

□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

# **File Upload**

**2025년 5월 13일**

학번 : 32231594

이름 : 박기쁨

## 1. 워게임 5 번 문항

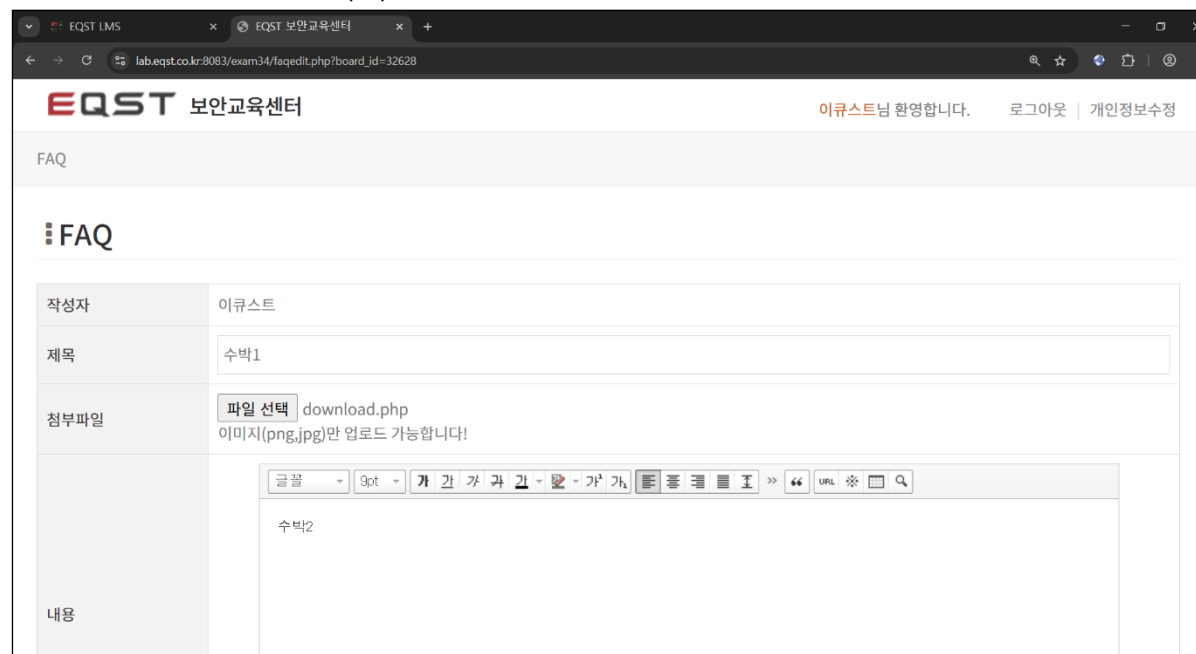
### <과정 설명>

File Upload 취약점 공격은 파일을 업로드하는 과정에서의 취약점을 이용하여 원하는 정보를 탈취하는 행위를 말한다. 본 실습에서는 File Upload 취약점을 이용해 .php 확장자의 파일을 업로드 해보고자 한다.

### Step 1. php 파일 업로드

#### Step 1-1. php 파일 업로드

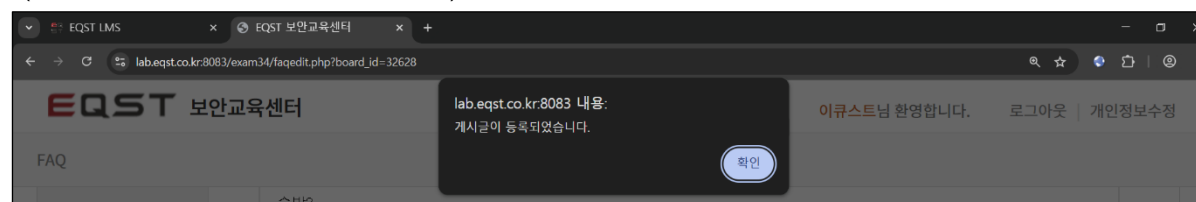
: 목표 페이지의 게시판에 php 파일을 업로드 한다.



The screenshot shows a web browser window with the URL `lab.eqst.co.kr:8083/exam34/faqedit.php?board_id=32628`. The page title is "EQST 보안교육센터" (EQST Security Education Center). The main content area is titled "FAQ" and contains a form for editing a post. The form fields are:

- 작성자 (Author): 이큐스트 (iquest)
- 제목 (Title): 수박1
- 첨부파일 (Attachments): A button labeled "파일 선택" (File Select) is next to the text "download.php". Below it, a message says "이미지(png,jpg)만 업로드 가능합니다!" (Only images (png, jpg) can be uploaded!).
- 내용 (Content): A text area containing the text "수박2".

(게시물이 정상적으로 업로드 된다.)

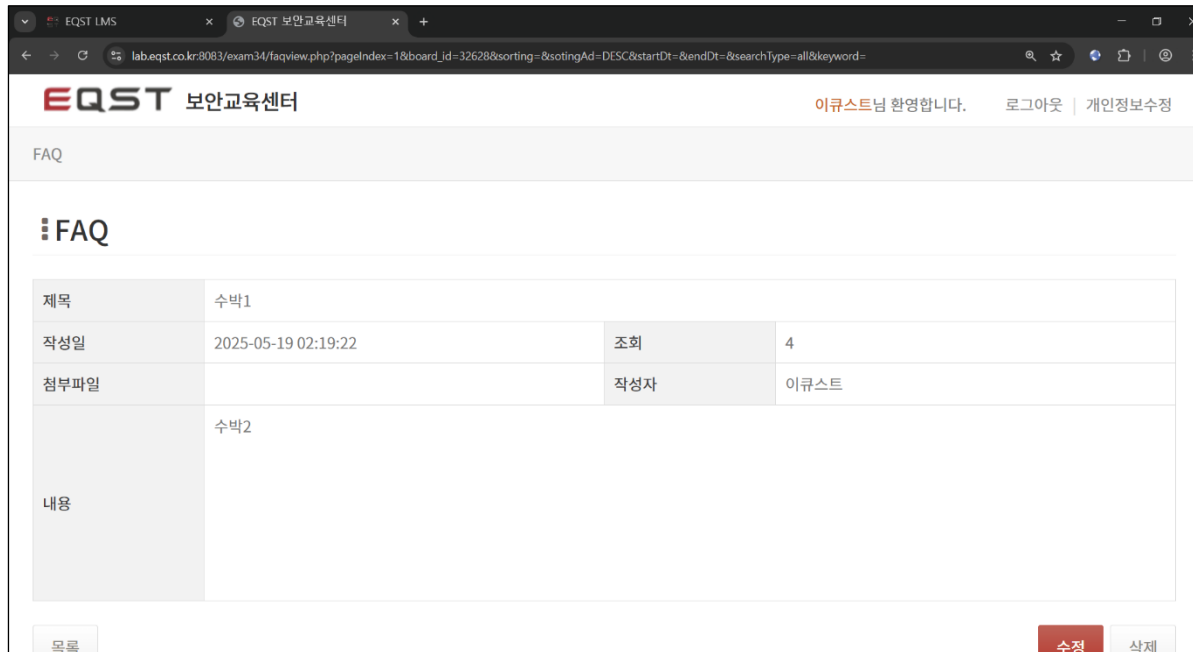


The screenshot shows the same web browser window as before, but with a confirmation message displayed in the center. The message box contains the text:

lab.eqst.co.kr:8083 내용:  
게시글이 등록되었습니다.

Below the message box is a blue button labeled "확인" (Confirm). The background of the page is dimmed.

: 게시물은 정상적으로 업로드 되었지만, php 파일은 삭제된 것을 알 수 있다.



EQST 보안교육센터

이큐스트님 환영합니다. 로그아웃 | 개인정보수정

FAQ

FAQ

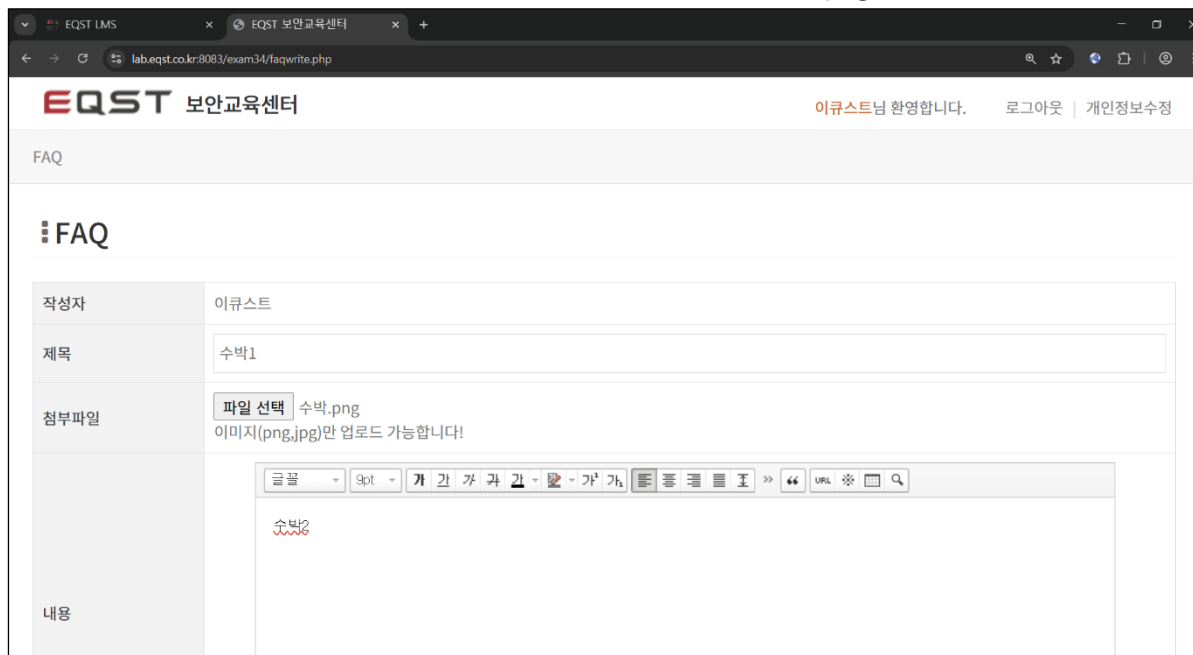
제목	수박1	조회	4
작성일	2025-05-19 02:19:22	작성자	이큐스트
첨부파일			
내용	수박2		

목록 수정 삭제

## Step 2. png 파일 업로드

### Step 2-1. png 파일 업로드

: 목표 페이지의 게시판 파일 업로드 방식 등을 확인하기 위하여 png 파일을 업로드 해본다.



EQST 보안교육센터

이큐스트님 환영합니다. 로그아웃 | 개인정보수정

FAQ

FAQ

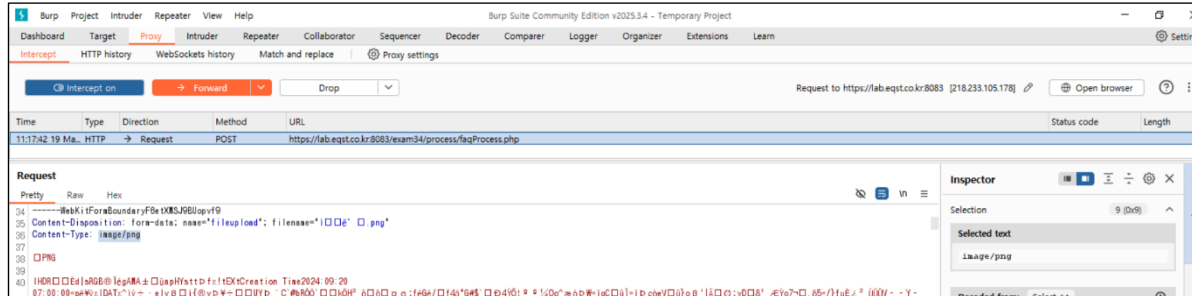
작성자	이큐스트
제목	수박1
첨부파일	파일 선택 수박.png 이미지(png,jpg)만 업로드 가능합니다!
내용	수박2

목록 수정 삭제

## Step 2-2. intercept 및 content-Type 확인

: 게시물을 저장 과정을 intercept 하여 패킷의 Content-Type 을 확인한다.

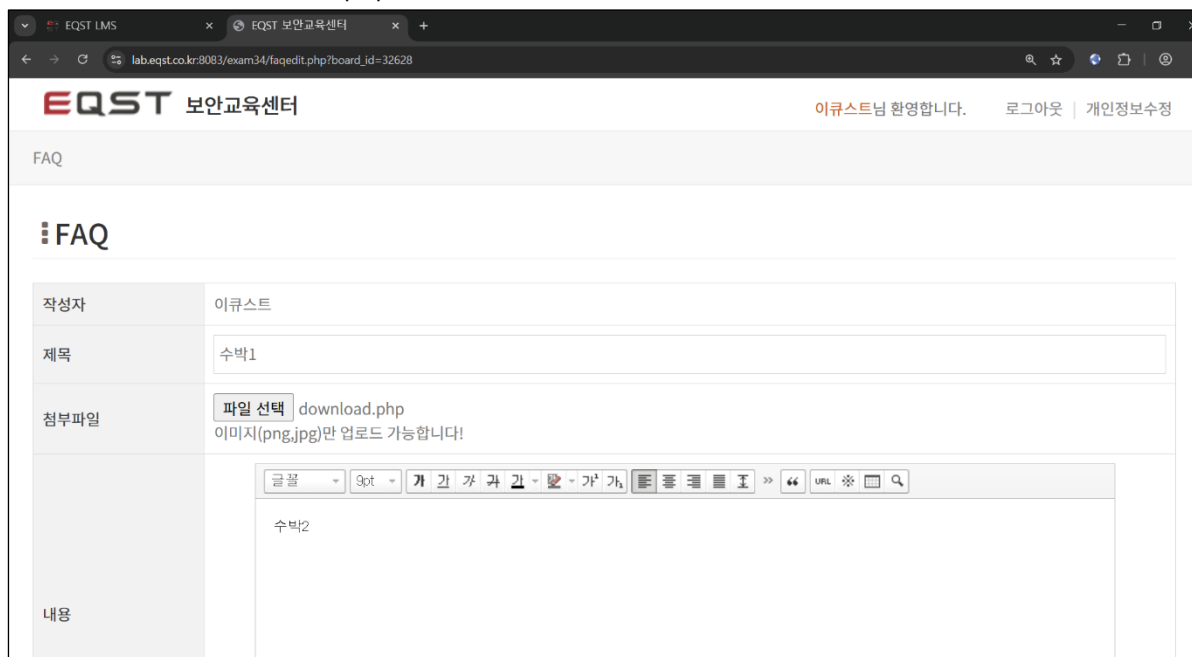
: 수박.png 파일의 Content-Type 은 image/png 이다.



## Step 3. php 파일 업로드 (재시도)

### Step 3-1. php 파일 업로드 (재시도)

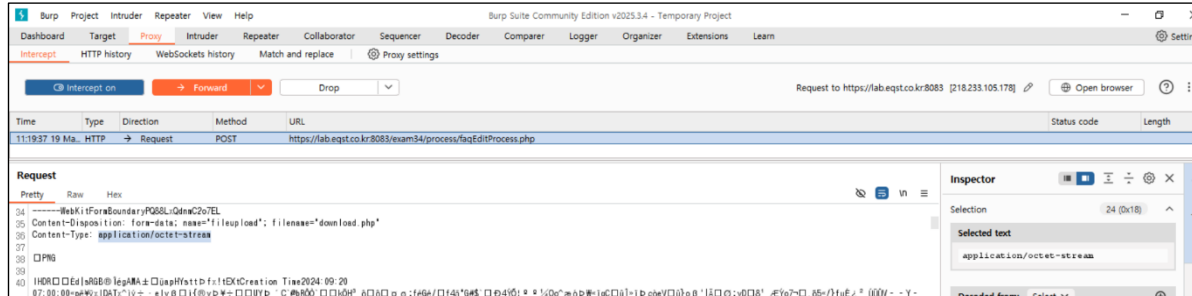
: 목표 페이지의 게시판에 php 파일을 업로드 한다.



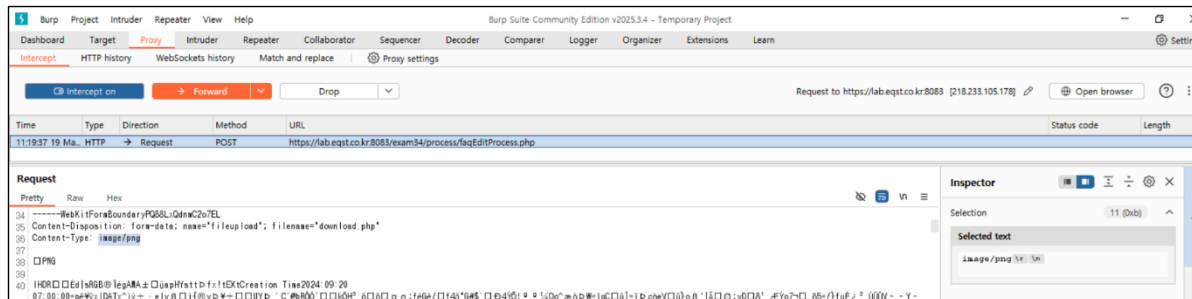
### Step 3-2. intercept 및 content-Type 확인

: 게시물을 저장 과정을 intercept 하여 패킷의 Content-Type 을 확인한다.

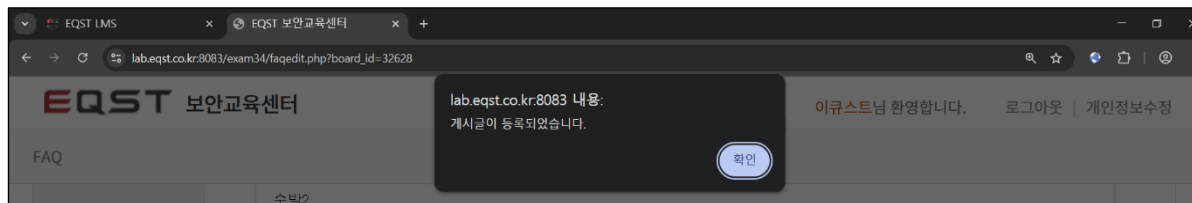
: download.php 파일의 Content-Type 은 application/octet-stream 이다.



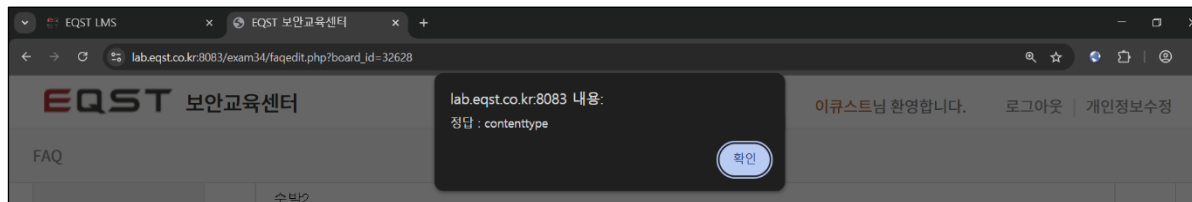
: download.php 파일의 Content-Type 을 image/png 로 변경한다.



(게시물이 정상적으로 업로드 된다.)



: php 파일이 성공적으로 업로드 되어 정답을 획득할 수 있다. (정답: contenttype)



## 2. 위게임 6 번 문항

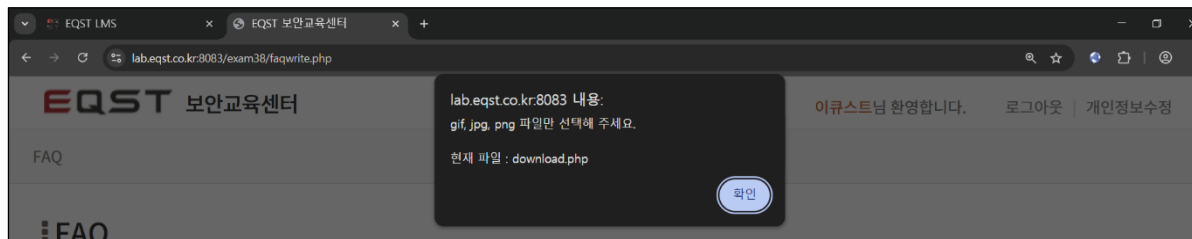
### <과정 설명>

#### Step 1. php 파일 업로드

##### Step 1-1. php 파일 업로드

: 목표 페이지의 게시판에 php 파일을 업로드 한다.

: 첨부파일 선택 과정에서 .php 확장자 파일 업로드가 필터링 되는 것을 알 수 있다.

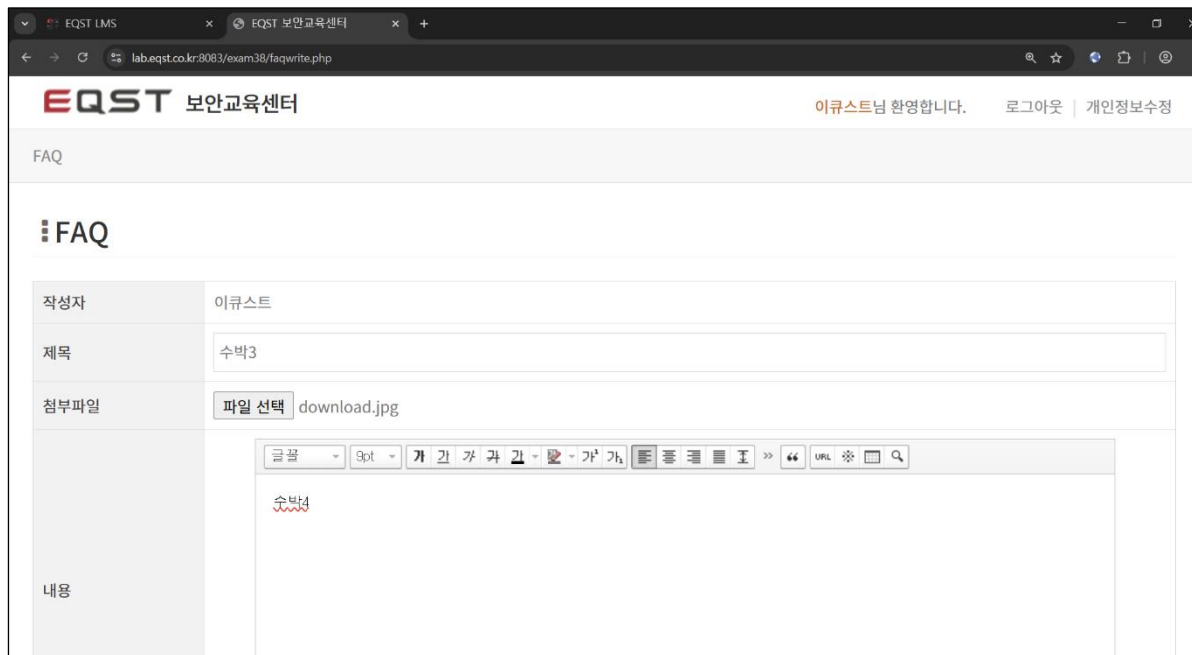


#### Step 2. php 파일 확장자 변경

##### Step 2-1. php 파일 확장자 변경

: 첨부파일 선택 과정에서의 확장자 필터링을 우회하기 위해, php 파일의 확장자를 jpg 로 변경하여 업로드한다.

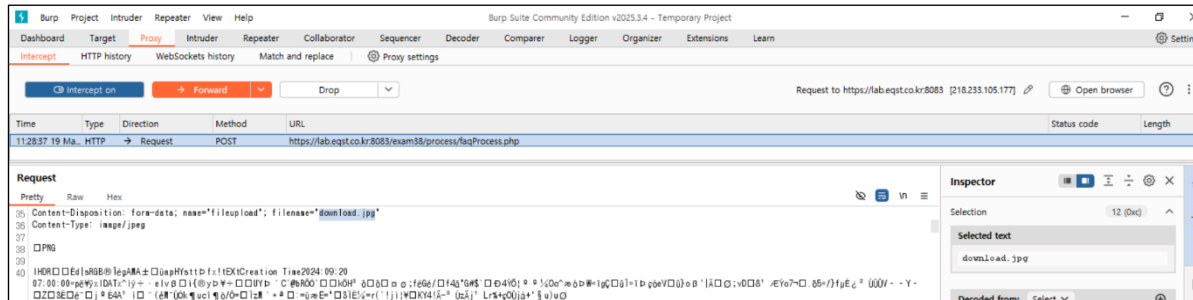
: 첨부파일 선택 과정에서 필터링 되지 않는다.



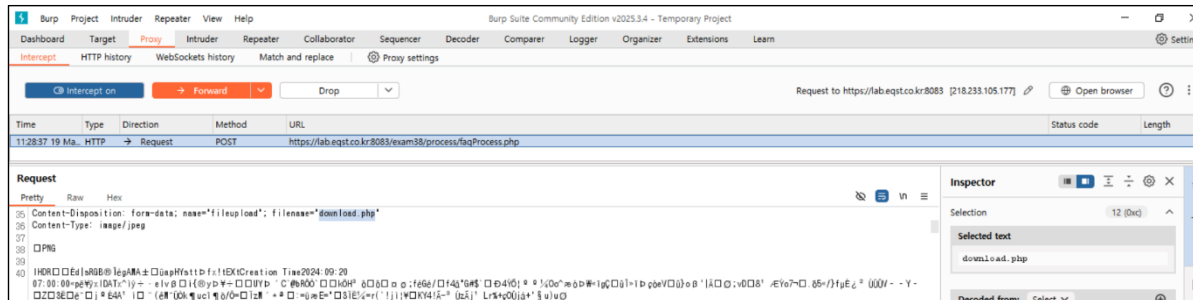
### Step 3. php 파일 확장자 원상복구 (재시도)

#### Step 3-1. php 파일 확장자 원상복구

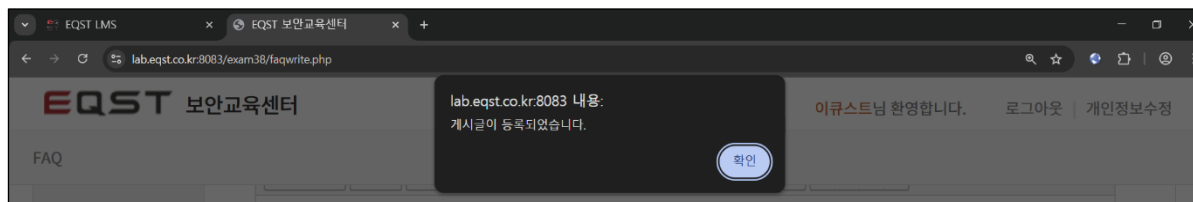
: 게시물을 저장 과정을 intercept 하여 확장자를 php 로 원상복구한다.



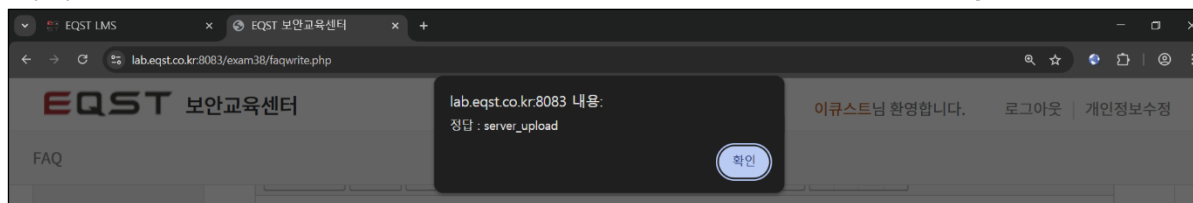
(.jpg 를 .php 로 변경한다.)



(게시물이 정상적으로 업로드 된다.)



: php 파일이 성공적으로 업로드 되어 정답을 획득할 수 있다. (정답: server\_upload)



### 3. 워게임 7 번 문항

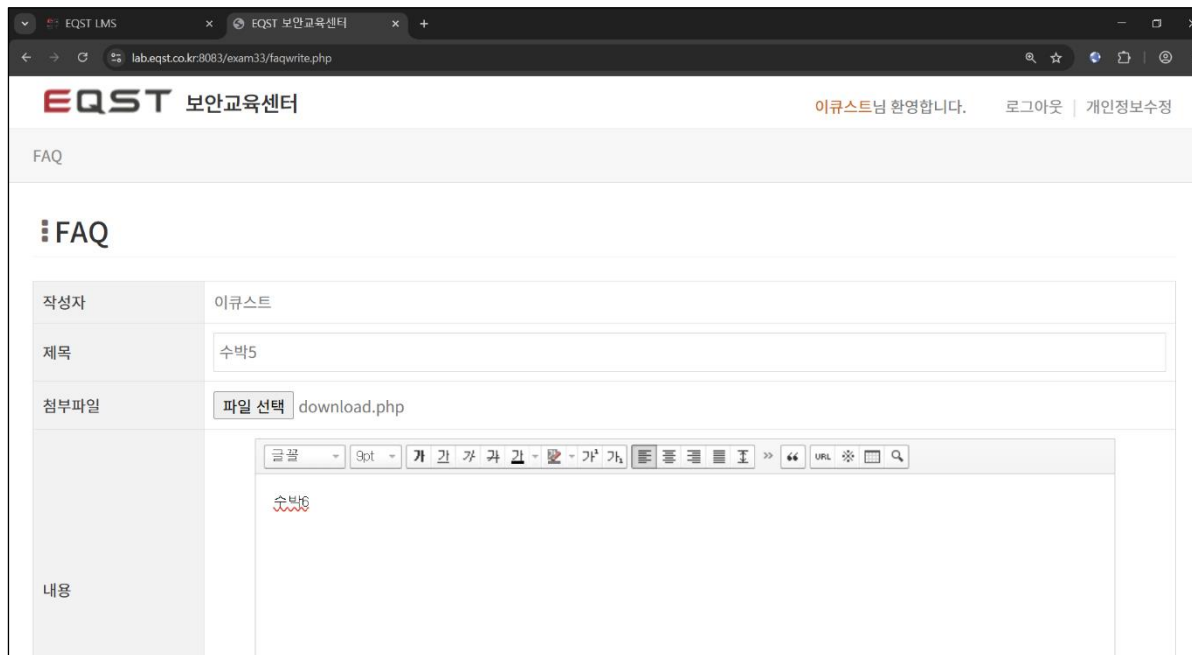
#### <과정 설명>

##### Step 1. php 파일 업로드

###### Step 1-1. php 파일 업로드

: 목표 페이지의 게시판에 php 파일을 업로드 한다.

: 첨부파일 선택 과정에서 .php 확장자 파일 업로드가 필터링 되는 것을 알 수 있다.



EQST 보안교육센터

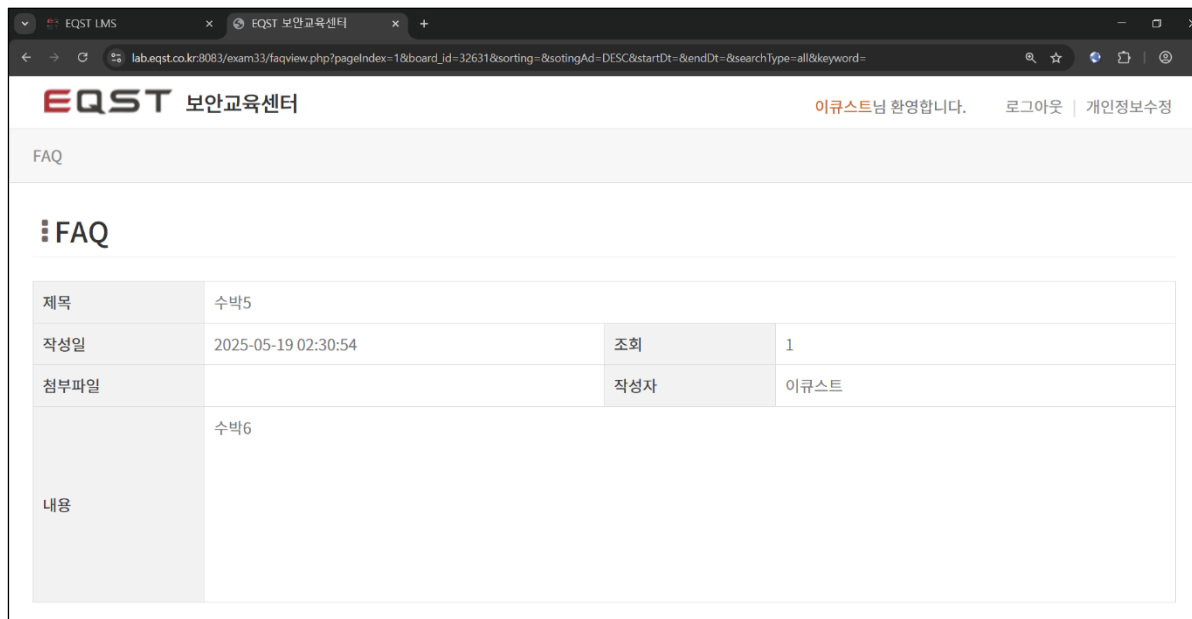
이큐스트님 환영합니다. | 로그아웃 | 개인정보수정

FAQ

FAQ

작성자	이큐스트
제목	수박5
첨부파일	파일 선택 download.php
내용	<p>수박6</p>

: 게시물은 정상적으로 업로드 되었지만, php 파일은 삭제된 것을 알 수 있다.



EQST 보안교육센터

이큐스트님 환영합니다. | 로그아웃 | 개인정보수정

FAQ

FAQ

제목	수박5		
작성일	2025-05-19 02:30:54	조회	1
첨부파일		작성자	이큐스트
내용	수박6		

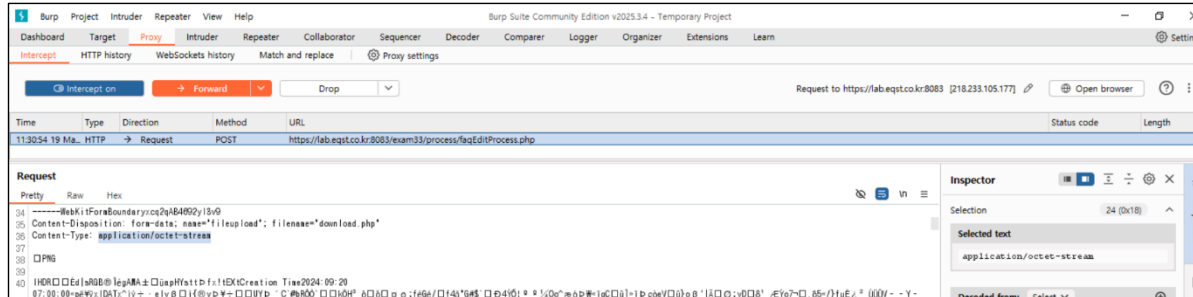


## Step 2. php 파일 업로드 및 Content-Type 변경

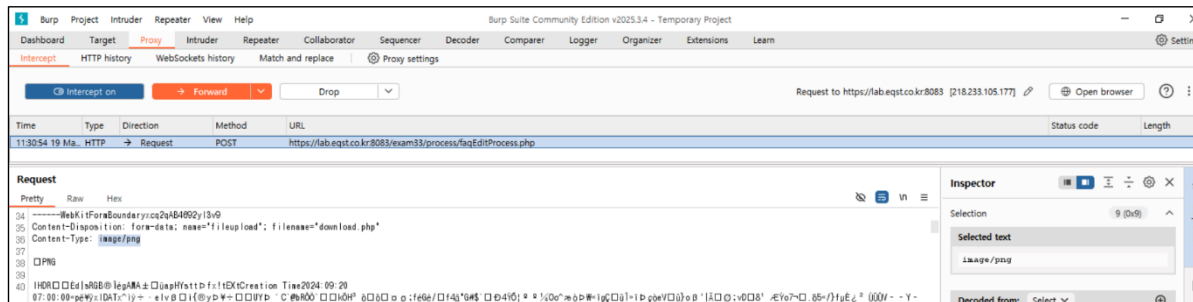
### Step 2-1. intercept 및 content-Type 확인

: 게시물을 저장 과정을 intercept 하여 패킷의 Content-Type 을 확인한다.

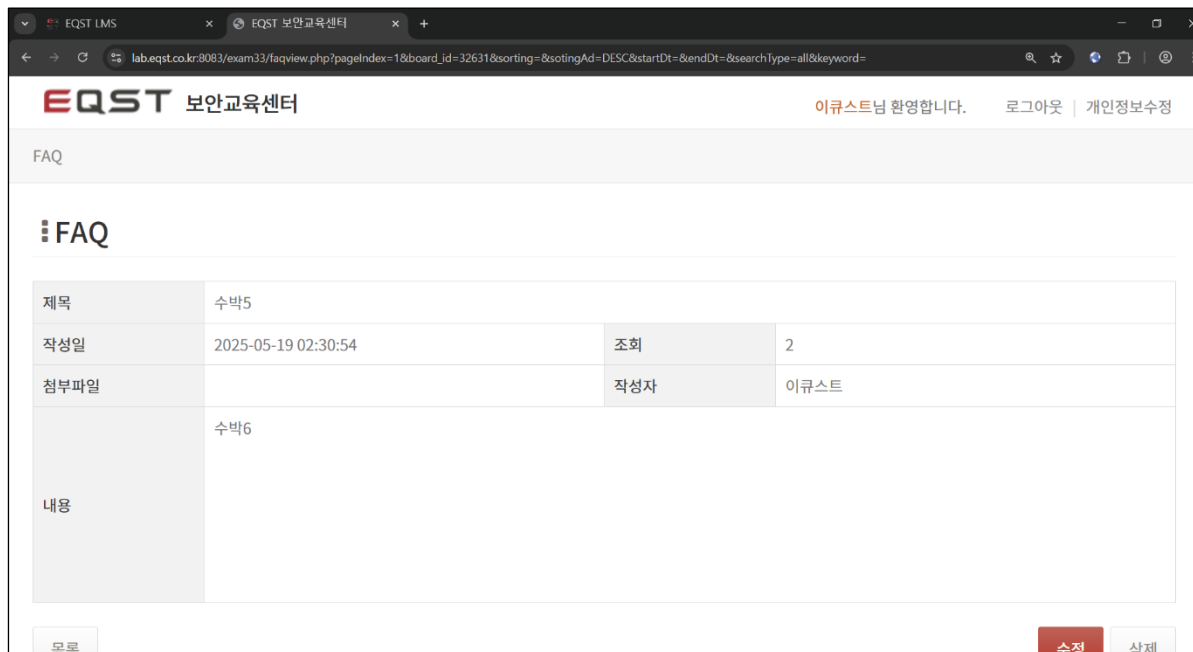
: download.php 파일의 Content-Type 은 application/octet-stream 이다.



: download.php 파일의 Content-Type 을 image/png 로 변경한다.



: 여전히 게시물은 정상적으로 업로드 되었지만, php 파일은 삭제된 것을 알 수 있다.





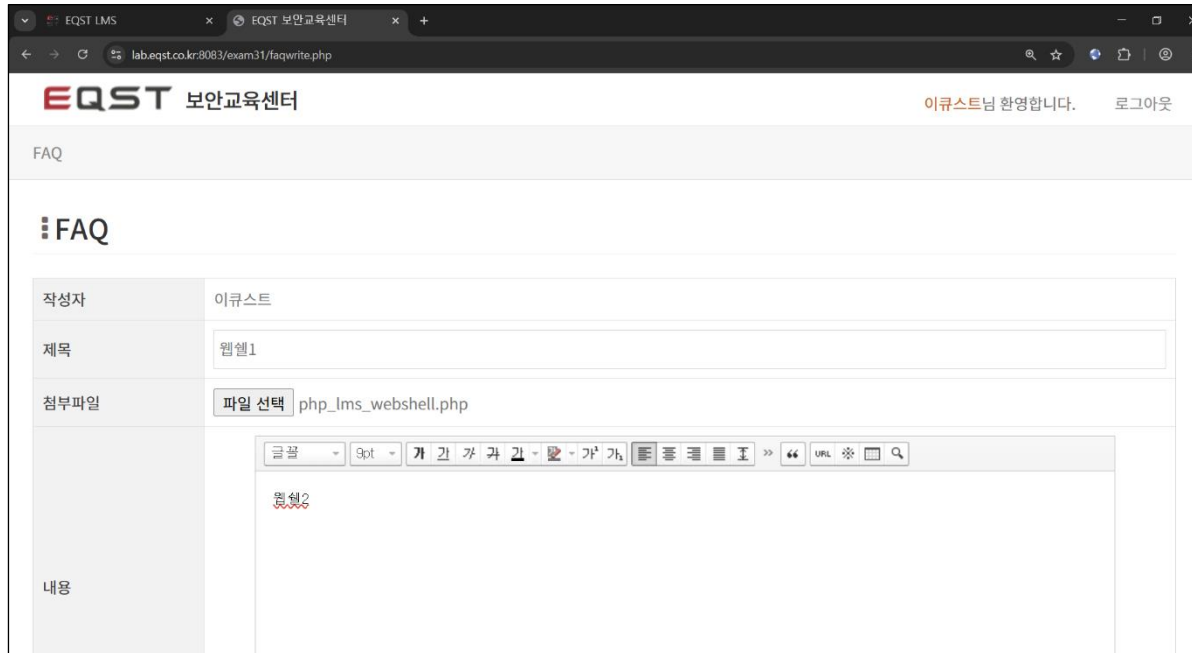
#### 4. 워게임 9 번 문항

##### <과정 설명>

##### Step 1. 웹쉘 업로드

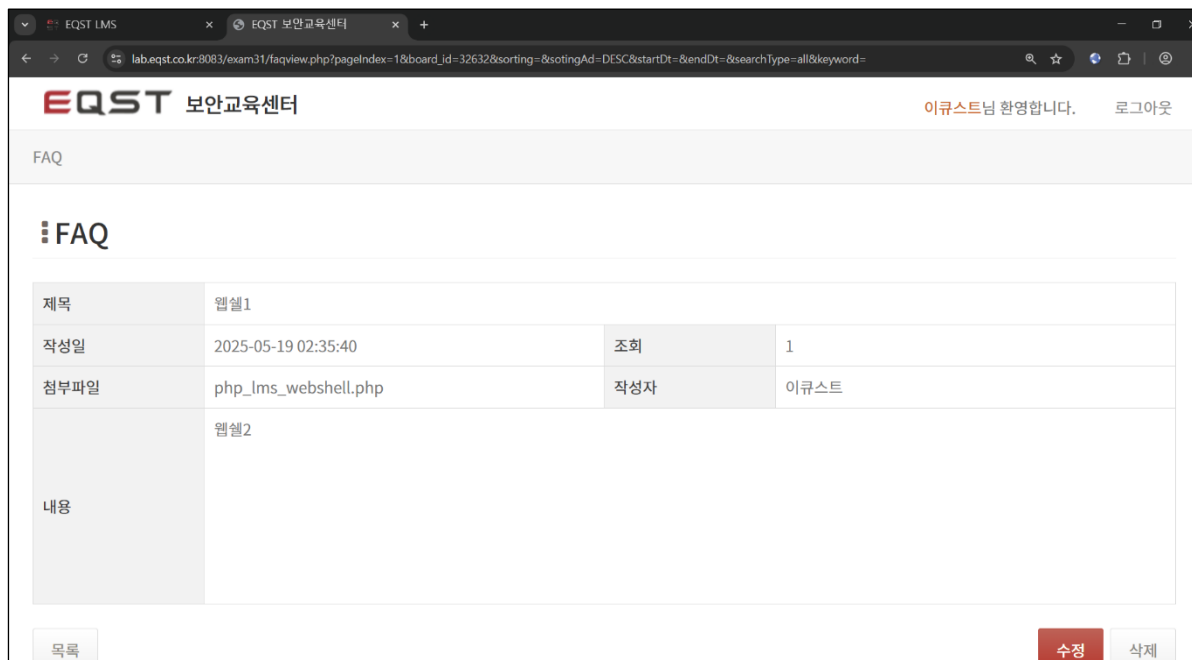
##### Step 1-1. 웹쉘 업로드

: 목표 페이지의 게시판에 웹쉘을 업로드 한다.



The screenshot shows the EQST LMS interface. The top navigation bar includes the EQST logo and '보안교육센터' (Security Education Center). A user is logged in as '이큐스트님' (iquest). The main content area is titled 'FAQ'. Below the title, there is a form for creating a new FAQ entry. The form fields are: '작성자' (Author) set to '이큐스트', '제목' (Title) set to '웹쉘1', and '첨부파일' (Attachment) with a file named 'php\_lms\_webshell.php' selected. The '내용' (Content) field is empty, showing a rich text editor toolbar with options like bold, italic, and link.

(게시물이 정상적으로 업로드 된다.)



The screenshot shows the EQST LMS interface after the FAQ entry has been uploaded. The 'FAQ' section now displays a list of entries. The first entry is: '제목' (Title): 웹쉘1, '작성일' (Date): 2025-05-19 02:35:40, '조회' (Views): 1, '첨부파일' (Attachment): php\_lms\_webshell.php, '작성자' (Author): 이큐스트. Below this entry, there is a '내용' (Content) field showing '웹쉘2'. At the bottom of the page, there are buttons for '목록' (List), '수정' (Edit), and '삭제' (Delete).

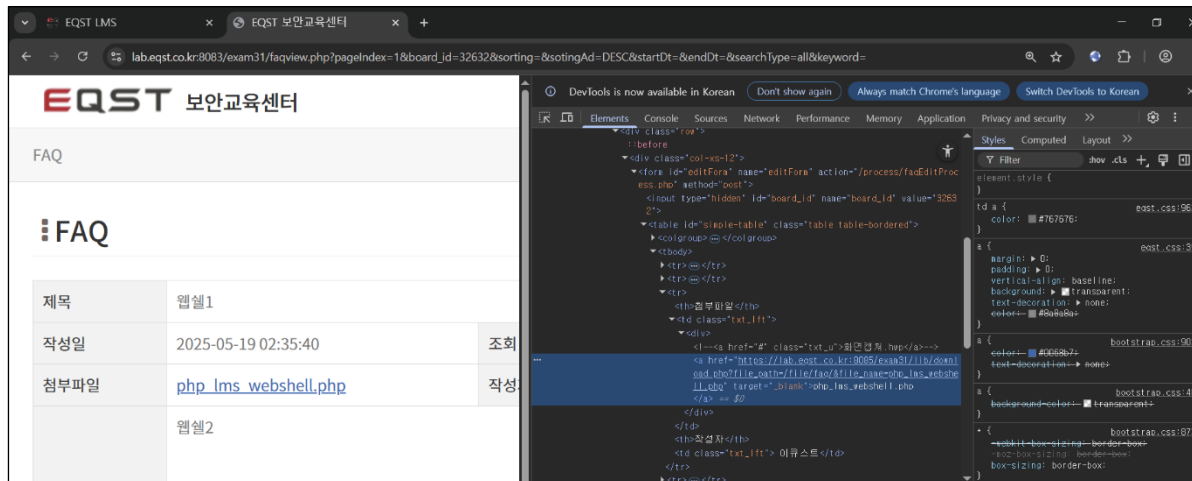
## Step 2. 웹쉘 경로 찾기 및 실행하기

### Step 2-1. 웹쉘 경로 찾기

: 개발자도구를 사용하여 첨부파일 관련 코드를 확인한다.

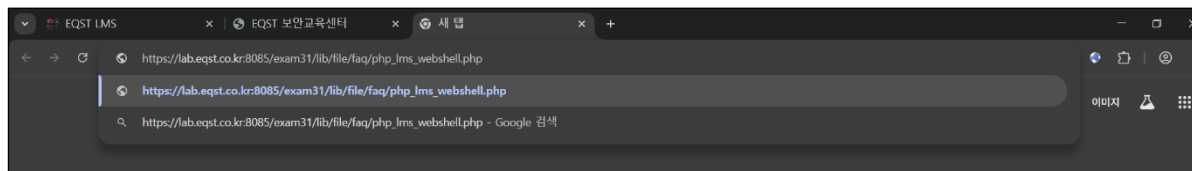
: 목표 페이지의 주소는 8083 이지만, 첨부파일의 주소는 8085 인 것을 확인할 수 있다.

([https://lab.eqst.co.kr:8085/exam31/lib/download.php?file\\_path=/file/faq/&file\\_name=php\\_lms\\_webshell.php](https://lab.eqst.co.kr:8085/exam31/lib/download.php?file_path=/file/faq/&file_name=php_lms_webshell.php))

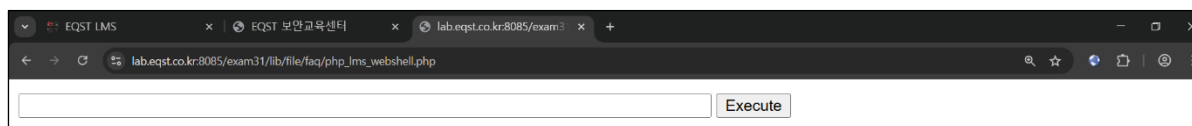


: 첨부파일의 주소를 참고하여 경로를 수정한 후 접속해본다.

([https://lab.eqst.co.kr:8085/exam31/lib/file/faq/php\\_lms\\_webshell.php](https://lab.eqst.co.kr:8085/exam31/lib/file/faq/php_lms_webshell.php))



: 웹쉘 실행에 성공한다.

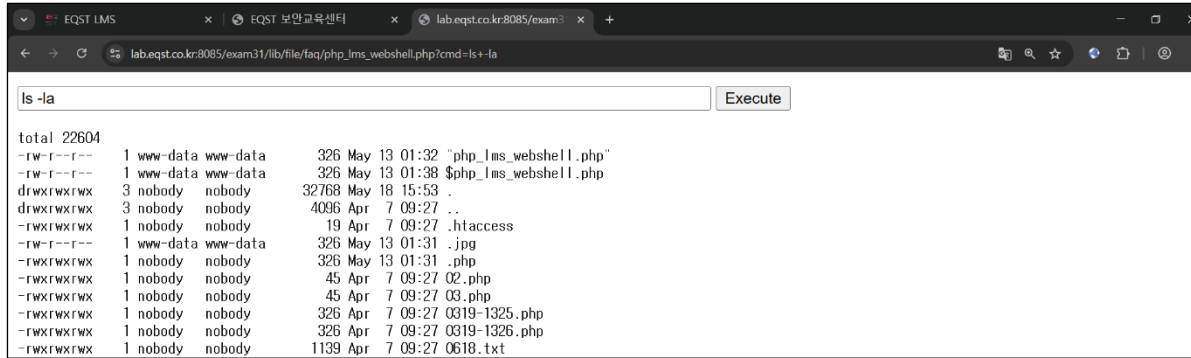


### Step 3. DB 설정 파일 내 DB 계정 비밀번호 탈취

#### Step 3-1. 웹셸 파악

: 실행된 웹셸의 검색창에 ls -la 를 입력해본다.

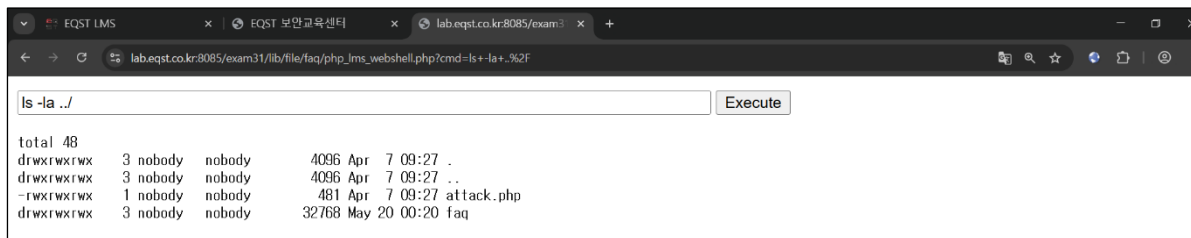
: 입력한 ls -la 가 정상적으로 작동하므로 웹셸이 리눅스 기반임을 알 수 있다.



```
total 22604
-rw-r--r-- 1 www-data www-data 326 May 13 01:32 "php_lms_webshell.php"
-rw-r--r-- 1 www-data www-data 326 May 13 01:38 $php_lms_webshell.php
drwxrwxrwx 3 nobody nobody 32768 May 18 15:53 .
drwxrwxrwx 3 nobody nobody 4096 Apr 7 09:27 ..
-rwxrwxrwx 1 nobody nobody 19 Apr 7 09:27 .htaccess
-rw-r--r-- 1 www-data www-data 326 May 13 01:31 .jpg
-rwxrwxrwx 1 nobody nobody 326 May 13 01:31 .php
-rwxrwxrwx 1 nobody nobody 45 Apr 7 09:27 02.php
-rwxrwxrwx 1 nobody nobody 45 Apr 7 09:27 03.php
-rwxrwxrwx 1 nobody nobody 326 Apr 7 09:27 0319-1325.php
-rwxrwxrwx 1 nobody nobody 326 Apr 7 09:27 0319-1326.php
-rwxrwxrwx 1 nobody nobody 1139 Apr 7 09:27 0618.txt
```

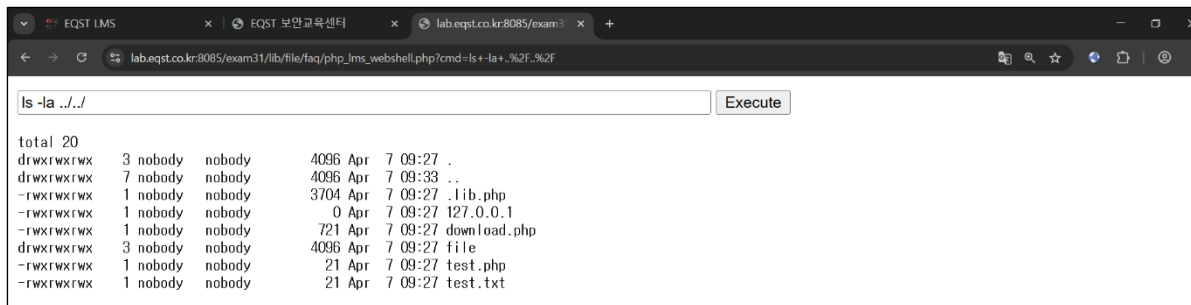
#### Step 3-2. DB 설정 파일 탐색

: ls -la 에 ../를 하나씩 붙여가며 상위 디렉토리를 탐색해본다.



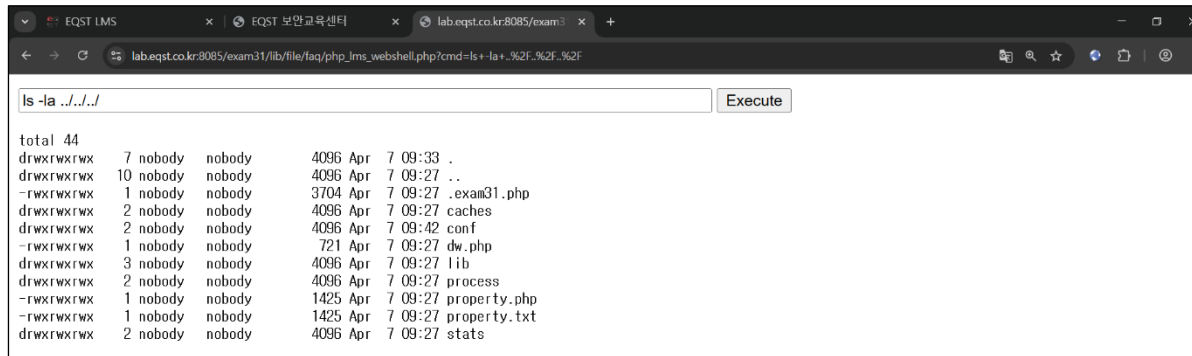
```
total 48
drwxrwxrwx 3 nobody nobody 4096 Apr 7 09:27 .
drwxrwxrwx 3 nobody nobody 4096 Apr 7 09:27 ..
-rwxrwxrwx 1 nobody nobody 481 Apr 7 09:27 attack.php
drwxrwxrwx 3 nobody nobody 32768 May 20 00:20 faq
```

(../.. 탐색)



```
total 20
drwxrwxrwx 3 nobody nobody 4096 Apr 7 09:27 .
drwxrwxrwx 7 nobody nobody 4096 Apr 7 09:33 ..
-rwxrwxrwx 1 nobody nobody 3704 Apr 7 09:27 .lib.php
-rwxrwxrwx 1 nobody nobody 0 Apr 7 09:27 127.0.0.1
-rwxrwxrwx 1 nobody nobody 721 Apr 7 09:27 download.php
drwxrwxrwx 3 nobody nobody 4096 Apr 7 09:27 file
-rwxrwxrwx 1 nobody nobody 21 Apr 7 09:27 test.php
-rwxrwxrwx 1 nobody nobody 21 Apr 7 09:27 test.txt
```

: ../..../ 디렉토리에서 DB 설정 파일로 추정되는 property.php 파일을 발견할 수 있다.



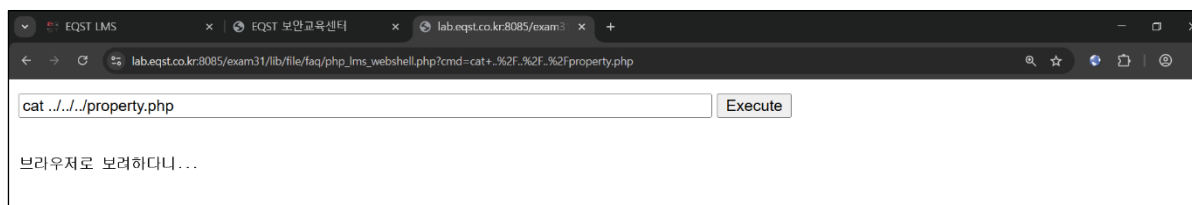
```
ls -la ../..../

total 44
drwxrwxrwx 7 nobody nobody 4096 Apr 7 09:33 .
drwxrwxrwx 10 nobody nobody 4096 Apr 7 09:27 ..
-rwxrwxrwx 1 nobody nobody 3704 Apr 7 09:27 .exam31.php
drwxrwxrwx 2 nobody nobody 4096 Apr 7 09:27 caches
drwxrwxrwx 2 nobody nobody 4096 Apr 7 09:42 conf
-rwxrwxrwx 1 nobody nobody 721 Apr 7 09:27 dw.php
drwxrwxrwx 3 nobody nobody 4096 Apr 7 09:27 lib
drwxrwxrwx 2 nobody nobody 4096 Apr 7 09:27 process
-rwxrwxrwx 1 nobody nobody 1425 Apr 7 09:27 property.php
-rwxrwxrwx 1 nobody nobody 1425 Apr 7 09:27 property.txt
drwxrwxrwx 2 nobody nobody 4096 Apr 7 09:27 stats
```

### Step 3-3. DB 설정 파일 확인

: 발견한 property.php 파일을 cat 명령어로 출력해본다.

: 브라우저로 내용을 확인할 수 없게 설정되어 있음을 알 수 있다.

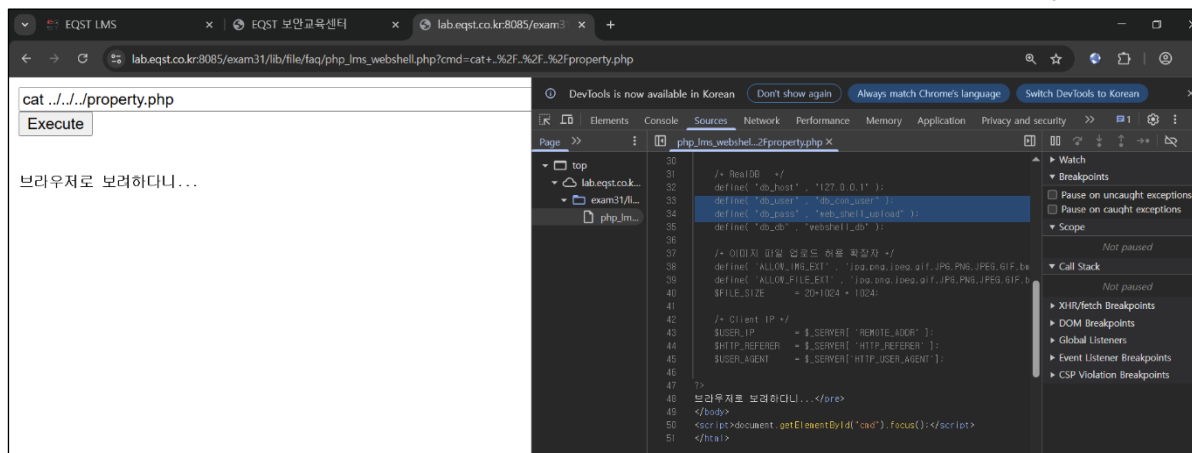


```
cat ../..../property.php

브라우저로 보려하니...
```

: 개발자도구를 이용하여 코드를 확인한다.

: 주석 처리된 DB User 명과 DB 계정 비밀번호를 획득할 수 있다. (정답: web\_shell\_upload)



```
cat ../..../property.php

브라우저로 보려하니...
```

```
30  /* RealDB */
31  define('db_host', '127.0.0.1');
32  define('db_user', 'lab_exam_user');
33  define('db_pass', 'web_shell_upload');
34  define('db_db', 'webshell_db');
35
36
37  /* DB User 및 암호를 불러오기 */
38  define('ALLOW_IMG_EXT', 'jpg.png.jpeg.gif.JPG.PNG.JPEG.GIF.ba');
39  define('ALLOW_FILE_EXT', 'log.png.jpeg.gif.JPG.PNG.JPEG.GIF.b');
40  $FILE_SIZE = 20*1024 + 1024;
41
42  /* Client IP */
43  $USER_IP = $_SERVER['REMOTE_ADDR'];
44  $HTTP_REFERER = $_SERVER['HTTP_REFERER'];
45  $USER_AGENT = $_SERVER['HTTP_USER_AGENT'];
46
47  To
48  브라우저로 보려하니...</pre>
49  </body>
50  <script>document.getElementById('cad').focus();</script>
51  </html>
```

성명	프로젝트 후 소감
박기쁨	<p>File Upload 취약점에 대해 실습하며, 다양한 방식으로 파일 업로드 필터링을 우회할 수 있음을 배웠다. 수많은 필터링 속에서도, 공격자의 입장에서는 우회 방법을 단 한 가지라도 발견한다면, 방어자의 입장에서는 취약점이 단 한 가지라도 존재한다면, 결국 해킹이 이루어지게 되고 정보가 누출된다는 점이 참 어렵고도 잔인하다고 느껴졌다.</p> <p>또, 파일 업로드 우회에 이어 실제로 웹쉘을 이용하여 DB 정보를 탈취하는 실습을 통해, 실제 해킹 시 사용될 법한 방법과 과정을 직접 경험해볼 수 있어 매우 흥미로웠다.</p>