□ **단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿**

25년 1학기 인터넷 보안 수업

File Upload

2025년 5월 13일

학번 : 32231594
이름 : 박기쁨

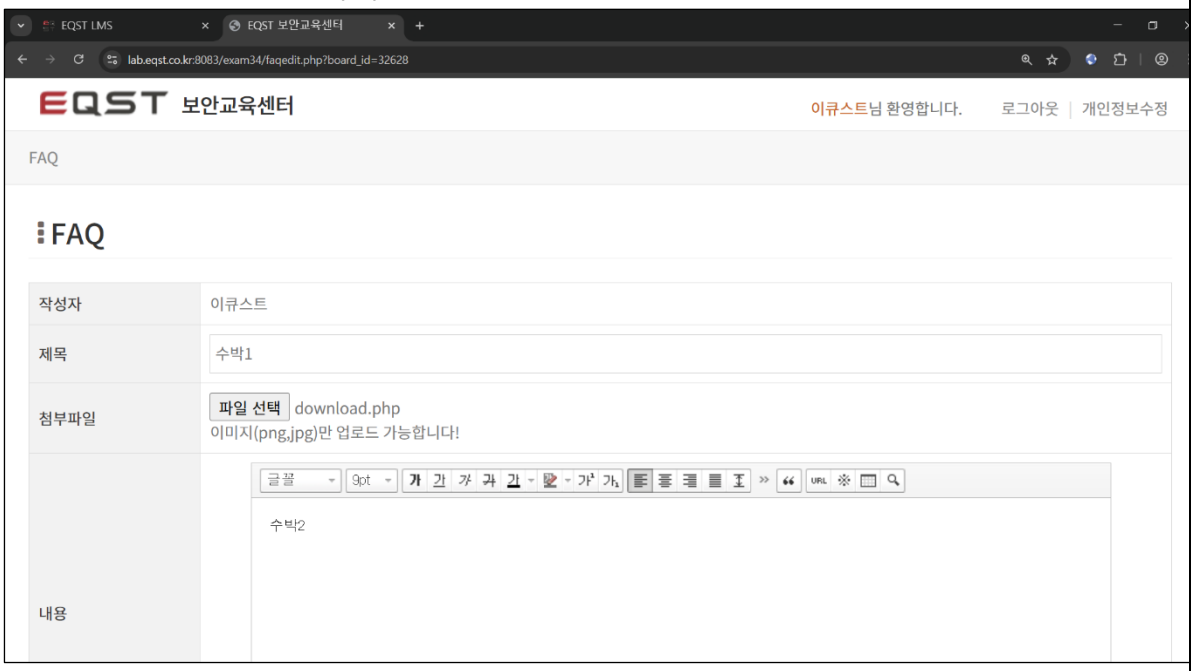
1. 워게임 5 번 문항

<과정 설명>

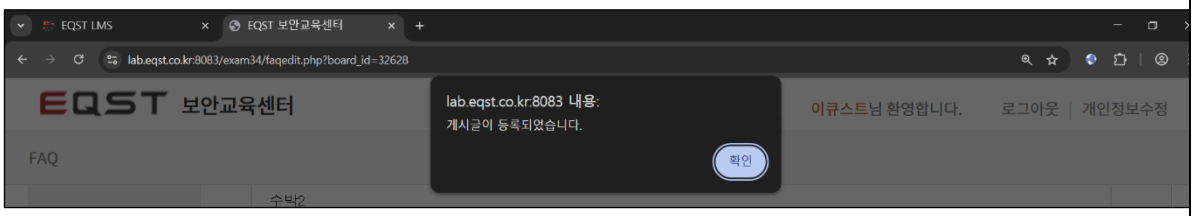
File Upload 취약점 공격은 파일을 업로드하는 과정에서의 취약점을 이용하여 원하는 정보를 탈취하는 행위를 말한다. 본 실습에서는 File Upload 취약점을 이용해 .php 확장자의 파일을 업로드 해보고자 한다.

Step 1. php 파일 업로드

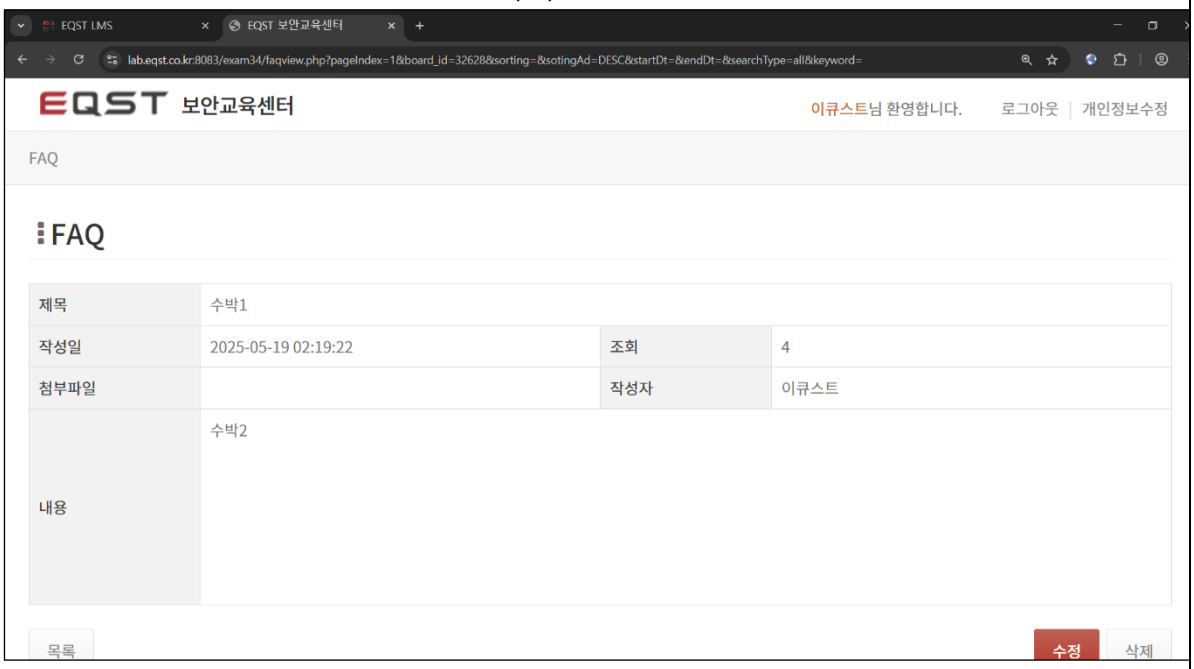
Step 1-1. php 파일 업로드
: 목표 페이지의 게시판에 php 파일을 업로드 한다.



(게시물이 정상적으로 업로드 된다.)



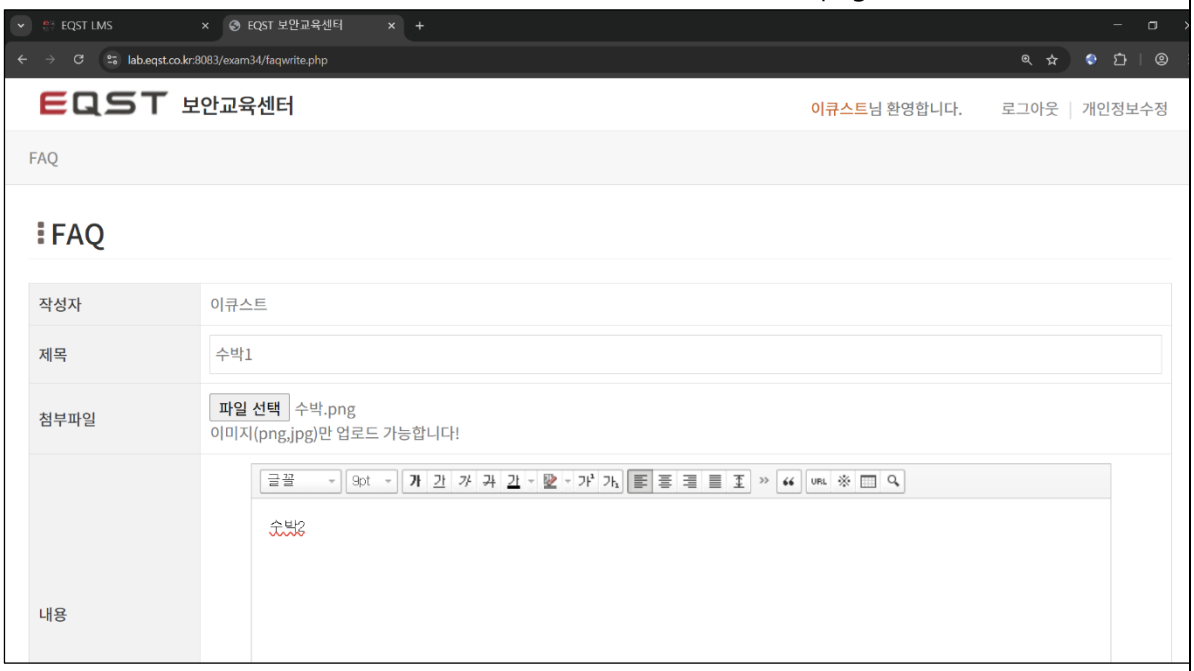
: 게시물은 정상적으로 업로드 되었지만, php 파일은 삭제된 것을 알 수 있다.



Step 2. png 파일 업로드

Step 2-1. png 파일 업로드

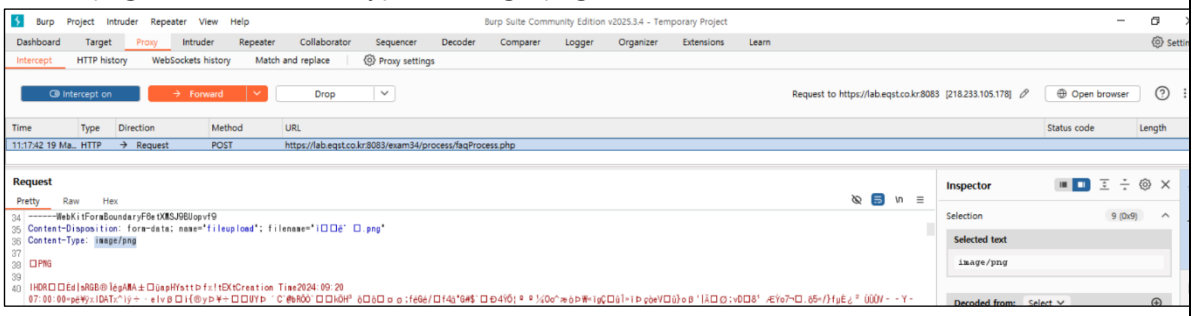
: 목표 페이지의 게시판 파일 업로드 방식 등을 확인하기 위하여 png 파일을 업로드 해본다.



Step 2-2. intercept 및 content-Type 확인

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

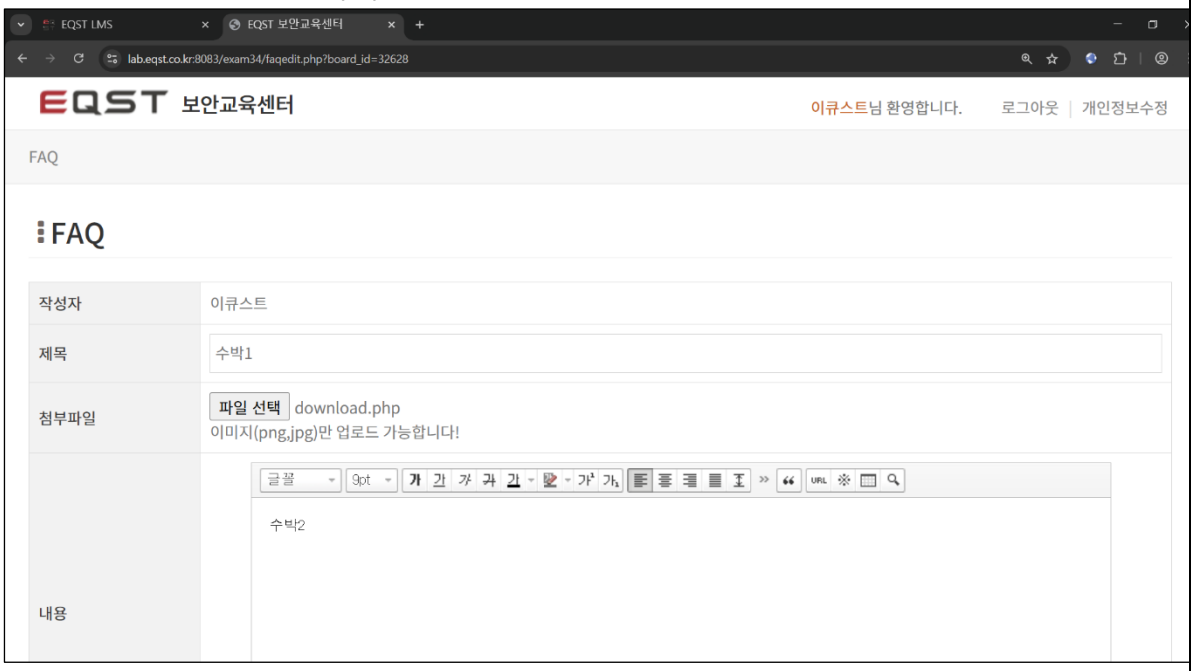
: 수박.png 파일의 Content-Type 은 image/png 이다.



Step 3. php 파일 업로드 (재시도)

Step 3-1. php 파일 업로드 (재시도)

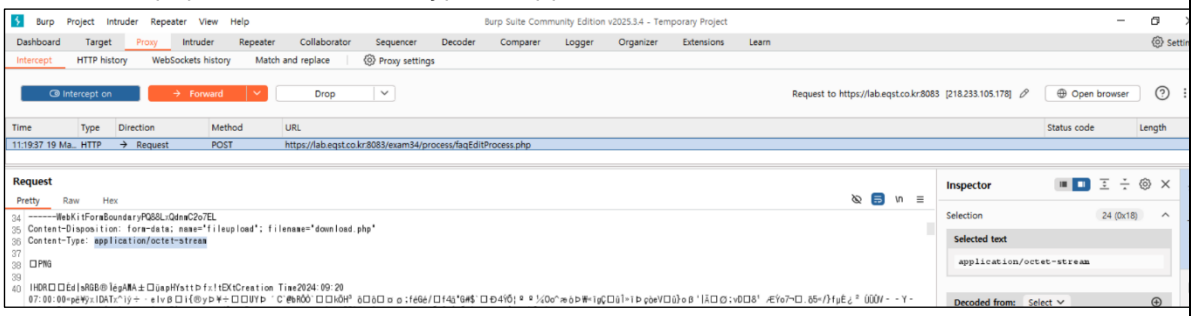
: 목표 페이지의 게시판에 php 파일을 업로드 한다.



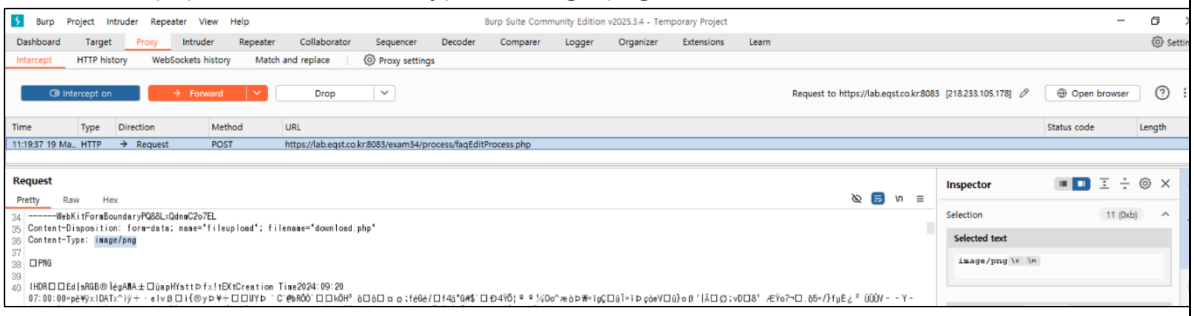
Step 3-2. intercept 및 content-Type 확인

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

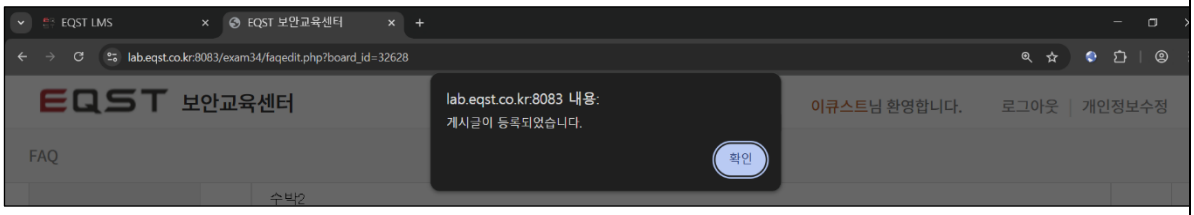
: download.php 파일의 Content-Type 은 application/octet-stream 이다.



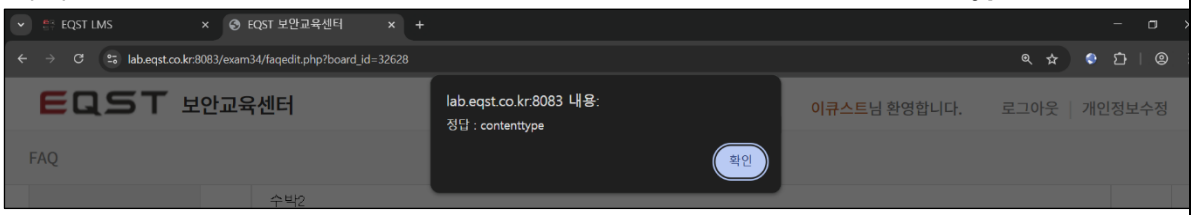
: download.php 파일의 Content-Type 을 image/png 로 변경한다.



(게시물이 정상적으로 업로드 된다.)



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

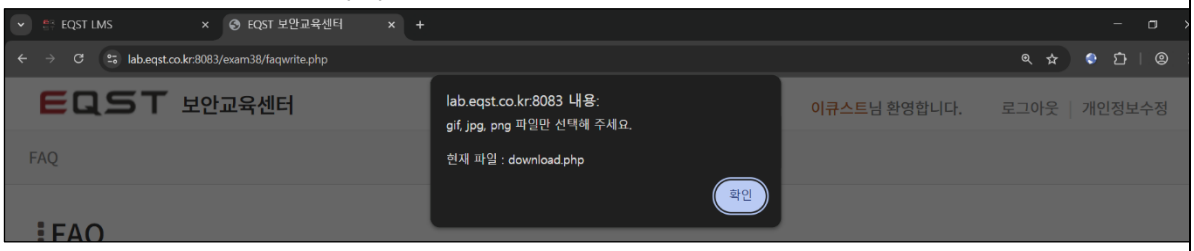


2. 워게임 6 번 문항

<과정 설명>

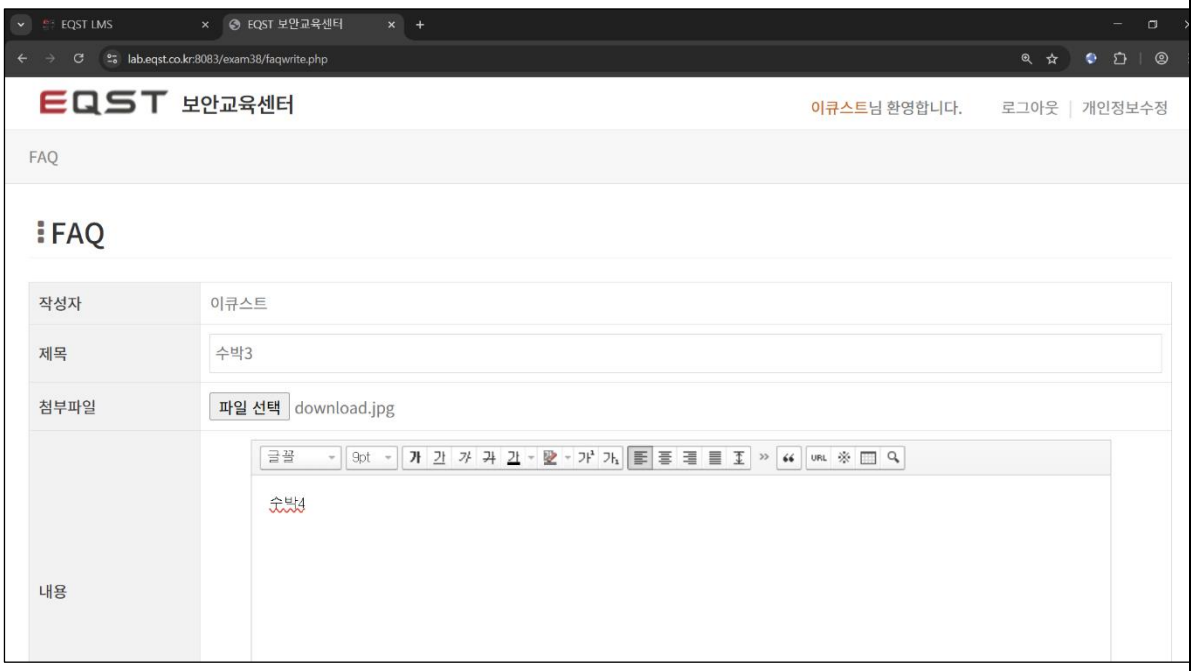
Step 1. php 파일 업로드

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

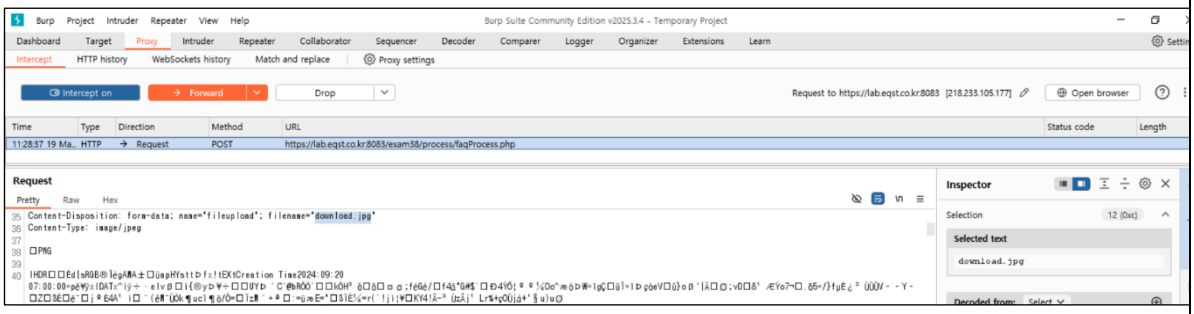
Step 2-1. php 파일 확장자 변경
: 첨부파일 선택 과정에서의 확장자 필터링을 우회하기 위해, php 파일의 확장자를 jpg 로 변경하여 업로드한다.
: 첨부파일 선택 과정에서 필터링 되지 않는다.



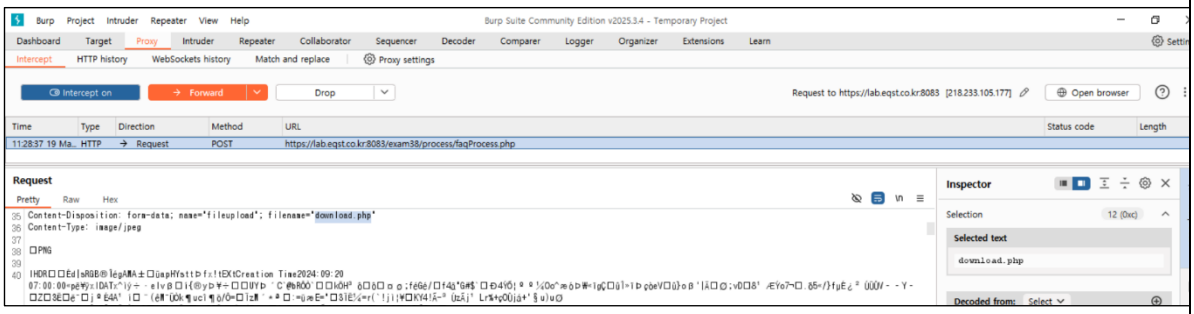
Step 3. php 파일 확장자 원상복구 (재시도)

Step 3-1. php 파일 확장자 원상복구

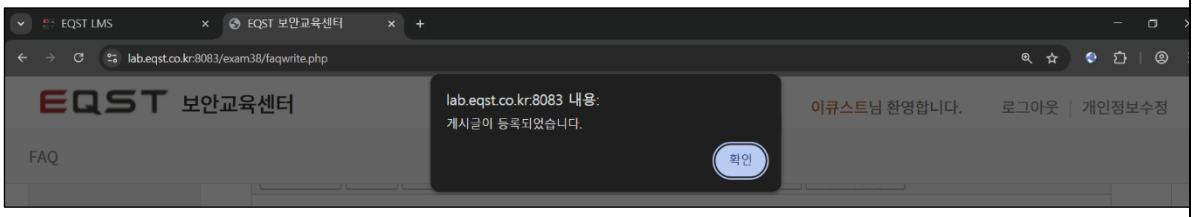
: 게시물을 저장 과정을 intercept 하여 확장자를 php 로 원상복구한다.



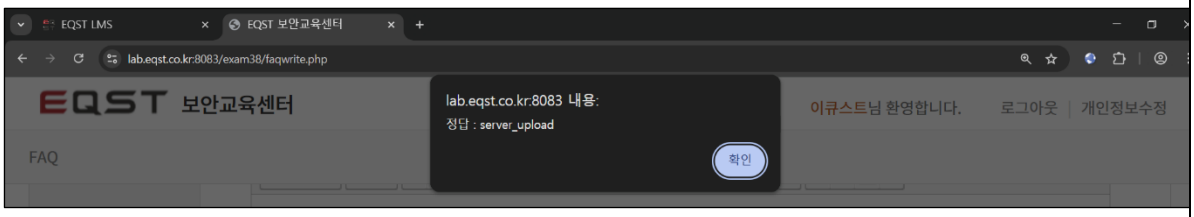
(.jpg 를 .php 로 변경한다.)



(게시물이 정상적으로 업로드 된다.)



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



3. 워게임 7 번 문항

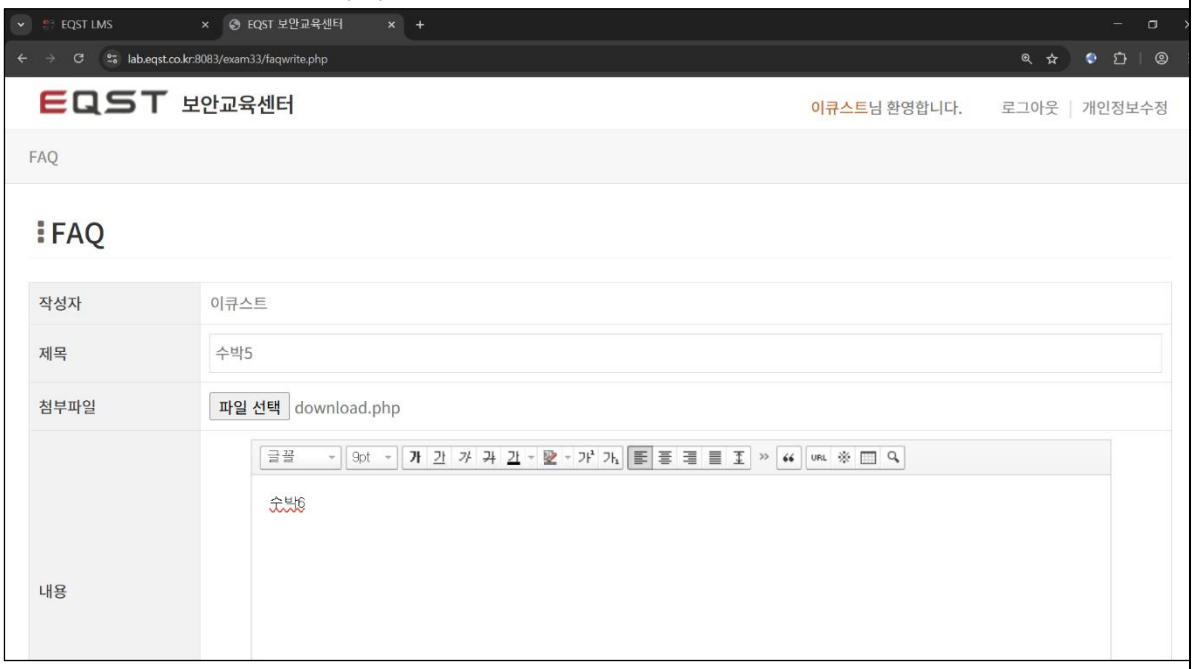
<과정 설명>

Step 1. php 파일 업로드

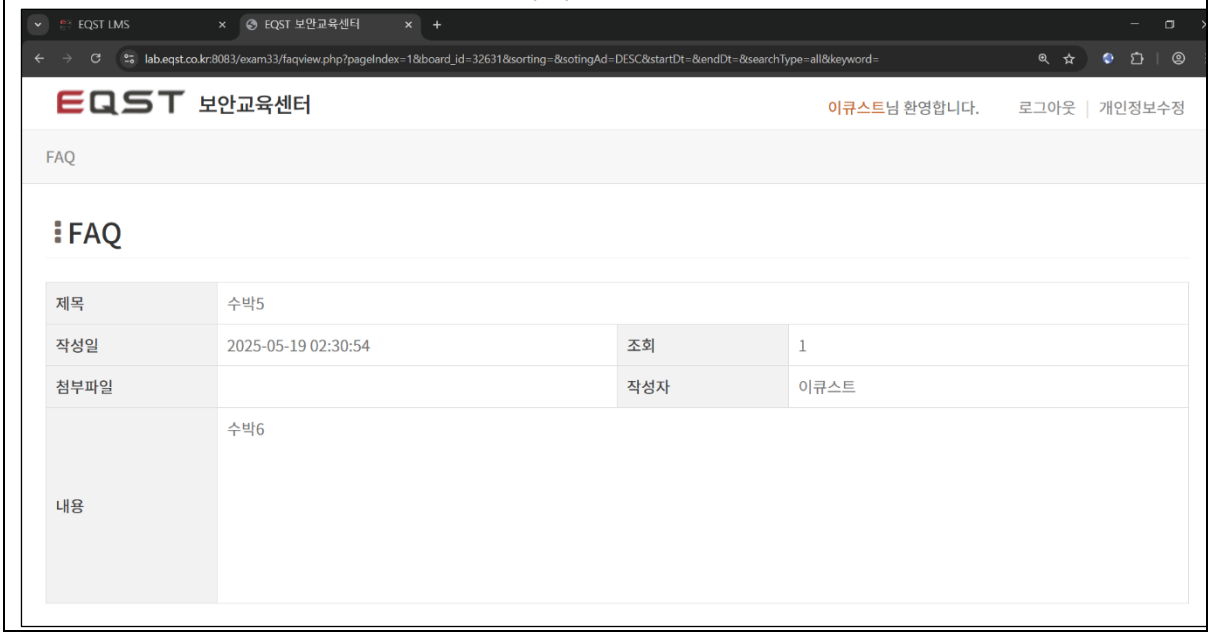
Step 1-1. php 파일 업로드

: 목표 페이지의 게시판에 php 파일을 업로드 한다.

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



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

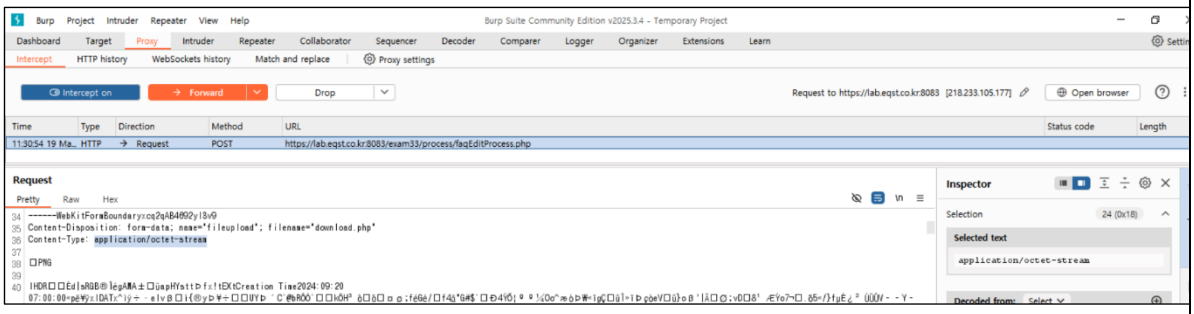


Step 2. php 파일 업로드 및 Content-Type 변경

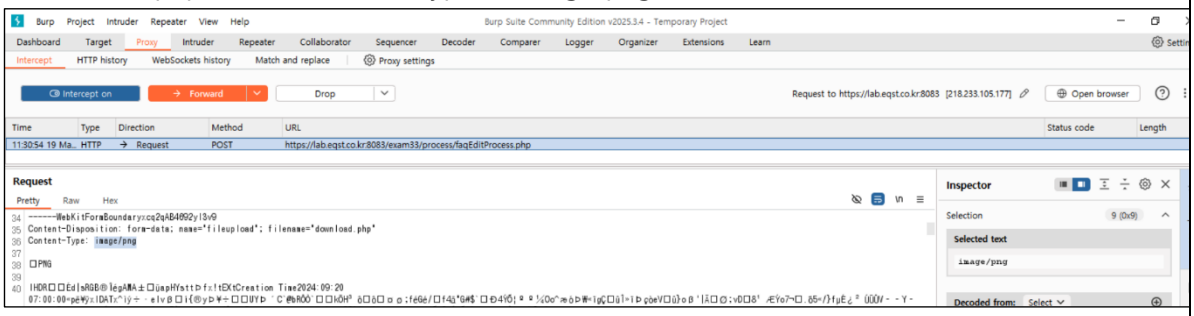
Step 2-1. intercept 및 content-Type 확인

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

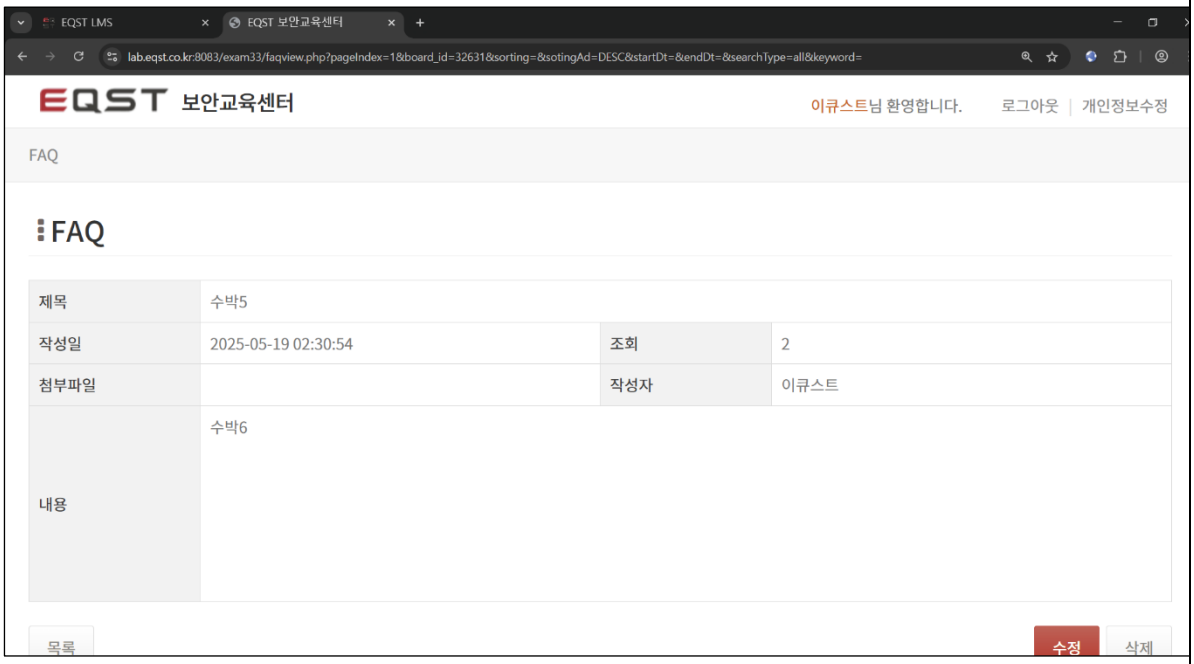
: download.php 파일의 Content-Type 은 application/octet-stream 이다.



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



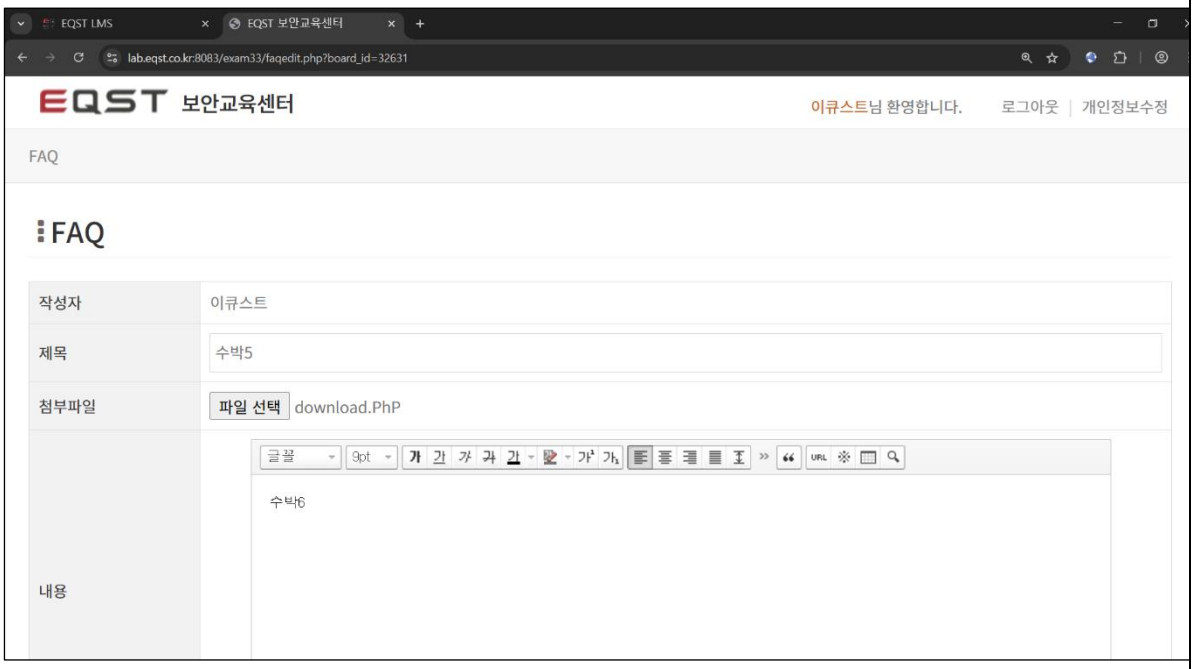
: 여전히 게시물은 정상적으로 업로드 되었지만, php 파일은 삭제된 것을 알 수 있다.



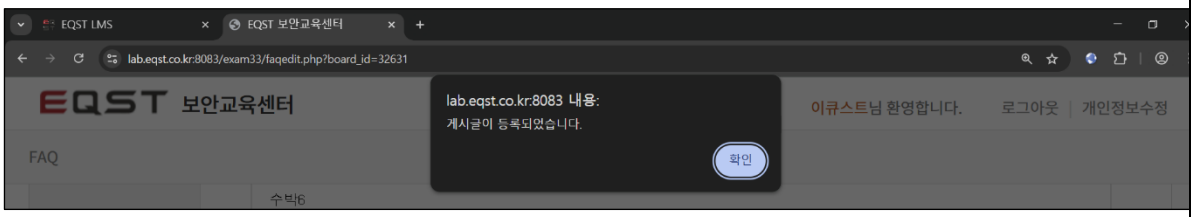
Step 3. php 파일 확장자 PhP 로 변경

Step 3-1. php 파일 확장자 PhP 로 변경

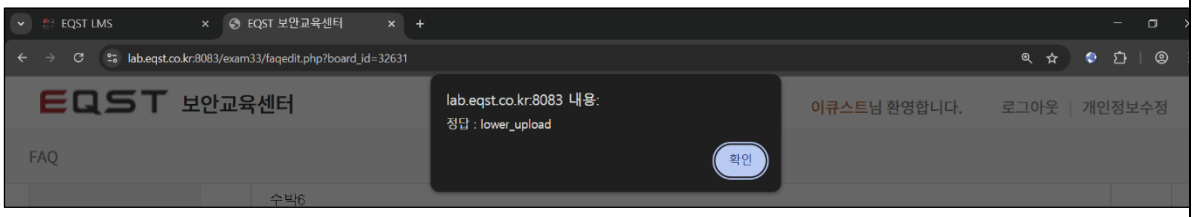
: .php 확장자 필터링을 우회하기 위해 php 파일의 확장자에 대문자를 섞어 PhP 로 변경한다.



(게시물이 정상적으로 업로드 된다.)



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



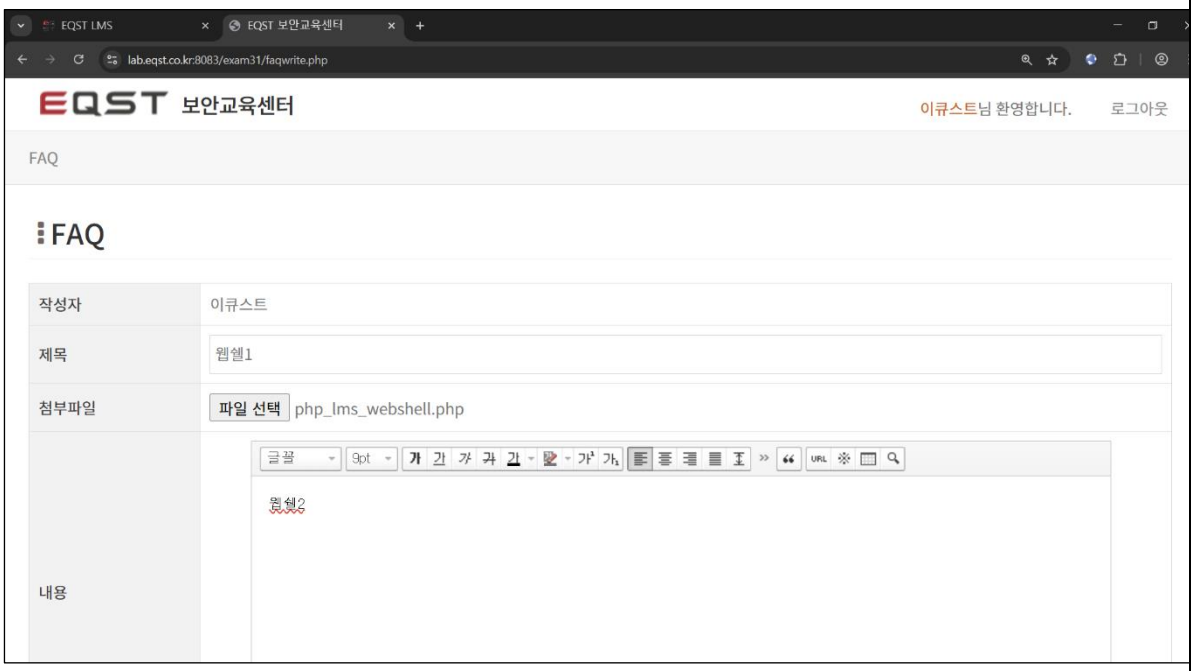
4. 워게임 9 번 문항

<과정 설명>

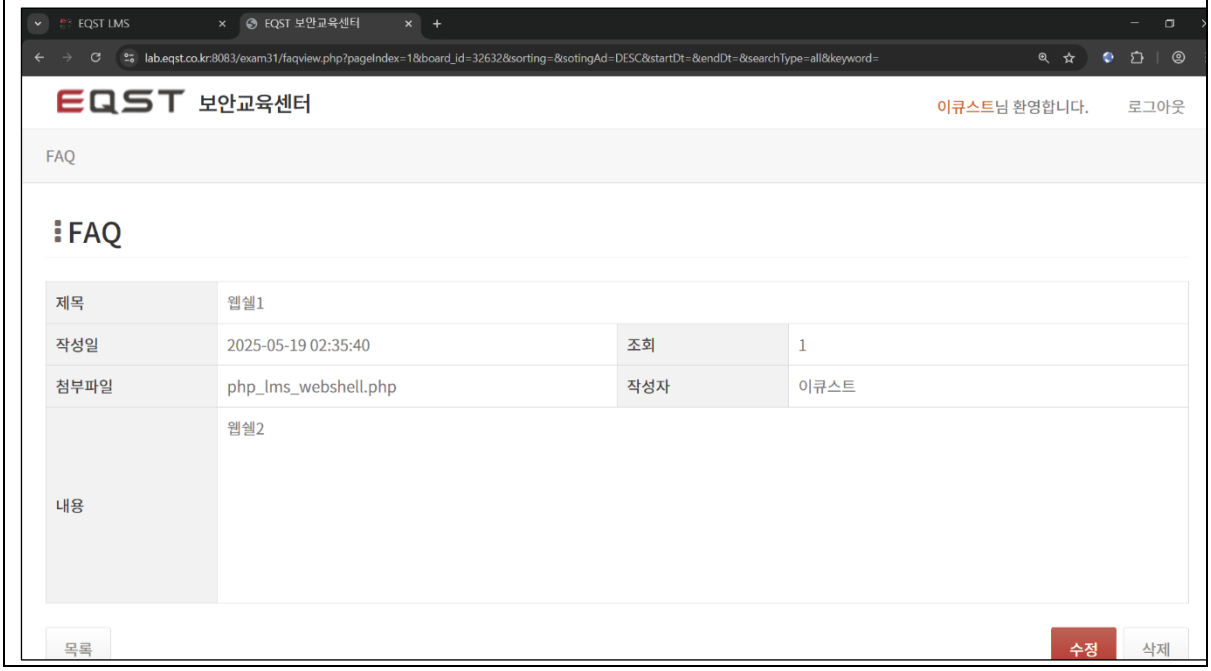
Step 1. 웹쉘 업로드

Step 1-1. 웹쉘 업로드

: 목표 페이지의 게시판에 웹쉘을 업로드 한다.



(게시물이 정상적으로 업로드 된다.)

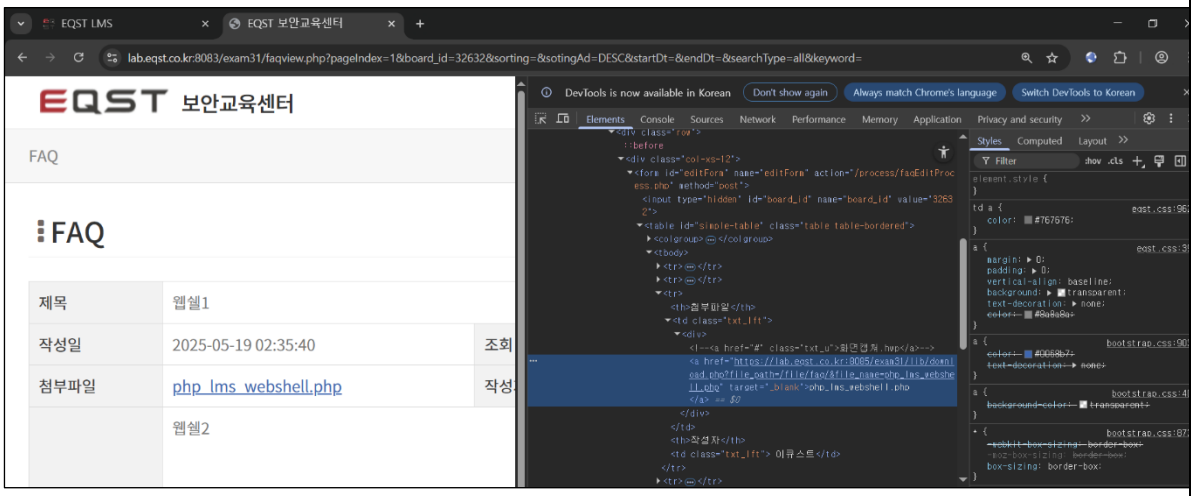


Step 2. 웹쉘 경로 찾기 및 실행하기

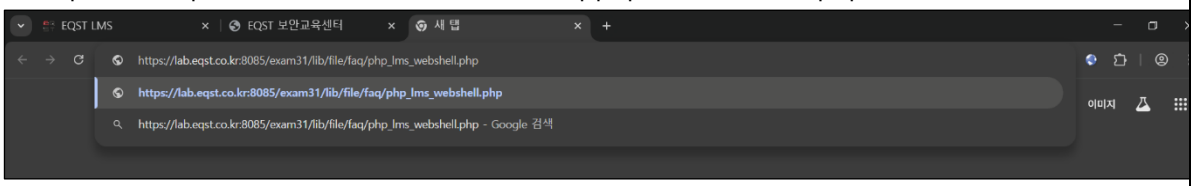
Step 2-1. 웹쉘 경로 찾기

: 개발자도구를 사용하여 첨부파일 관련 코드를 확인한다.

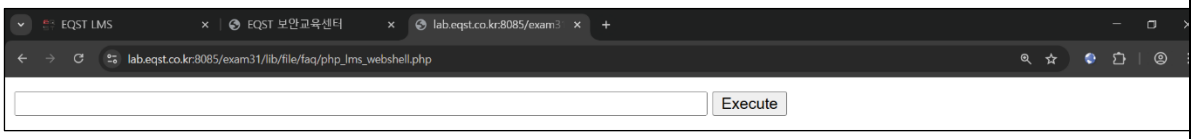
: 목표 페이지의 주소는 8083 이지만, 첨부파일의 주소는 8085 인 것을 확인할 수 있다.
(https://lab.eqst.co.kr:8085/exam31/lib/download.php?file_path=/file/faq/&file_name=php_lms_webshell.php)



: 첨부파일의 주소를 참고하여 경로를 수정한 후 접속해본다.
(https://lab.eqst.co.kr:8085/exam31/lib/file/faq/php_lms_webshell.php)



: 웹쉘 실행에 성공한다.

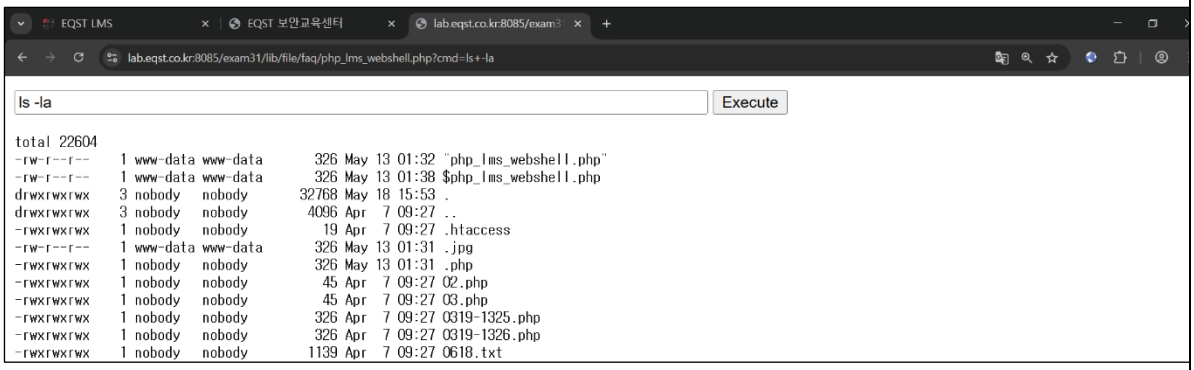


Step 3. DB 설정 파일 내 DB 계정 비밀번호 탈취

Step 3-1. 웹쉘 파악

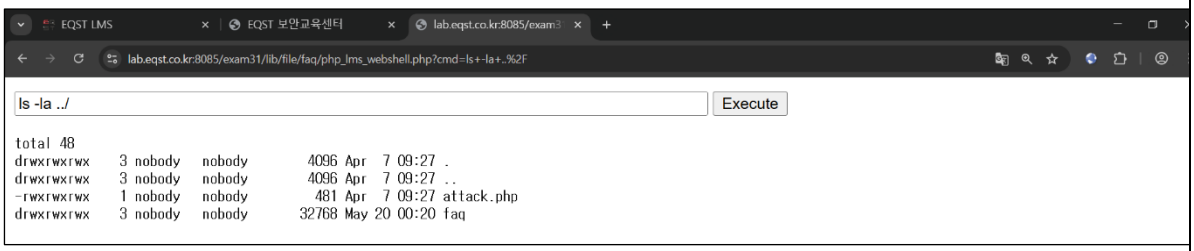
: 실행된 웹쉘의 검색창에 ls -la 를 입력해본다.

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

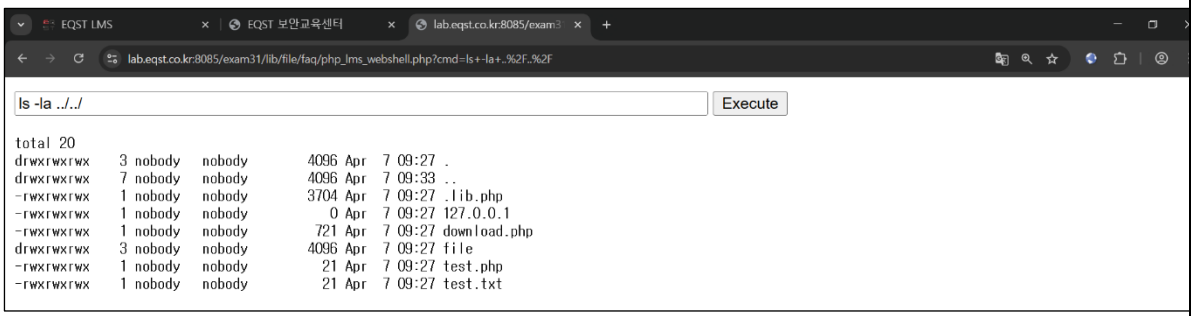


Step 3-2. DB 설정 파일 탐색

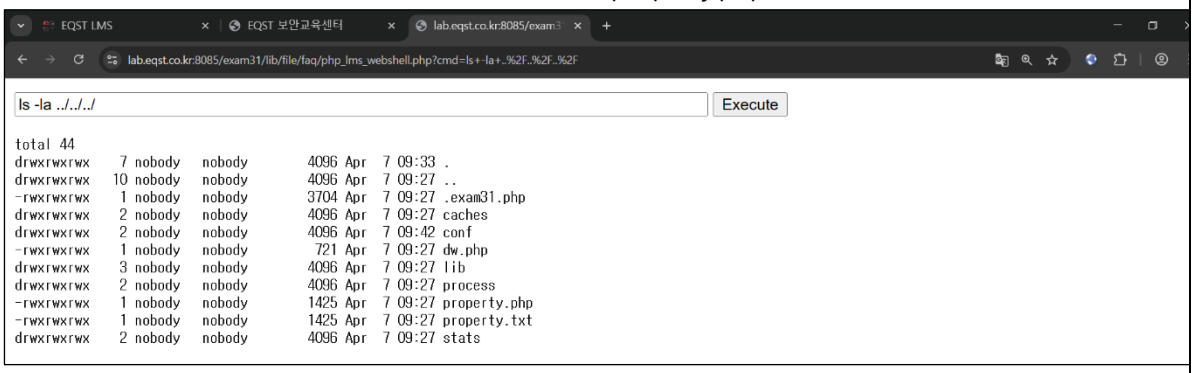
: ls -la 에 ../를 하나씩 붙여가며 상위 디렉토리를 탐색해본다.



(../../ 탐색)



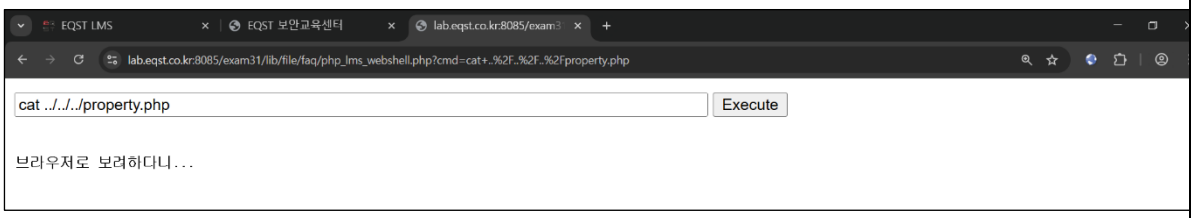
: ../../../ 디렉토리에서 DB 설정 파일로 추정되는 property.php 파일을 발견할 수 있다.



Step 3-3. DB 설정 파일 확인

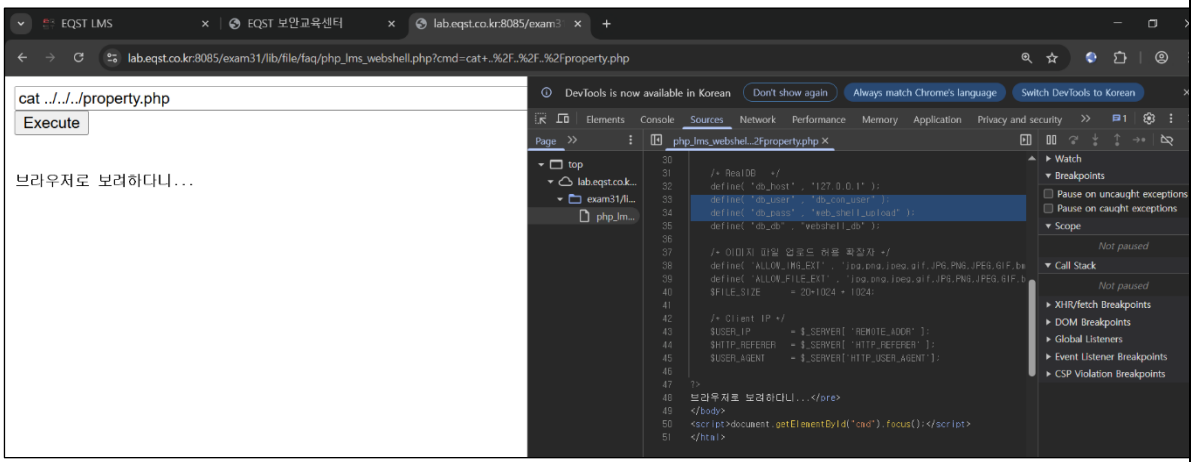
: 발견한 property.php 파일을 cat 명령어로 출력해본다.

: 브라우저로 내용을 확인할 수 없게 설정되어 있음을 알 수 있다.



: 개발자도구를 이용하여 코드를 확인한다.

: 주석 처리된 DB User 명과 DB 계정 비밀번호를 획득할 수 있다. (정답: web_shell_upload)



성명	프로젝트 후 소감
박기쁨	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. 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.