SUA 모의해킹

모의해킹 결과 보고서 (예비)

이선아

목차

1 Stored XSS

1.1 취약점 개요

사용자의 입력 값 검증 미흡으로 인해 스크립트 코드 실행 가능

1.2 상세 설명

도메인 : http://124.60.4.10:6662/

발생 위치 : 게시글 작성 (write.php)

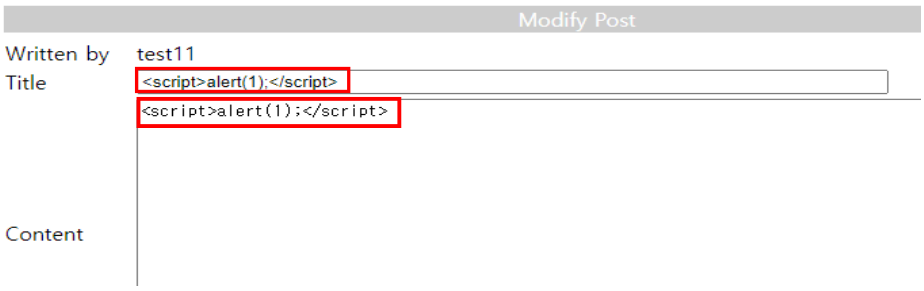

그림 1 xss 공격 쿼리 삽입

게시판에 글을 작성하는 write 서비스에 글 제목과 내용을 작성하는 곳에 스크립트 구문을 입력한다.

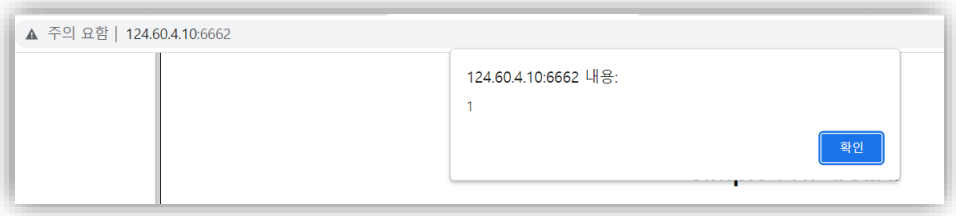

그림 2 제목과 본문에 각각 xss 실행

1.3 조치 방안

1. 게시글 모든 입력 가능한 곳에 xss 필터링 추가
2. 스크립트에서 주로 사용되는 특수문자 치환

```
< → &lt;
> → &gt;
" → &quot;
' → &#39;
```

2 불충분한 인증 (1)

2.1 취약점 개요

사용자, 세션 등의 인증 절차 미흡으로 권한이 없는 사용자가 게시글 작성, 수정, 삭제되도록 허용

2.2 상세 설명

도메인 : http://124.60.4.10:6662/

발생 위치 : 게시글 작성 (write.php)

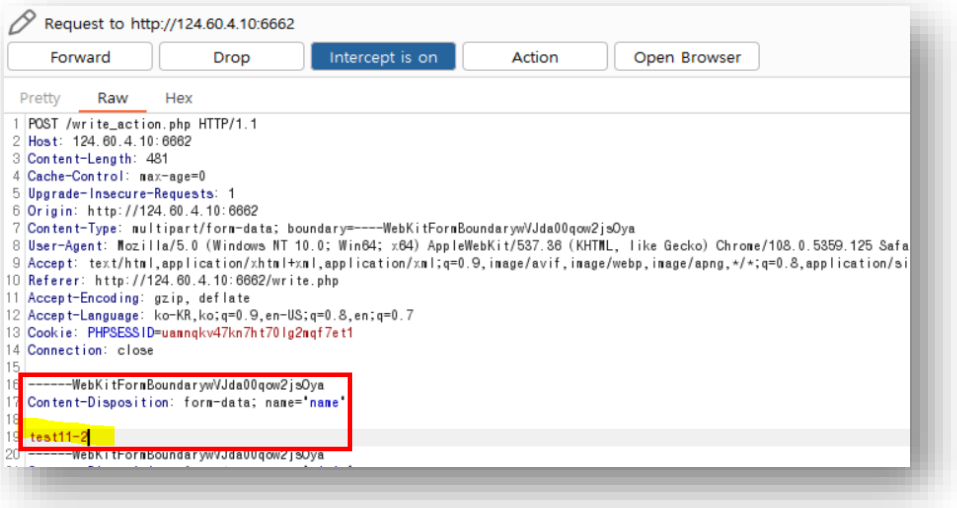

그림 3 작성자 임의로 수정

로그인한 사용자 test11 로 게시글 작성 후 버프 스위트로 패킷을 잡는다. 작성자 이름 name 에 들어가는 값을 test11-2 로 임의로 수정한다.

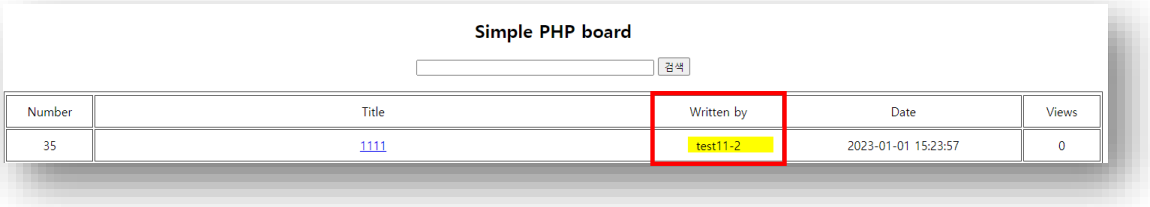

그림 4 임의의 사용자로 작성자 수정

분명 test11 로 로그인 후 작성한 글임에도 불구하고 임의의 사용자가 작성한 글이 올라간다. 회원이 아닌 검증되지 않은 사용자가 게시글을 작성할 수 있고 해당 게시글에 악의적인 코드 삽입 등이 가능하다.

2.3 조치 방안

1. 글이 작성되고 업로드될 때 세션값 검증
2. 작성자의 필드값에 현재 로그인한 사용자 아이디가 들어가도록 정적으로 설정

3 불충분한 인증(2)

3.1 취약점 개요

사용자, 세션 등의 인증 절차 미흡으로 권한이 없는 사용자가 게시글 작성, 수정, 삭제되도록 허용

3.2 상세 설명

도메인 : http://124.60.4.10:6662/

발생 위치 : 게시글 작성 (write.php)

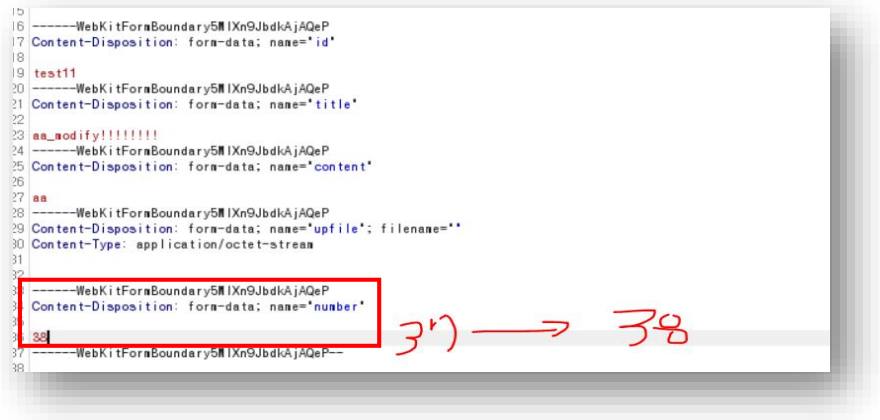

그림 5 게시글번호 수정

사용자가 임의로 게시글 37 번을 작성한 후 버프 스위트로 패킷을 잡는다. 패킷에는 게시글번호인 number 필드 값을 37 에서 38 로 고치면, 다른 사람이 작성한 38 번 게시글의 내용을 수정한다.

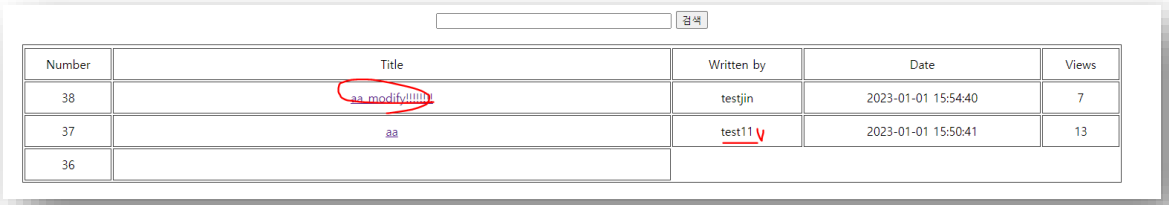

그림 6 수정된 게시글

3.3 조치 방안

1. 사용자가 게시글 작성시 입력값들이 서버 쪽에서 처리되도록 조치

4 정보 누출

4.1 취약점 개요

에러 발생 시 중요 정보가 노출되어 공격 시도에 유리한 정보가 될 수 있음

4.2 상세 설명

도메인 : http://124.60.4.10:6662

발생 위치 : http://124.60.4.10:6662/admin.php

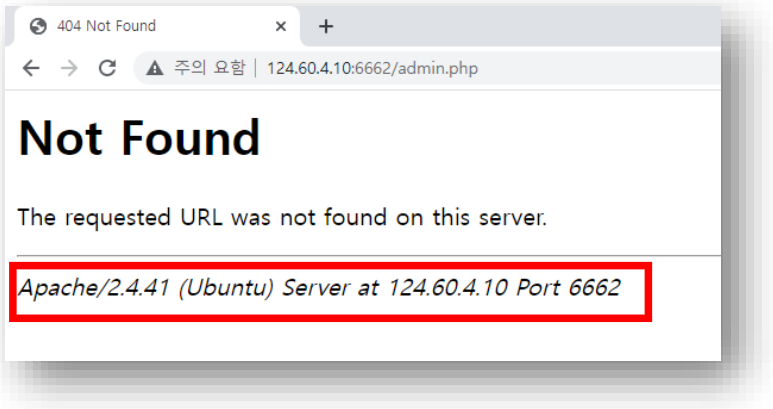

그림 7 아파치 버전 정보 및 서버 정보 노출

/admin.php 등 쉽게 공격할 수 있는 경로로 들어가면 Not Found 로 에러 페이지와 http 상태정보를 제공한다. 동시에 아파치 웹 서버 버전 정보와 서버 ip, port 가 노출된다.

4.3 조치 방안

1. http.conf 파일 내에 정보가 노출되는 내용을 아래와 같이 수정

```
ServerTokens Prod
ServerSignature Off
```

출처 : https://grooveshark.tistory.com/260

5 파일 업로드 취약점

5.1 취약점 개요

파일 업로드 기능이 존재하는 웹 사이트의 확장자 필터링이 미흡할 경우, 공격자가 악성 파일을 업로드하여 시스템을 장악할 수 있는 취약점

5.2 상세 설명

도메인 : http://124.60.4.10:6662

발생 위치 : http://124.60.4.10:6662/up_file

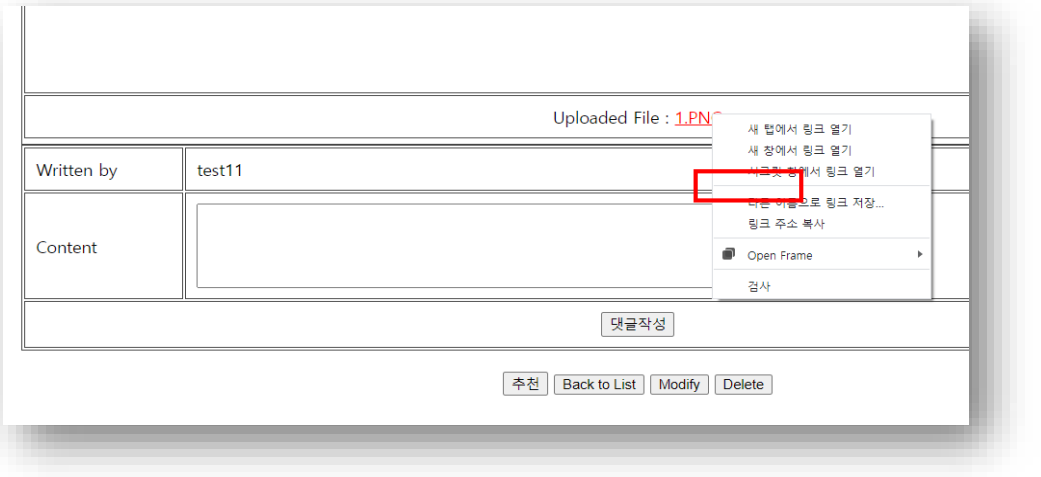

그림 8 다운로드 경로 확인

"1.png" 파일을 업로드한 게시글에서 해당 파일의 링크 주소를 확인한다.

http://124.60.4.10:6662/download.php?file_name=1.PNG

download.php 를 거쳐서 해당 파일을 다운로드받음을 파악한다.

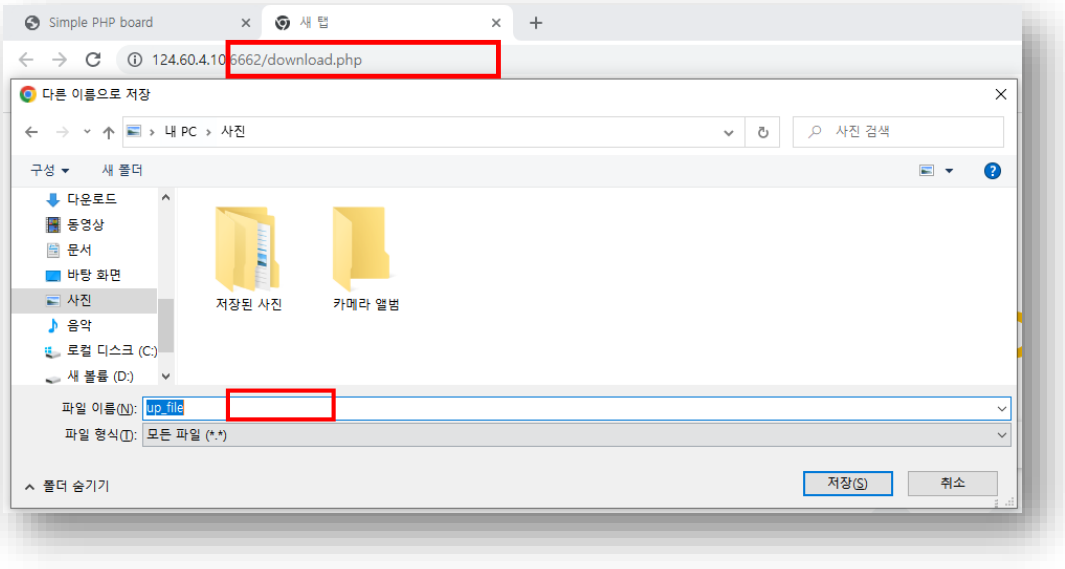

그림 9 download.php 를 통해 다운로드 경로 확인

http://124.60.4.10:6662/download.php 경로로 들어가면 바로 up_file 이라는 어떠한 파일이 다운로드된다. up_file 이 다운로드하는 폴더의 경로라는 것으로 추측가능하다.

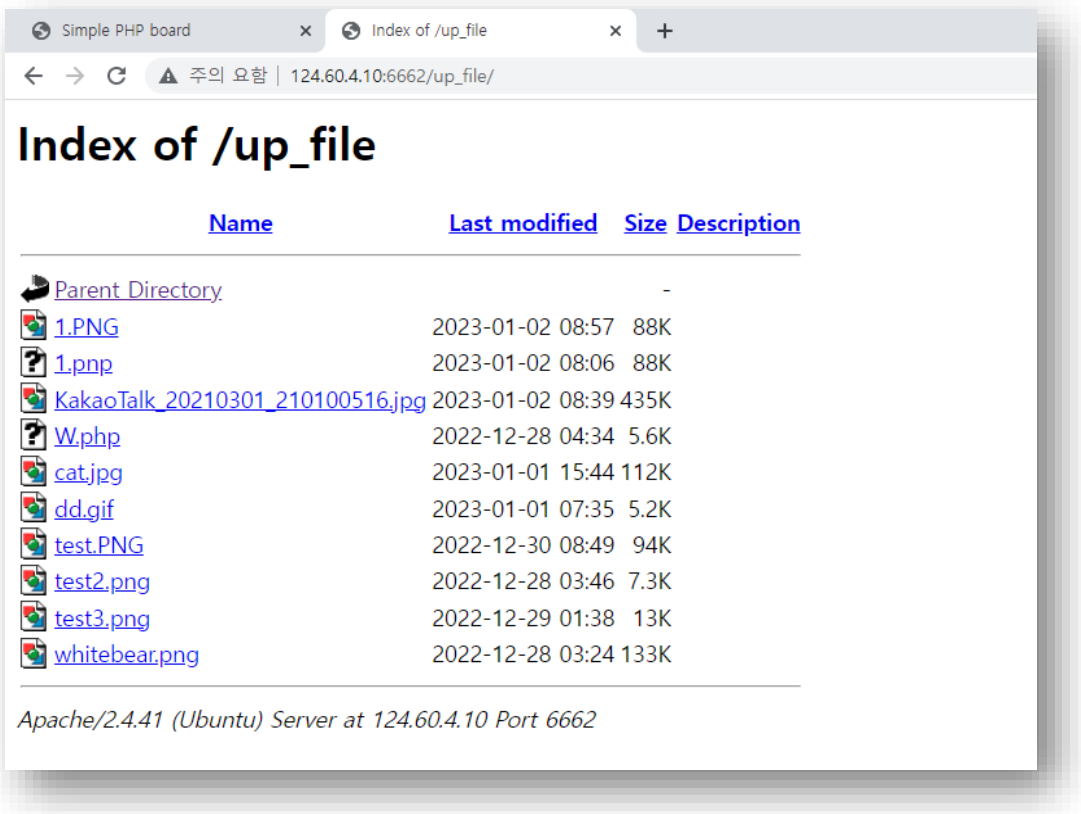

그림 10 up_file 위치로 접근

http://124.60.4.10:6662/up_file 로 접근하면 download.php 에서 업로드된 파일 목록이 존재한다.

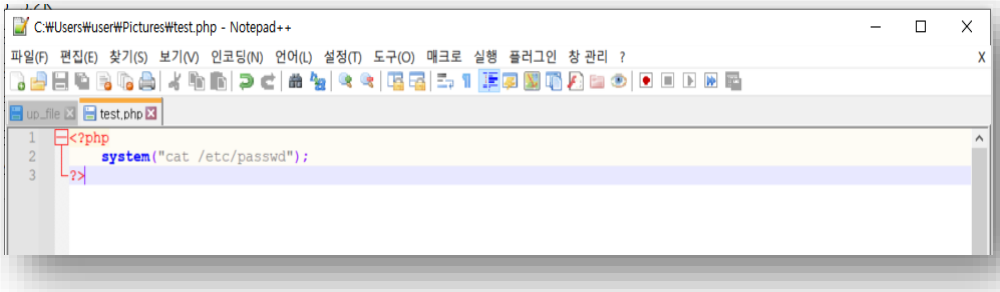

그림 11 정보탈취 php 생성

임의로 /etc/passwd 내용을 출력하게 하는 test.php 파일을 만든 후에 게시판에 업로드한다.

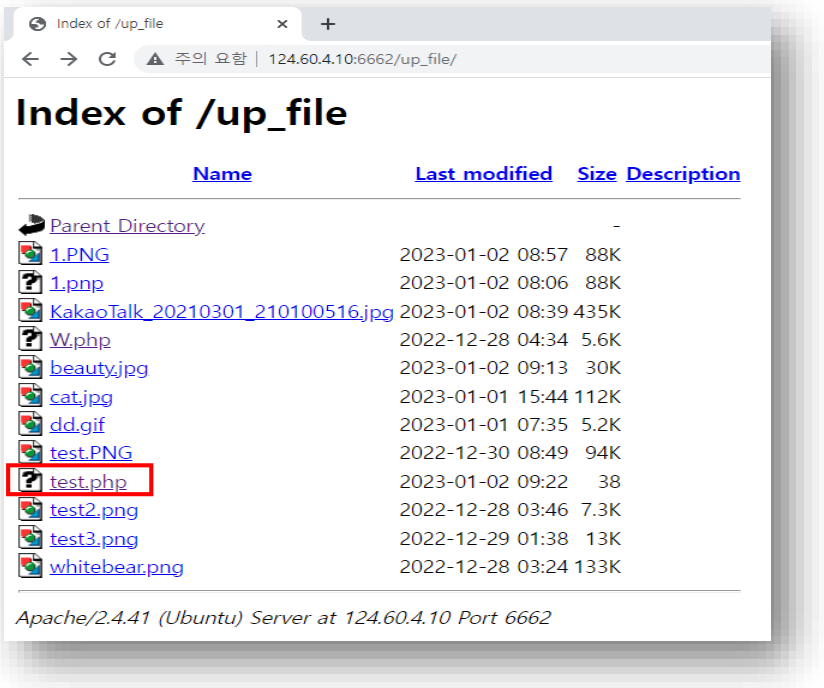

그림 12 test.php 업로드 확인

업로드된 것을 확인하고 실행한다.

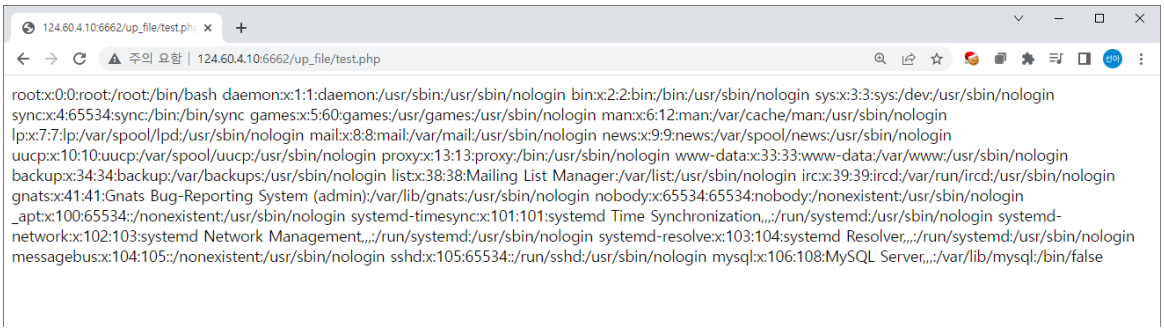

그림 13 /etc/passwd

노출되면 매우 위험한 /etc/passwd 파일의 내용을 확인한다.

5.4 조치 방안

1. http.conf 내용에서 options 항목 뒤에 indexs 단어 삭제
2. 화이트 리스트 방식으로 파일 업로드 가능한 확장자를 png, jpg 등만 허용하도록 설정