실습 개요

1. 파일 업로드 기능을 통한 보안 취약점 파악
2. 악성 확장자 파일 fiddler를 통한 우회 업로드

실습 개요

1단계: 피들러에서 업로드 요청 가로채는 방법

1단계: 피들러에서 업로드 요청 가로채는 방법

1.Flask 웹 서버를 실행합니다 (basic_upload.py)

2.웹 브라우저 접속

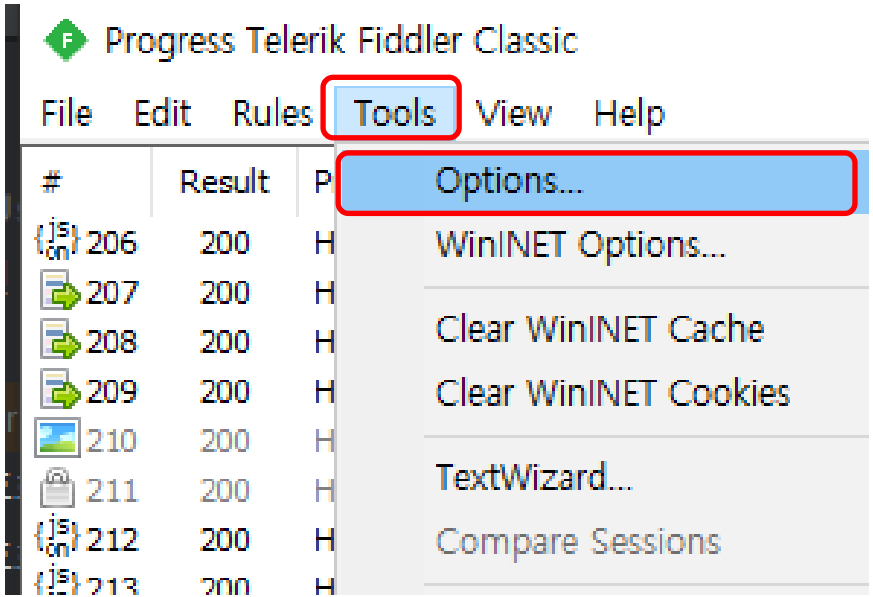
3.피들러(Fiddler)를 실행하고, **"HTTPS 해독(Decrypt HTTPS)" 기능을 활성화**

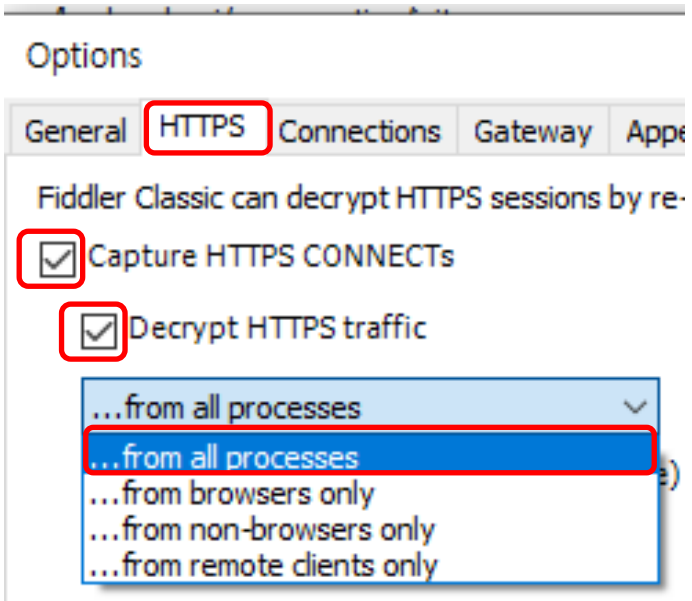
4. .txt나 .jpg 같은 정상 파일을 선택하고 업로드

5. 업로드 요청이 피들러의 **"Web Sessions" 창에 표시되면** 해당 요청을 클릭

6.마우스 오른쪽 클릭 → **"Replay > Reissue and Edit"**를 선택

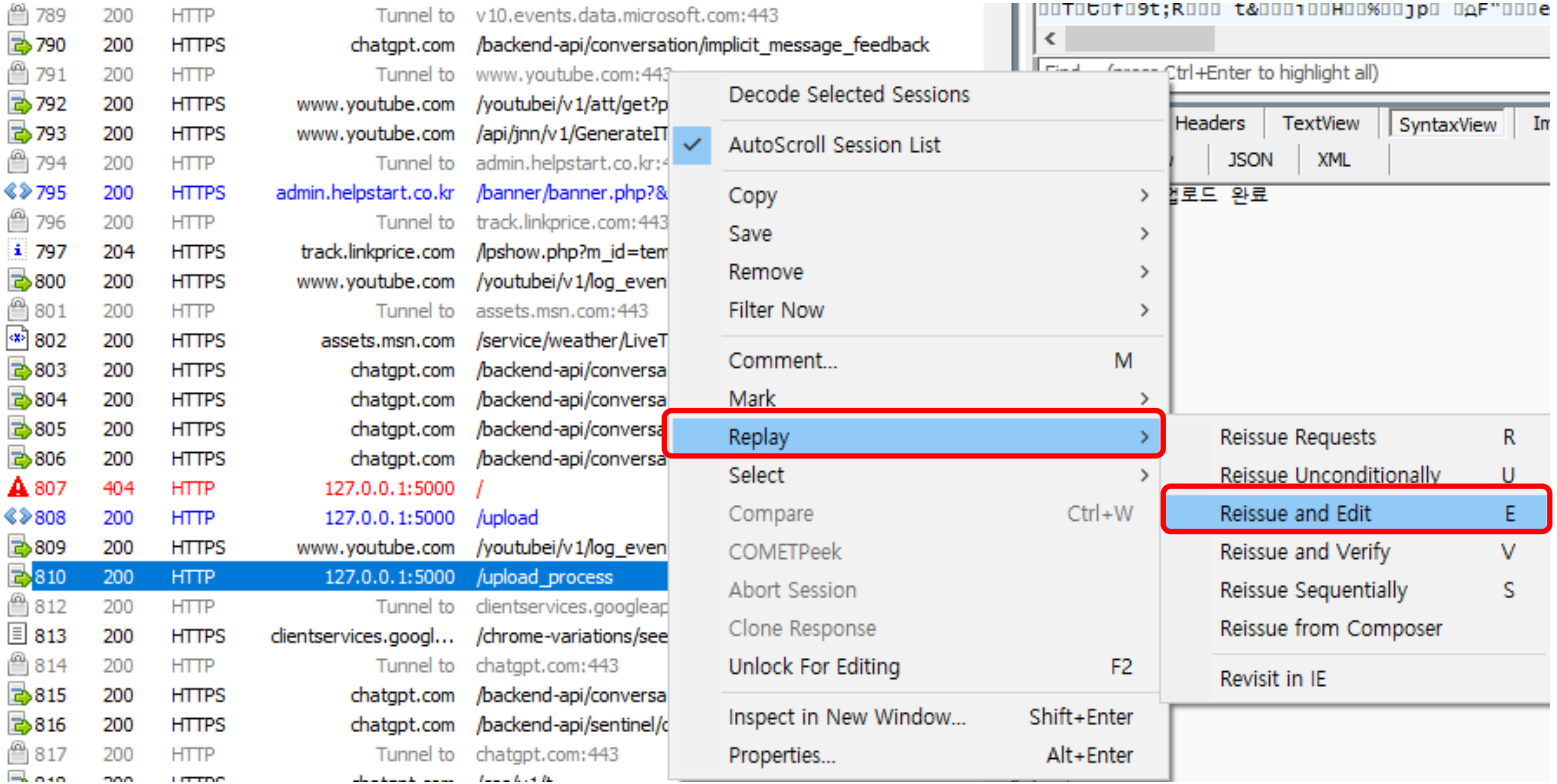
HTTPS 해독 기능 활성화





Tools->options->https->capture .... 체크-> Decrypt...체크-> from all processes 선택-> 경고창 나오면 루트 인증서 설치

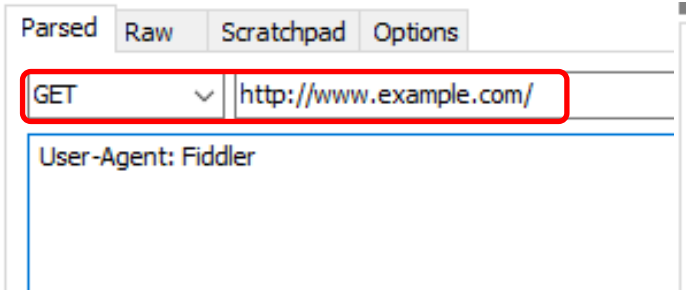
Replay > Reissue and Edit

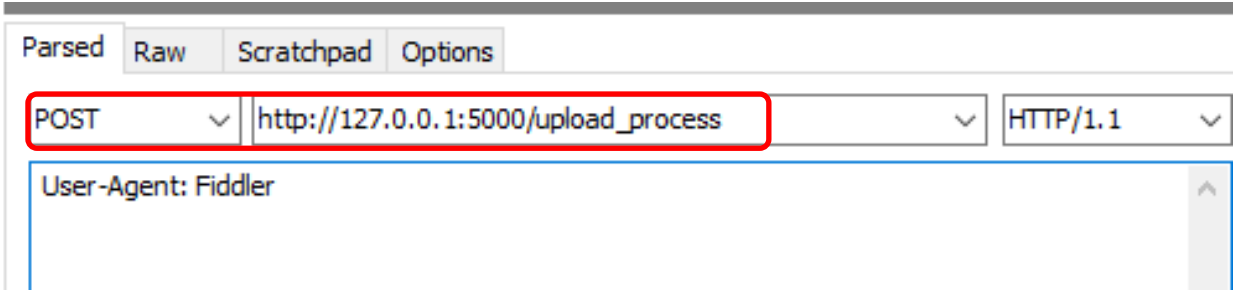


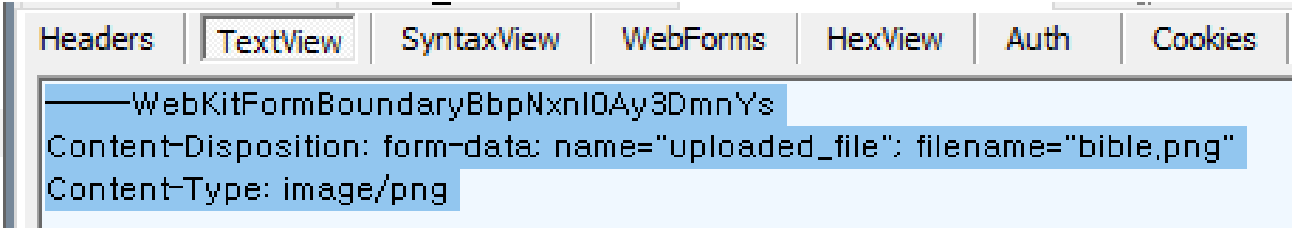
Web Sessions /**upload_process -> replay -> reissue and edit**

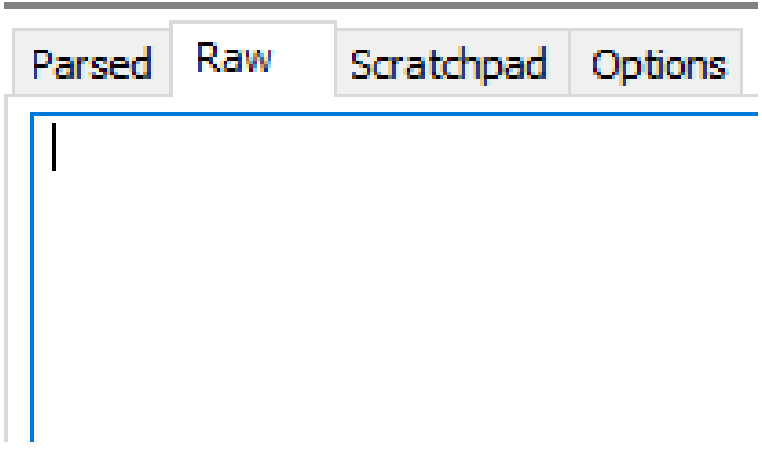
실습 개요

2단계: .php 확장자 조작 및 Content-Type 변경 실습


Parsed
Raw
Scratchpad
Options
GET
http://www.example.com/
User-Agent: Fiddler


Parsed
Raw
Scratchpad
Options
POST
http://127.0.0.1:5000/upload_process
HTTP/1.1
User-Agent: Fiddler


Headers
TextView
SyntaxView
WebForms
HexView
Auth
Cookies
------WebKitFormBoundaryBbpNxnI0Ay3DmnYs
Content-Disposition: form-data; name="uploaded_file"; filename="bible.png"
Content-Type: image/png


Parsed
Raw
Scratchpad
Options

Use this page to compose a Request. You can clone a prior request by dragging and dropping a session from the Web Sessions list.

Execute

Parsed | Raw | Scratchpad | Options

```
POST /upload_process HTTP/1.1
Host: 127.0.0.1:5000
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryrCTJCqIsLQsL6Q4T


------WebKitFormBoundaryrCTJCqIsLQsL6Q4T
Content-Disposition: form-data; name="uploaded_file"; filename="html.php"
Content-Type:image/jpeg

코드테스트


------WebKitFormBoundaryrCTJCqIsLQsL6Q4T--
```

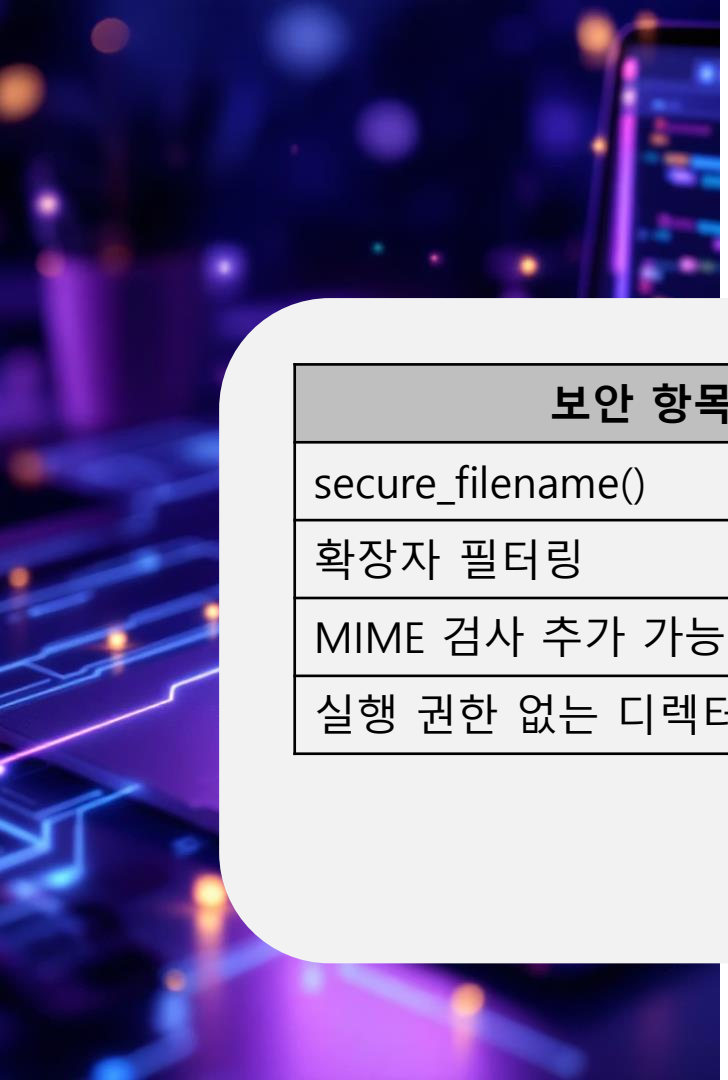
Content-Type: image/png : 실제로는 PHP 파일인데 마치 이미지처럼 보이게 속이는 것

실습 개요

3단계: 서버 업로드 결과 확인

실습 개요

보안 적용 포인트



보안 적용 포인트

보안 항목	설명
secure_filename()	위험한 문자 제거, 디렉터리 탈출 방지
확장자 필터링	.php, .exe 등의 위험 확장자 차단
MIME 검사 추가 가능	Content-Type을 서버에서 다시 확인 가능
실행 권한 없는 디렉터리로 저장	웹에서 바로 실행되지 않게 함