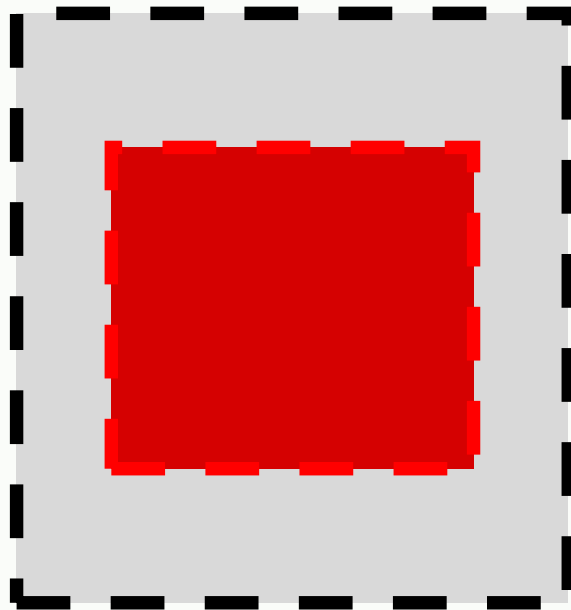
보안 취약점

해킹 등 실제 침해사고에 이용되는 시스템 상의 보안 약점

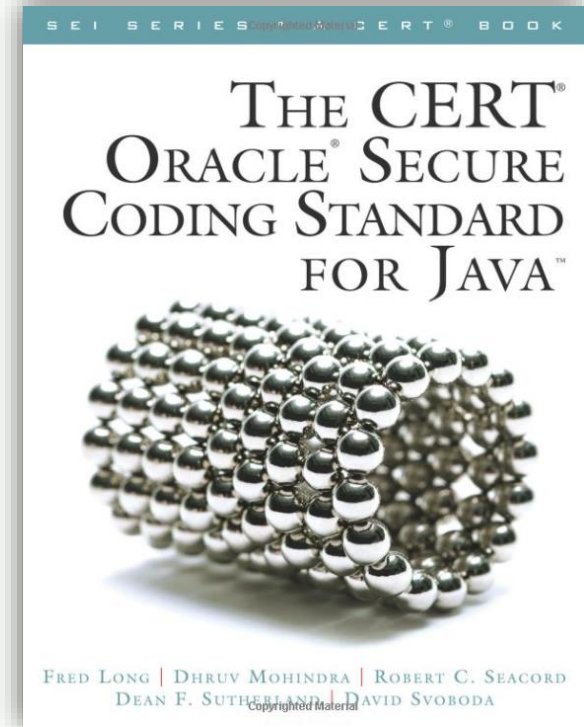
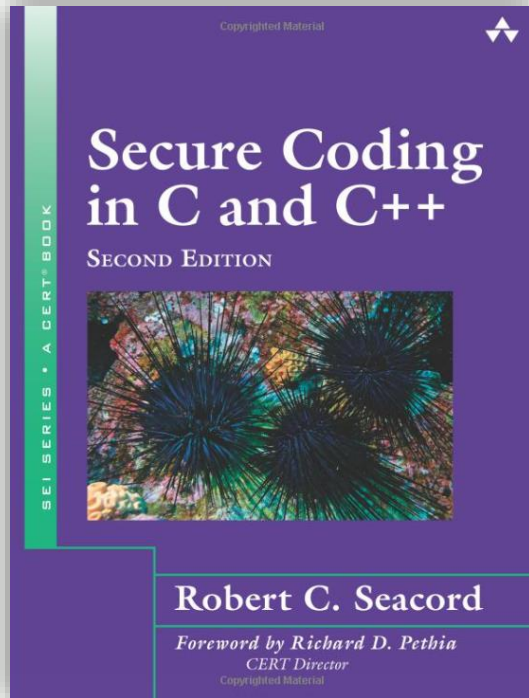
보안 약점

보안취약점의 원인이 되는 시스템 상의 보안 결함 오류

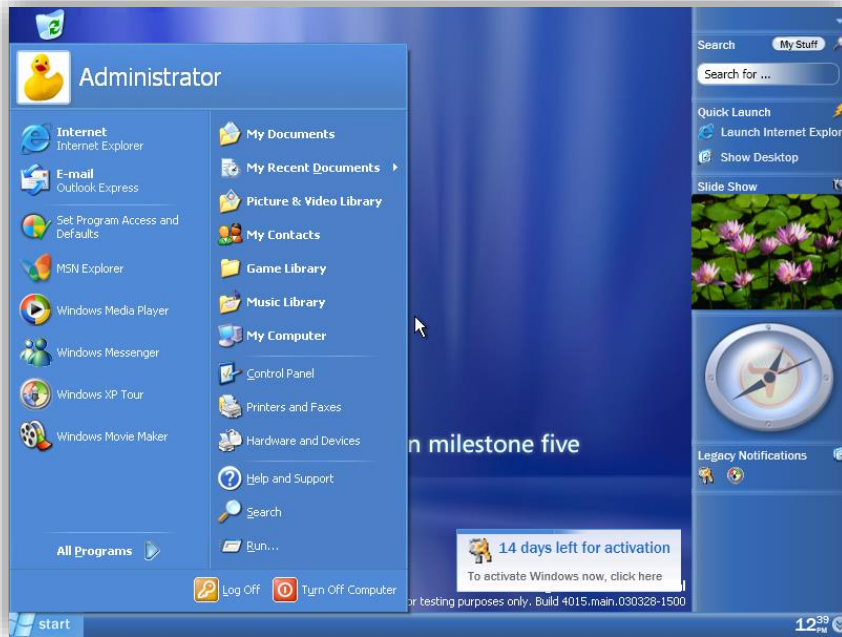




다양한 시큐어코딩



MS-SDL

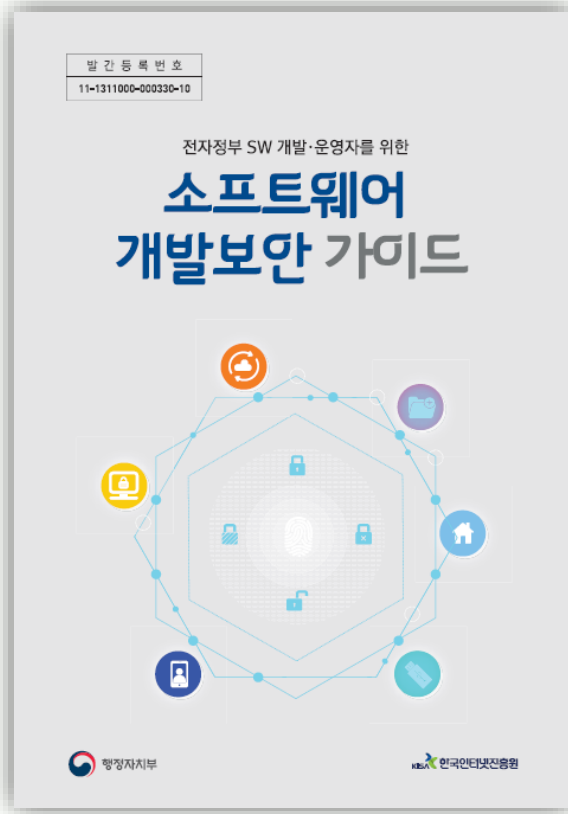


Vista bata(Longhorn)



WindowXP Blue Screen

소프트웨어 개발보안 가이드



47개의 시큐어 코딩 가이드 제공

OWASP TOP10 공격에 대한 시큐어 코딩

SW 개발 보안 의무제

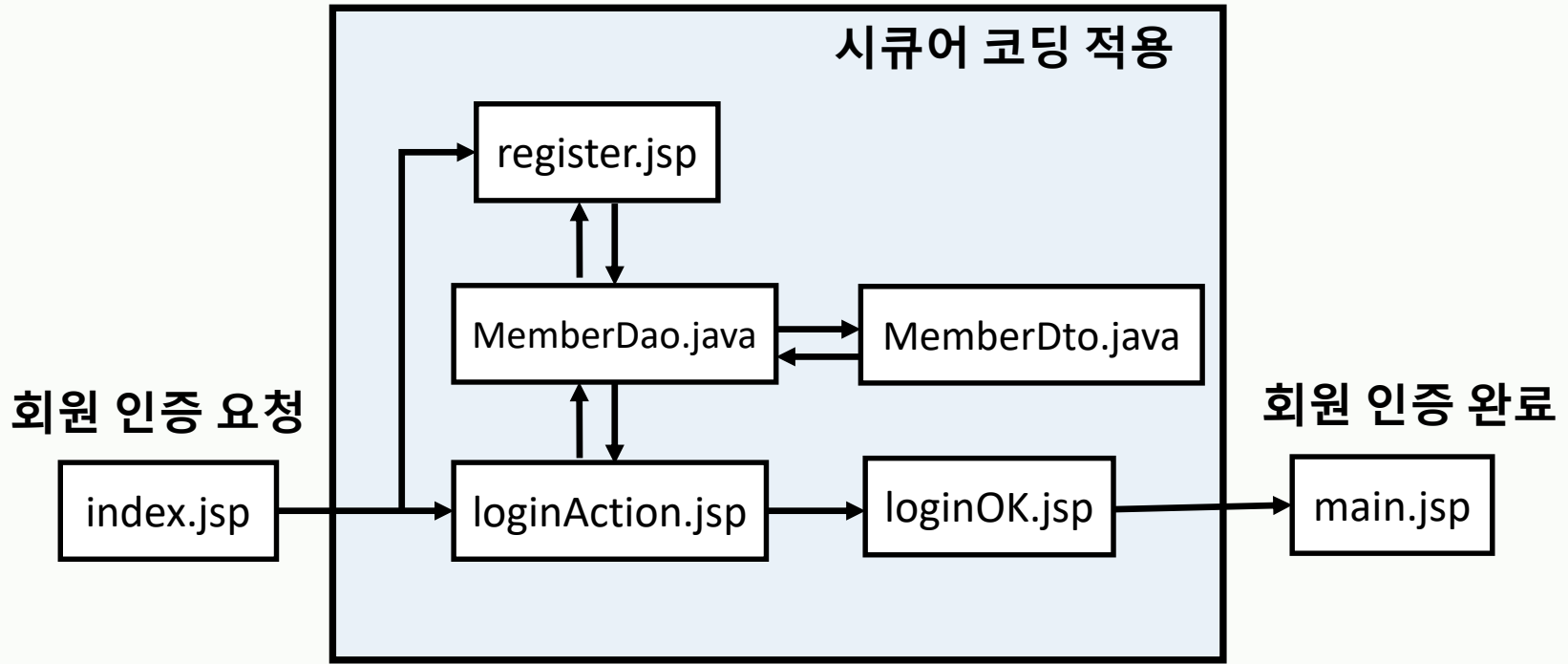
1	입력데이터 검증 및 표현	프로그램 입력값에 대한 검증 누락 또는 부적절한 검증, 데이터의 잘못된 형식지정으로 인해 발생할 수 있는 보안약점	
1	SQL 삽입	[검증되지 않은 외부 입력값이 SQL 쿼리 생성에 사용되어 데이터베이스가 실행될 수 있는 보안약점]	
2	경로 조작 및 자원 삽입	2	보안기능
3	크로스사이트 스크립트	1	적절한 인증 없는 중요기능 허용
4	운영체제 명령어 삽입	2	부적절한 인가
5	위험한 형식 파일 업로드	3	중요한 자원에 대한 잘못된 권한 설정
6	신뢰되지 않는 URL 주소로 자동접속 연결	4	취약한 암호화 알고리즘 사용
7	XQuery 삽입	5	중요정보 평문저장
8	XPath 삽입	6	중요정보 평문전송
9	LDAP 삽입		
10	크로스사이트 요청 위조	3	시간 및 상태
11	HTTP 응답분할	1	경쟁조건: 검사 시점과 사용 시점(TOCTOU)
12	정수형 오버플로우	2	종료되지 않는 반복문 또는 재귀 함수
13	보안기능 결정에 사용 되는 부적절한 입력		
14	메모리 버퍼 오버플로우		
15	포맷 스트링 삽입	4	에러처리
2	보안기능	1	오류 메시지를 통한 정보 노출
	보안기능(인증, 접근제어, 기밀성, 암호화,	2	오류 상황 대응 부재
		3	부적절한 예외 처리
6	캡슐화	중요한 데이터 또는 기능을 불충분하게 캡슐화 하였을 때, 인가되지 않은 사용자에게 데이터 누출이 가능해지는 보안약점	
1	잘못된 세션에 의한 데이터 정보 노출	잘못된 세션에 의해 인가되지 않은 사용자에게 중요정보가 노출될 수 있는 보안약점	
2	제거되지 않고 남은 디버그 코드	디버깅을 위해 작성된 코드를 통해 인가되지 않은 사용자에게 중요정보가 노출될 수 있는 보안약점	
3	시스템 데이터 정보노출	사용자가 볼 수 있는 오류 메시지나 스택 정보에 시스템 내부 데이터나 디버깅 관련 정보가 공개되는 보안약점	
4	Public 메소드부터 반환된 Private 배열	Private로 선언된 배열을 Public으로 선언된 메소드를 통해 반환(return)하면, 그 배열의 레퍼런스가 외부에 공개되어 외부에서 배열이 수정될 수 있는 보안약점	
5	Private 배열에 Public 데이터 할당	Public으로 선언된 데이터 또는 메소드의 인자가 Private로 선언된 배열에 저장되면, Private 배열을 외부에서 접근할 수 있게 되는 보안약점	
7	API 오용	의도된 사용에 반하는 방법으로 API를 사용하거나, 보안에 취약한 API를 사용하여 발생할 수 있는 보안약점	
1	DNS lookup에 의존한 보안결정	DNS는 공격자에 의해 DNS 스푸핑 공격 등이 가능하므로 보안결정을 DNS 이름에 의존할 경우, 보안결정 등이 노출되는 보안약점	
2	취약한 API 사용	취약하다고 알려진 함수를 사용함으로써 예기치 않은 보안위협에 노출될 수 있는 보안약점	

모의 침투 구성

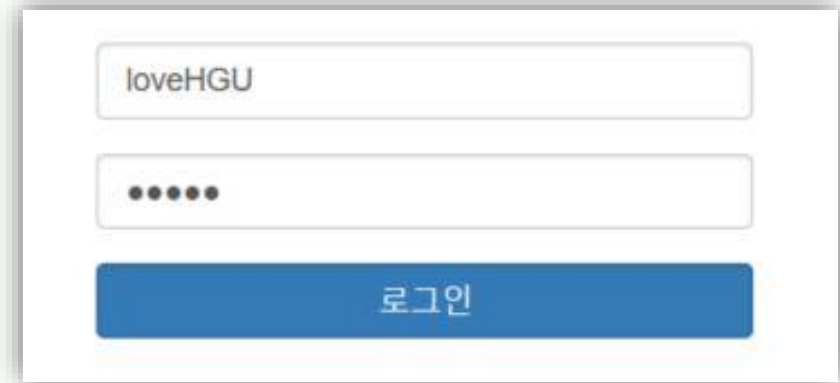
웹 취약점 공격

취약점 코드 발견

시큐어 코드 적용



SQL Injection



A login form with a text input field containing 'loveHGU', a password input field with six dots, and a blue button labeled '로그인' (Login).

SELECT id, passwd FROM users

WHERE ID = 'LoveHGU' AND pwd = 123

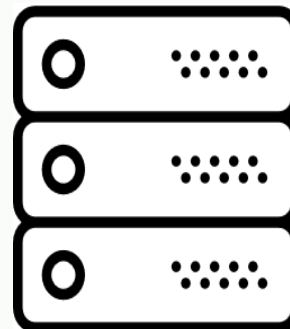


User

ID = LoveHGU
PW = 123



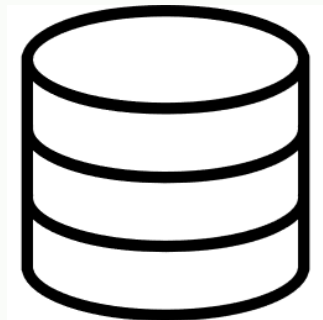
인증 완료



Server



인증 완료



DB

SQL Injection



loveHGU' OR '1'='1

.....

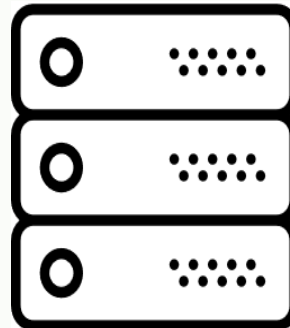
로그인

ID = 'LoveHGU' OR '1'='1'--

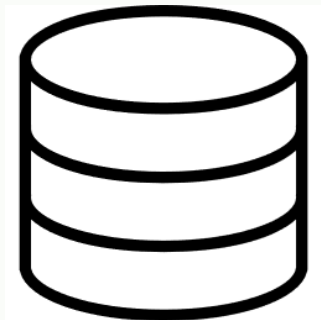
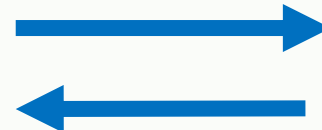
SELECT id, passwd FROM users
WHERE ID = 'LoveHGU' OR '1'='1'--



User



Server



DB

취약한 코드 (MemberDao.java)

```
10 String userid=request.getParameter("userid");
3: String password=request.getParameter("password");
10
4: ....

Statement stmt = conn.createStatement();

11 ResultSet rs = stmt.executeQuery("SELECT count(*) FROM member
2: WHERE userid='"+userid+"'AND password='"+password+"'");

11
```

```
4: SELECT count(*) FROM member
WHERE userid=ID = `LoveHGU` OR `1 = 1`-- AND password=`11111`
```

└─ 비밀번호 검증 주석

시큐어 코딩 적용 (MemberDao.java)

```
10 String query = "SELECT userPassword FROM user WHERE userID = ?";
6:
try {
10     connection = getConnection();
8:     pstmt = connection.prepareStatement(query);
10     pstmt.setString(1, id);
9:     set = pstmt.executeQuery();
11
0:
11     if(set.next()) {
1:         dbPw = set.getString("userPassword");
11         if(dbPw.equals(pw)) {
2:             ri = MemberDao.MEMBER_LOGIN_SUCCESS;      // 로그인 성공
11         } else {
3:             ri = MemberDao.MEMBER_LOGIN_PW_NO_GOOD;    // 패스워드 틀림
11         }
5:
11
6:
11
7:
```

중요정보 평문저장

```
sqlmap -u  
"http://192.168.0.11/dvwa/vulnerabilities/sqli_blind/?id=&Submit=Submit#"  
-p "id" --dbs --cookie="security=low;  
PHPSESSID=532241126619564c18e1b1c628e09aa9" --tables -D "dvwa" --  
columns -T "users" -C "users,password" --dump
```

```
Database: dvwa  
Table: users  
[5 entries]  
+-----+-----+-----+-----+-----+-----+  
| user_id | user      | avatar                                     | last_name | first_name | password |  
+-----+-----+-----+-----+-----+-----+  
| 1       | admin     | http://localhost/dvwa/hackable/users/admin.jpg | admin     | admin     | 5f4dcc3  
b5aa765d61d8327deb882cf99 (password) |  
| 2       | gordonb   | http://localhost/dvwa/hackable/users/gordonb.jpg | Brown     | Gordon    | e99a18c  
428cb38d5f260853678922e03 (abc123) |  
| 3       | 1337      | http://localhost/dvwa/hackable/users/1337.jpg   | Me         | Hack      | 8d3533d  
75ae2c3966d7e0d4fcc69216b (charley) |  
| 4       | pablo     | http://localhost/dvwa/hackable/users/pablo.jpg   | Picasso   | Pablo     | 0d107d0  
9f5bbe40cade3de5c71e9e9b7 (letmein) |  
| 5       | smithy    | http://localhost/dvwa/hackable/users/smithy.jpg  | Smith     | Bob       | 5f4dcc3  
b5aa765d61d8327deb882cf99 (password) |  
+-----+-----+-----+-----+-----+-----+
```

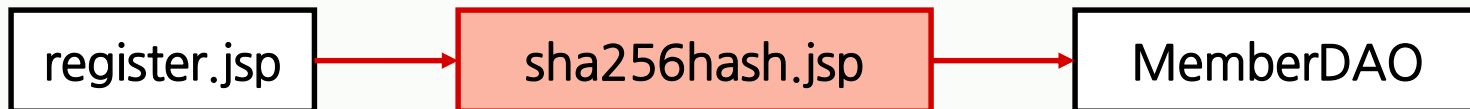
취약한 코드 (MemberDao.java)

```
35: String query = "INSERT INTO users VALUES (?, ?, ?, ?, ?)";
36:
37: try {
38:     connection = getConnection();
39:     pstmt = connection.prepareStatement(query);
40:     pstmt.setString(1, dto.getUserID());
41:     pstmt.setString(2, dto.getUserName());
42:     pstmt.setString(3, dto.getUserPassword());
43:     pstmt.setString(4, dto.getStudentNumber());
44:     pstmt.setString(5, dto.getPhoneNumber());
45:     pstmt.executeUpdate();
46:     ri = MemberDao.MEMBER_JOIN_SUCCESS;
```

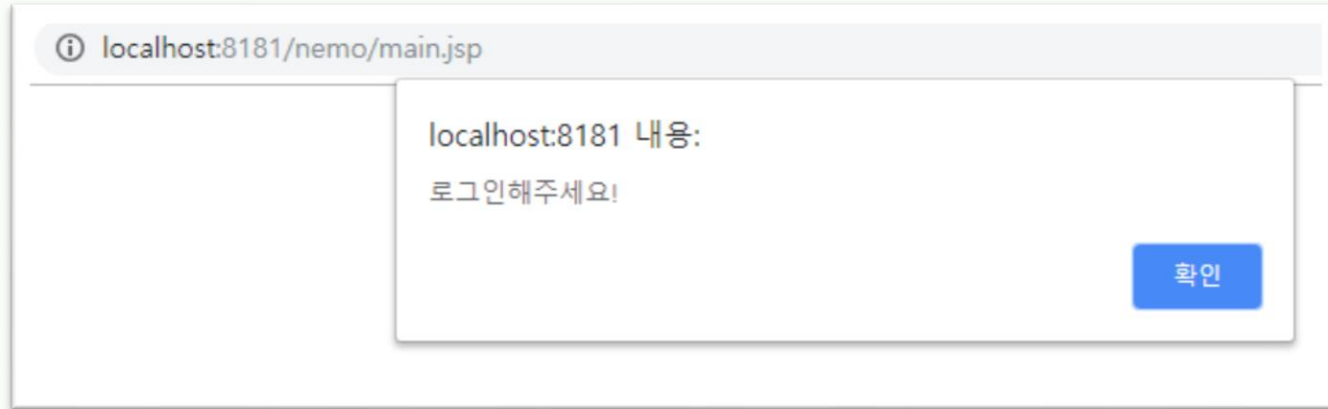
개선 코드 (sha256hash.jsp)

```
01: <%@page import="sun.misc.BASE64Encoder"%>
02: <%@page import="java.io.*"%>
03: <%@page import="KISA.SHA256"%>
04: <%
05:     SeedCBC s = new SeedCBC();
06:     String retMsg = s.LoadConfig(KEY_PATH);
07:     if(retMsg.equals("OK") == false){
08:         out.println(retMsg);
09:     }else{
10:         String sPlainText = s.Encryption(sPlainText.getBytes());

    ...
```



부적절한 인가



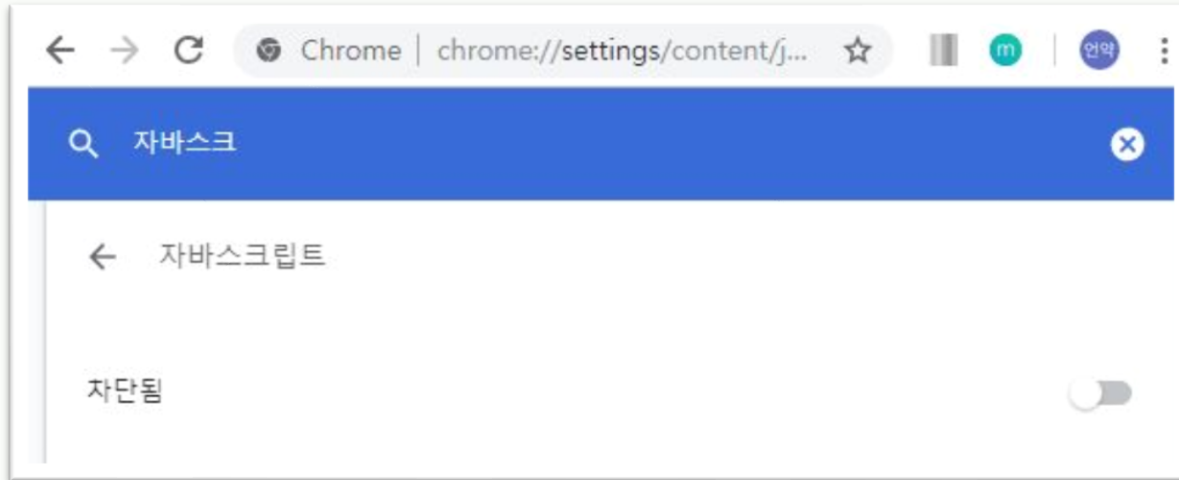
123
123
123

pw: 123
name: 123
id: 123

sessionID : 9A32126AA2AE55C52C4F0F13E5F358B9
sessionInter : 1800

session valid

부적절한 인가



```
null
null
null
*****
*****
sessionID : A929F281844F5FBF499420410CCB8671
sessionInter : 1800
*****
session valid
```

서울에서 출발하는 KTX 목록입니다.

ID	출발 도시	도착 도시	출발 역	도착 역	출발 시간	도착 시간	출발 날짜	방장 PK	방장 이름	참가 하기
29	서울특별시	경기도	서울	수원	101700	104700	20181127	null	null	참가하기

취약한 코드 (main.java)

```
44: String loginID = (String)session.getAttribute("id");
45:
46: if(session.getAttribute("id") == null){
47:     script.println("<script>");
48:     script.println("alert('로그인해주세요!')");
49:     script.println("location.href='index.jsp'");
50:     script.println("</script>");
51: }
```

개선 코드 (main.jsp)

```
4 String loginID = (String)session.getAttribute("id");
4:
4 if(session.getAttribute("id") == null){
5:     script.println("<script>");
4     script.println("alert('로그인해주세요!')");
6:     script.println("</script>");
4     response.sendRedirect("index.jsp");
7: }
4
8:
4
9:
5
0:
5
```

```

<%
String userID = null;
if (session.getAttribute("userID") != null) {
    userID = (String)session.getAttribute("userID");
}
if (userID == null) {
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('로그인을 하세요.')");
    script.println("location.href = 'login.jsp'");
    script.println("</script>");
}

```

출처 : <https://github.com/LEEHANI/JSP-BBS/blob/master/WebContent/update.jsp>

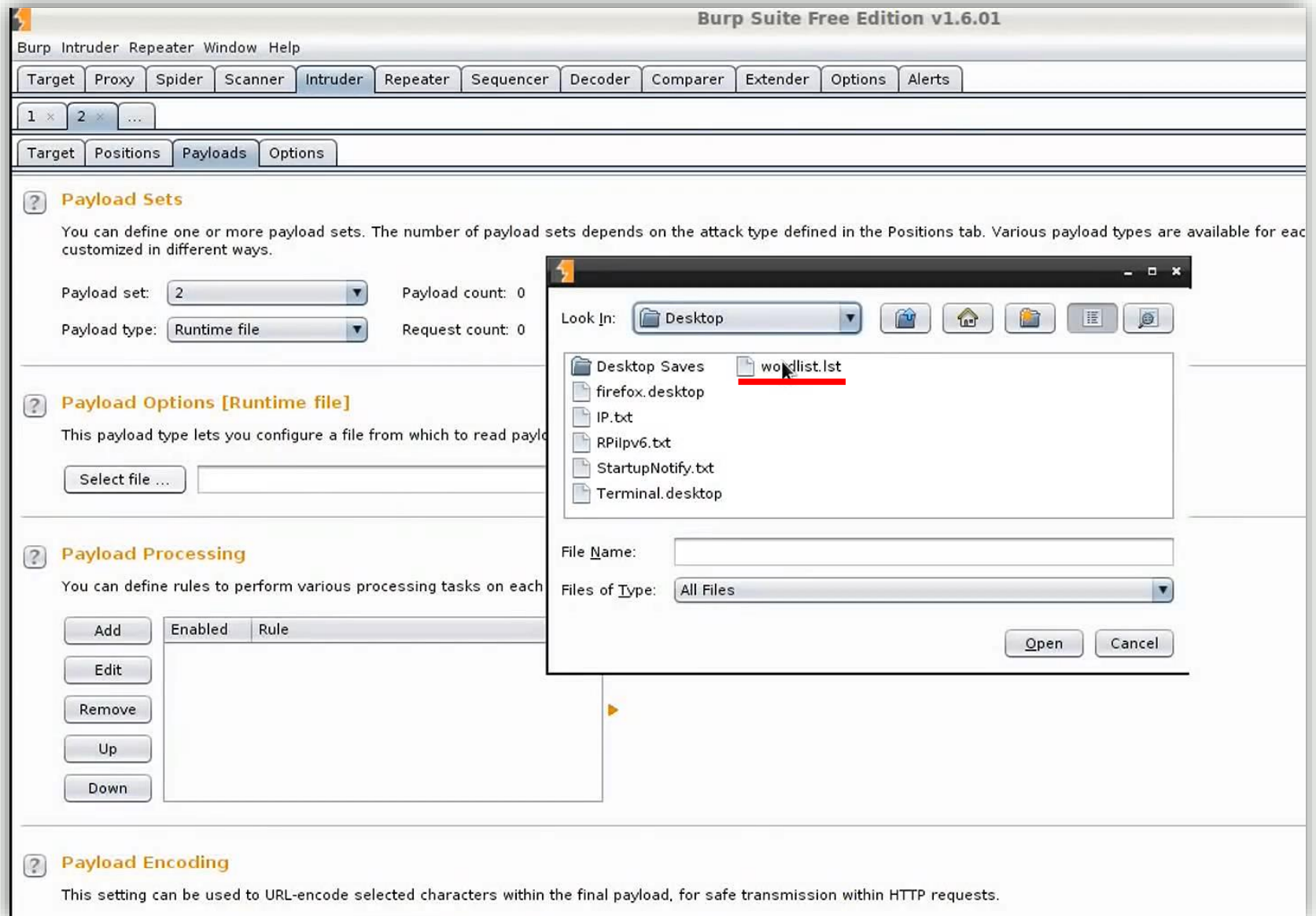
```

<%
String id = request.getParameter("id");
String passwd = request.getParameter("passwd");
if(check){
    session.setAttribute("id", id);
    response.sendRedirect("main.jsp");
}else{%>
<script>
    alert("Login Fail");
    history.go(-1);
</script>
<}%>

```

출처 : <http://hyeonstorage.tistory.com/125>

취약한 비밀번호 허용



<> Code

Issues 1

Pull requests 1

Projects 0

Wiki

Insights

Bruteforce database <http://duyetdev.github.io/bruteforce-...>

duyetdev

password-dictionaries

seclists

password

brute-force

bruteforce

brute-force-attacks

68 commits

2 branches

0 releases

8 contributors

MIT

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



duyetdev Merge pull request #10 from danivijay/add-forced-browsing ...

Latest commit 719017b Dec 9, 2017

forced-browsing	add forced browsing wordlists	Dec 5, 2017
1000000-password-seclists.txt	Remove 1000000_password_seclists.txt -> 1000000-password-seclists.txt	Oct 4, 2015
2151220-passwords.txt	Remove comment header of 2151220-passwords.txt	Oct 4, 2015
38650-password-sktorrent.txt	add sktorrent passwords	Nov 2, 2016
38650-username-sktorrent.txt	add sktorrent usernames	Nov 3, 2016
7-more-passwords.txt	Move mangle.lst -> 7-more-passwords.txt	Oct 4, 2015
8-more-passwords.txt	Change wordlist.lst > 8-more-passwords.txt	Oct 4, 2015
LICENSE	Initial commit	Oct 4, 2015
README.md	add forced browsing wordlists	Dec 5, 2017
bitcoin-brainwallet.lst	Add bitcoin-brainwallet with 394748 lines	Oct 4, 2015

개인정보의 안전성 확보조치 기준 (2011.9.30.제정, 행정안전부 고시 제2011-43호)

제5조(비밀번호 관리) 개인정보처리자는 개인정보취급자 또는 정보주체가 안전한 비밀번호를 설정하여 이행할 수 있도록 비밀번호 작성규칙을 수립하여 적용하여야 한다

출처 : 국가 법령 보호센터 <http://law.go.kr/>

❖ 세가지 종류 이상의 문자구성으로 8자리 이상의 길이로 구성된 문자열

또는

❖ 두가지 종류 이상의 문자구성으로 10자리 이상의 길이로 구성된 문자열

※ 문자종류는 알파벳 대문자와 소문자, 특수문자, 숫자의 4가지임

출처 : KISA 패스워드 선택 및 이용 가이드 <https://seed.kisa.or.kr/iwt/ko/sup/EgovSafePwdLb.do>

취약한 코드 (joinAction.jsp)

```
21: MemberDao dao = MemberDao.getInstance();
22: PrintWriter script = response.getWriter();
23:
24: if (user.getUserID() == null ||
25:     user.getUserPassword() == null ||
26:     user.getUserName() == null ||
27:     user.getStudentNumber() == null ||
28:     user.getPhoneNumber() == null )
29: {
    ...
    ...
33: int ri = dao.insertMember(user);
34: if(ri == MemberDao.MEMBER_JOIN_SUCCESS)
```

개선 코드 (loginAction.jsp)

```
21: MemberDao dao = MemberDao.getInstance();
22: PrintWriter script = response.getWriter();
23:
24: bVaild = checkRegexp(user.getUserPassword(),
25: /^(?=.*(?:^.{10,20}$)(?=.*\d)?=.*[a-zA-Z])
26: (?=.*[!@#$%^&+=]).*$/ ,
27: "Password field only allow:10-20 character and special
28: character");
29:
30: if (user.getUserID()          == null    ||
31:     user.getUserPassword()    == null    ||
32:     user.getUserName()        == null    ||
33:     user.getStudentNumber()    == null    ||
34:     user.getPhoneNumber()      == null    || bVaild )
35: {
```

End.