2023년 JH 커뮤니티 모의해킹 보고서

2023 / 01 / 06

2023년 JH커뮤니티 모의해킹

**모의해킹 결과 보고서 v1.0**

목차

1. **모의해킹 수행 정보**
   1. **개요**
   2. **대상**
   3. **수행 기간**
   4. **수행 인력**
2. **모의해킹 결과**
   1. **총평**
   2. **결과 요약**
3. **취약점 상세 내용**
   1. **발생 취약점**

**3.1.1 상세 내용**

1. **보안 권고안**
   1. **취약점에 대한 권고**
   2. **시큐어 코딩**
2. **결론**
3. 모의해킹 수행 정보
   1. **개요**

본 프로젝트는 JH커뮤니티의 웹 애플리케이션에 대한 모의 해킹을 수행한 후 발견된 취약점에 대한 대응 방안을 제시함으로써 차후 발생할 수 있는 침해 사고를 예방하고 정보 보호 수준을 향상할 수 있는 대책을 수립하는 데 기여하는 것을 목적으로 합니다.

**1.2. 대상**

본 프로젝트 점검 대상은 다음과 같습니다.

|  |  |  |
| --- | --- | --- |
| No | Platform | URL |
| 1 | **WEB** | **http://normaltic.com:9191/** |

* 1. **수행 기간**
* 2023년 1월 22일 ~ 2023년 1월 25일
  1. **수행 인력**

|  |  |  |
| --- | --- | --- |
| 소속 | 성명 | 담당 업무 |
| SegFault | **도원준** | **- 웹 모의해킹** |

1. 모의해킹 결과

**2.1. 총평**

**해당 웹 서비스는 전체적으로 시큐어 코딩이 되어 있어 안전한 편 입니다. 그러나**

**시큐어 코딩 및 필터링이 되어있지 않은 페이지가 몇몇 존재하며, 시큐어 코딩이 된**

**페이지도 필터링이 미흡한 부분이 있습니다. 때문에 웹 페이지 중 시큐어 코딩이**

**되어있지 않은 페이지를 시큐어 코딩 해야 하며, 추가로 필터링이 미흡한 페이지의**

**방어를 강화해야합니다.**

* 1. 결과 요약

|  |  |  |
| --- | --- | --- |
| No | 취약점 | 내용 |
| 1 | **파일 다운로드** | **파일 다운로드 기능에서 파일 다운로드**  **취약점이 발생하였습니다.** |
| 2 | **크로스 사이트**  **스크립팅** | **게시판 글 읽기 기능에서 XSS 취약점이**  **발생하였습니다.** |
| 3 | **SQL 인젝션** | **게시판 글 검색 기능, 회원정보 수정 페이지에서 SQL 인젝션 취약점이 발생하였습니다** |
| 4 | **디렉터리 인덱싱** | **비회원 문의 게시판에서 디렉터리 인덱싱 취약점이 발생했습니다.** |
| 5 | **정보누출** | **정보 누출 취약점이 발생했습니다.** |
| 6 | **데이터 평문 전송** | **데이터 평문 전송 취약점이 발생했습니다.** |
| 7 | **약한 문자열 강도** | **회원 가입 시 보안 강도가 약한 문자열을**  **패스워드로 사용 가능합니다.** |
| 8 | **크로스사이트**  **리퀘스트 변조(CSRF)** | **회원정보 수정창에서 CSRF취약점이**  **발생했습니다.** |
| 9 | **세션 고정** | **세션 고정 취약점이 발생했습니다.** |
| 10 | **파일 업로드** | **파일 업로드 기능에서**  **파일 업로드 취약점이 발생했습니다.** |
| 11 | **악성 콘텐츠** | **악성 콘텐츠를 서버에 업로드 할 수 있습니다.** |
| 12 | **불충분한**  **세션 만료** | **불충분한 세션 만료 취약점이 발생하였습니다.** |
| 13 | **자동화 공격** | **웹 페이지에 대한 자동화 공격이 가능합니다** |

1. 취약점 상세 내용

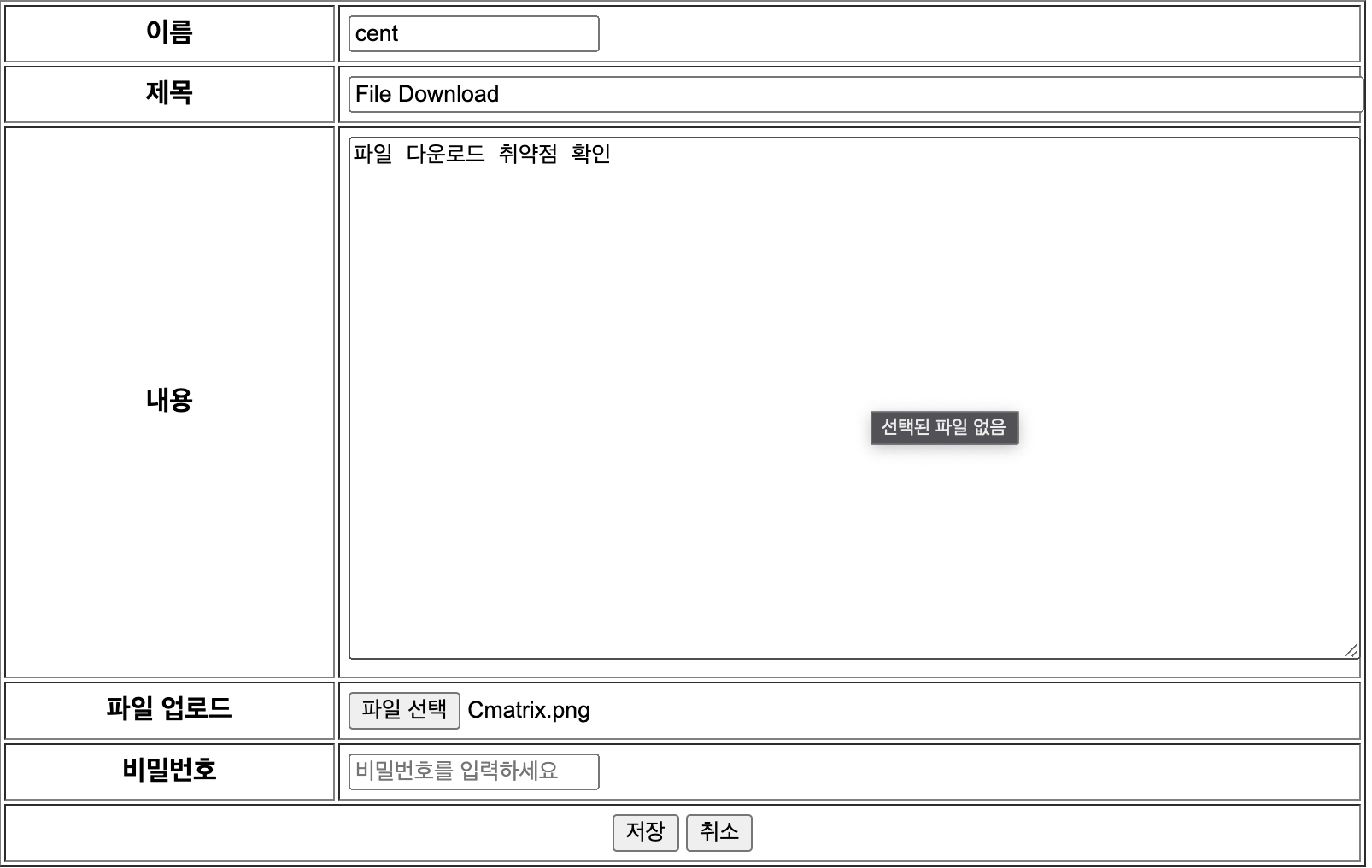
**3.1 File Download(**[**http://normaltic.com:9191/download.php**](http://normaltic.com:9191/download.php)**)**

**3.1.1 File Download**

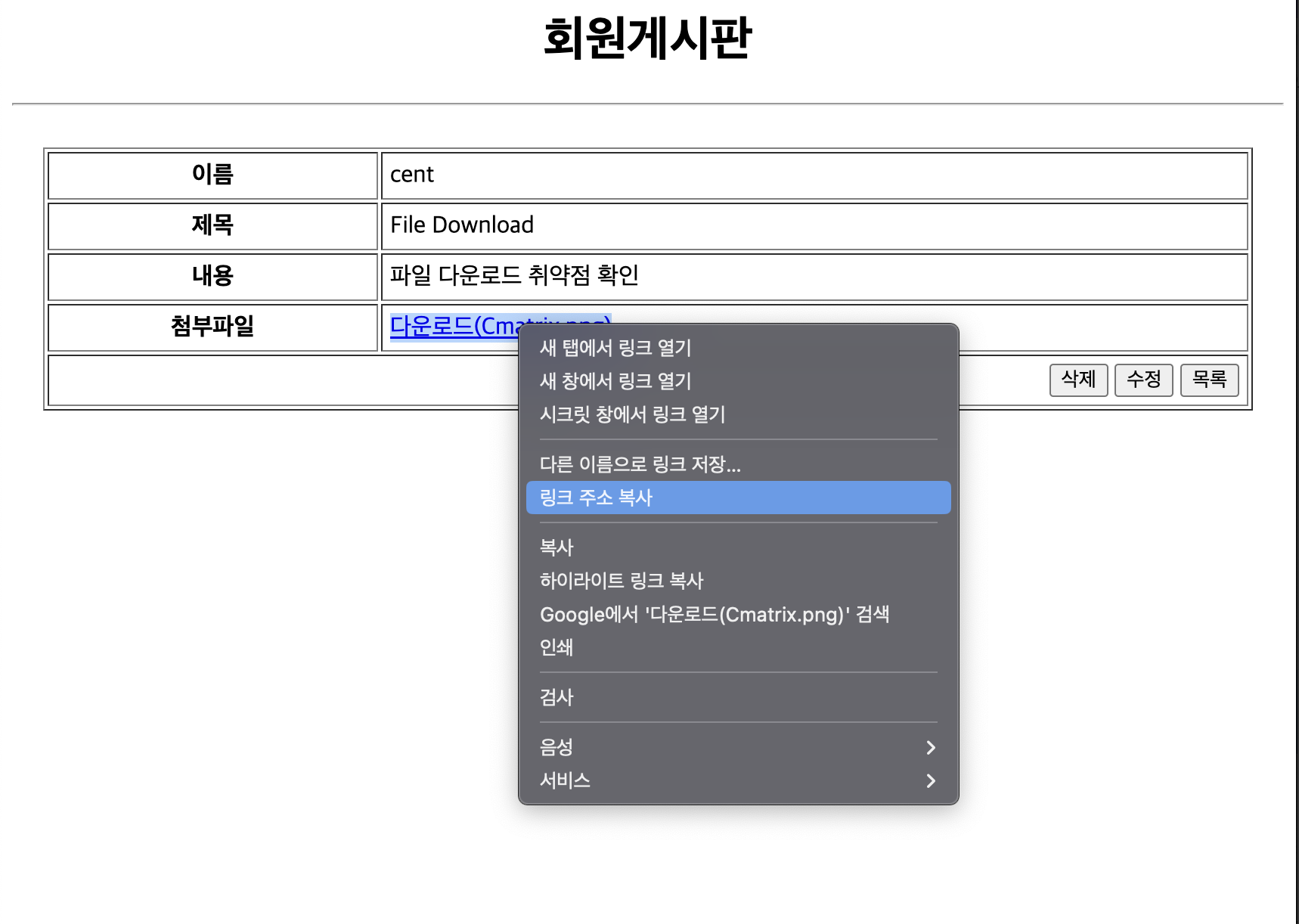
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| http://normaltic.com:9191/download.php | http://normaltic.com:9191/download.php?file=../index.php | 게시판 글 읽기 -> 파일 다운로드 |
| 취약점 상세내역 | 다운로드할 파일에 대한 값을 URL로 받고, 받아들이는 파라미터에 어떠한 필터링도 존재하지 않기에 파일 다운로드 취약점이 발생합니다. | | |

**3.1.2 취약점 확인**

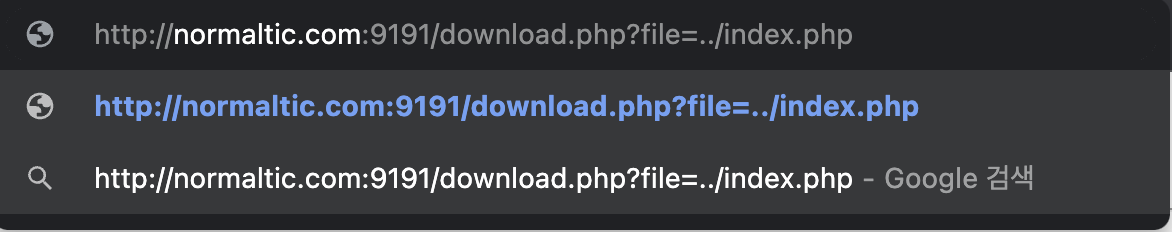
1. **아무 글이나 올리고, 첨부파일을 첨부한다.**

****

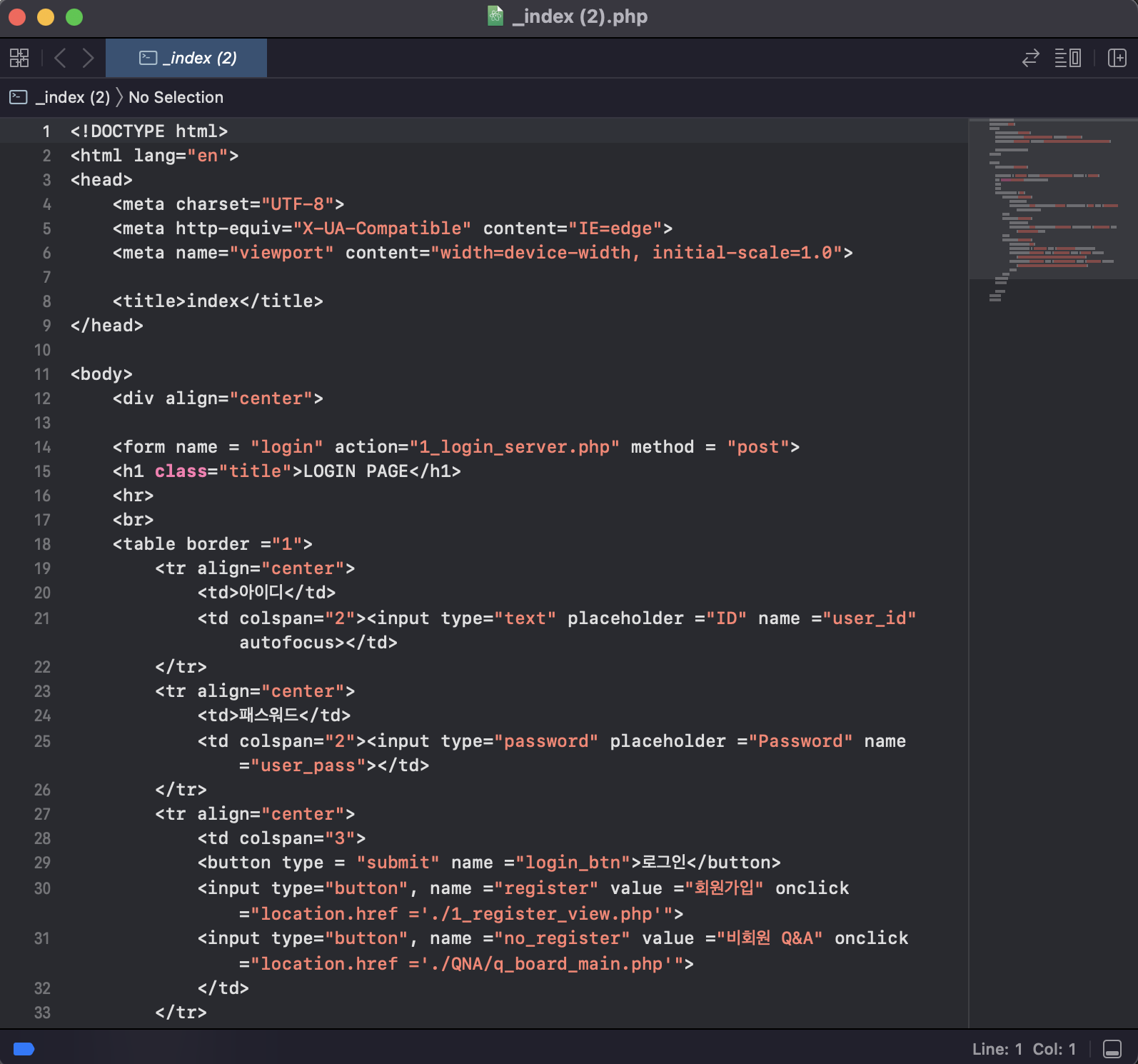
1. **파일 다운로드 링크를 복사한다.**

****

1. **복사한 URL을 붙여넣고 URL을 다음과 같이 수정해준다.**

****

1. **Index.php의 소스코드가 다운로드 된다.**

****

**[보안 권고안]**

**파일을 요청하는 파라미터에서 다른 경로의 파일을 다운로드할 수 있는**

**../ 또는 /, ..와 같은 문자를 필터링한다.**

**3.2 Cross Site Scripting**

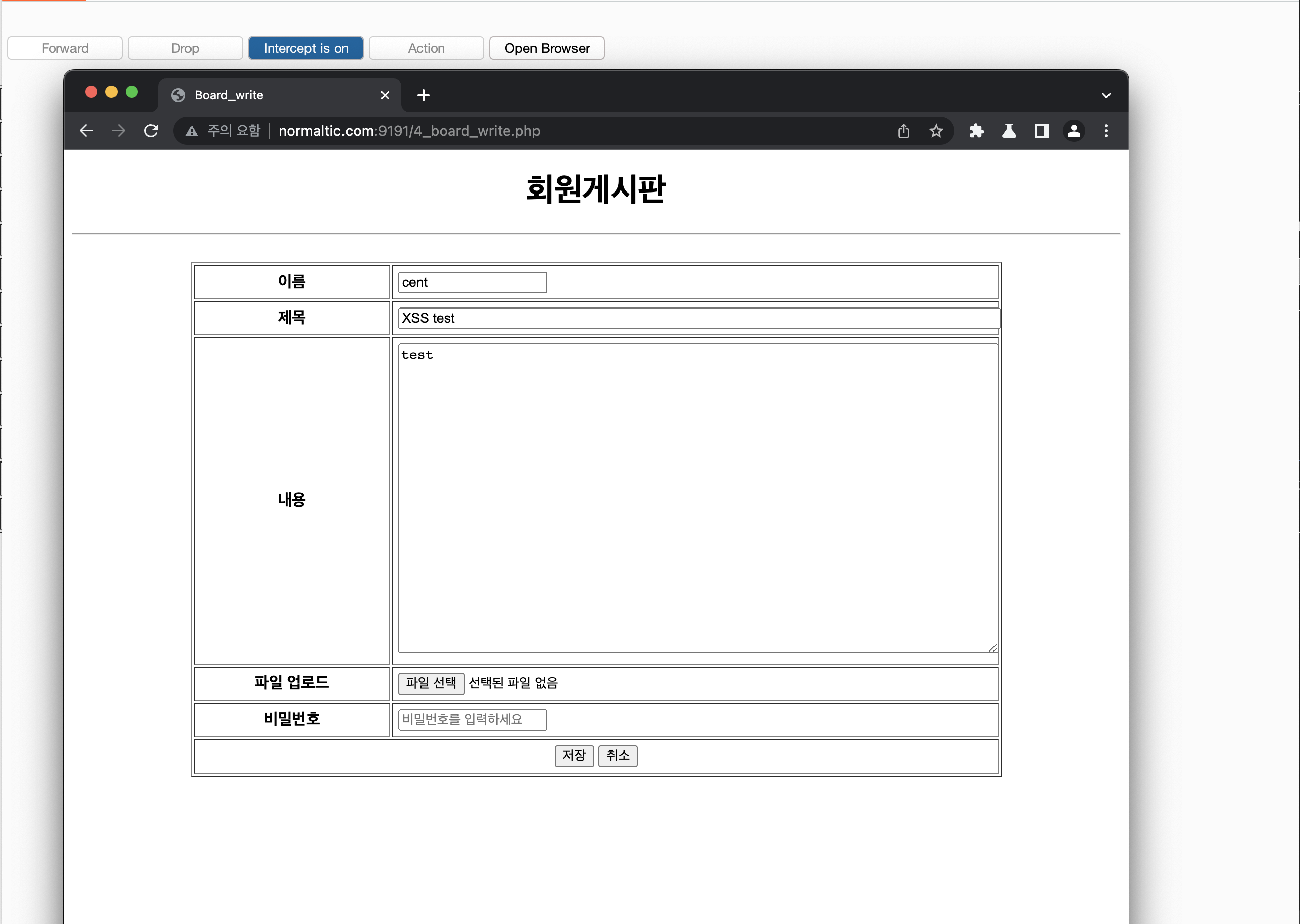
**(http://normaltic.com:9191/4\_board\_main.php)**

**3.2.1 Cross Site Scripting**

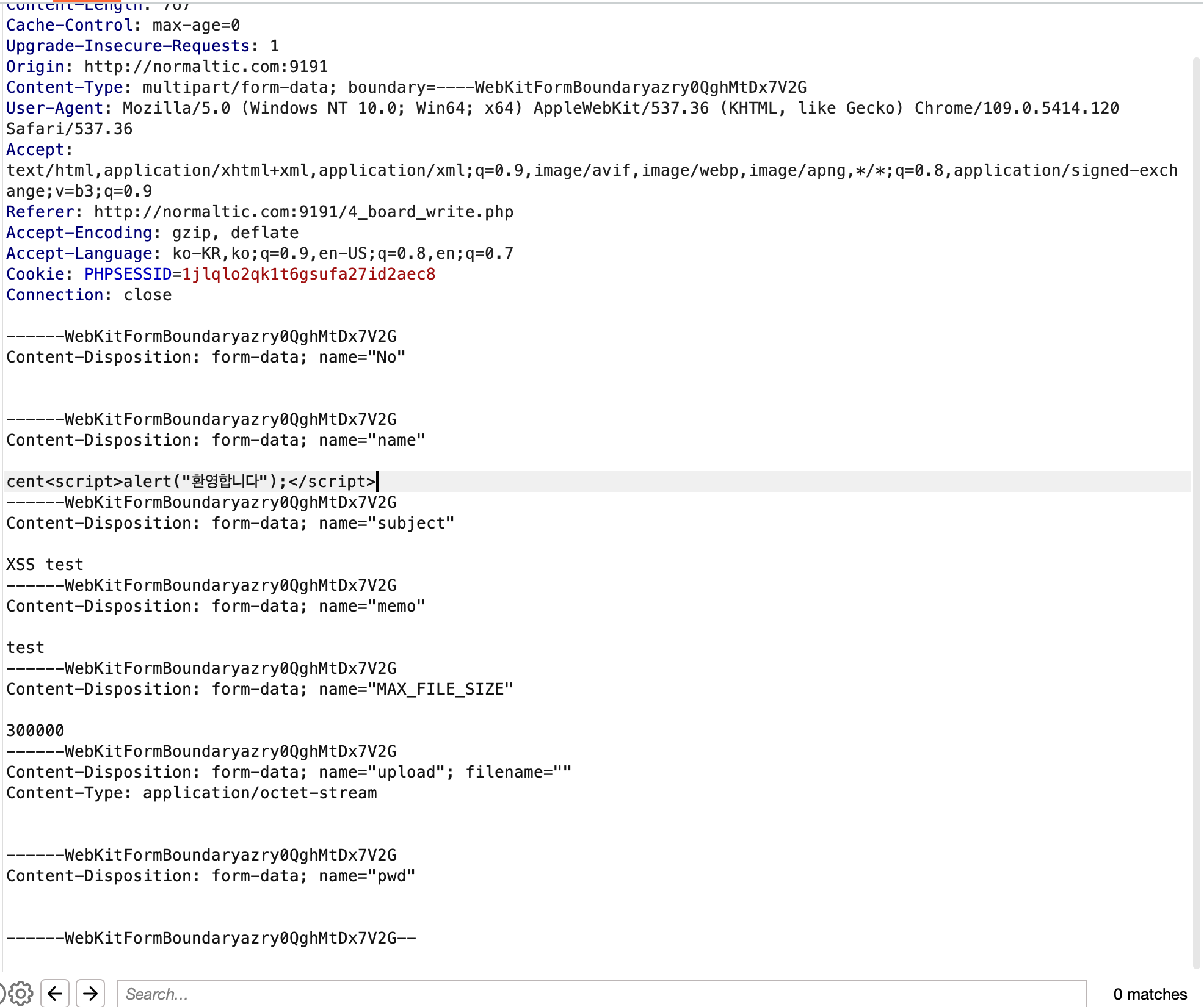
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| http://normaltic.com:9191/4\_board\_main.php | x | 게시판 글 읽기 |
| 취약점 상세내역 | 회원 게시판 기능에서 조회수 , 글의 인덱스 번호에 대한 xss필터링이 이루어지지 않아 xss공격이 가능합니다. | | |

**3.2.2 취약점 확인**

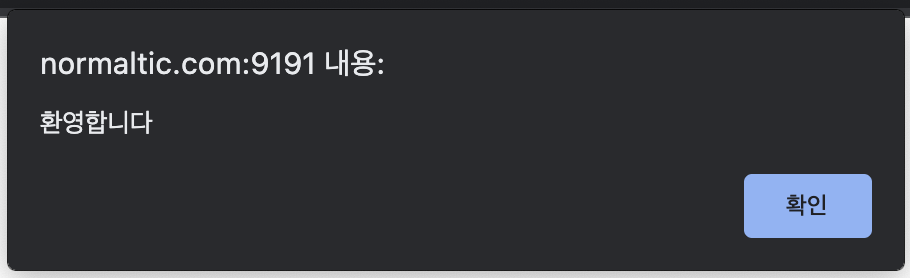
**1. Burp Suite에서 proxy기능을 켠 상태로 글을 작성한다.**

****

**2. name파라미터에 xss구문을 삽입한다.**

****

**3. xss구문 작동 확인**

****

**[보안 권고안]**

**글의 제목, 글의 작성자, 파일이름, 날짜, 글의 인덱스 번호에도**

**Htmlspecialchars 필터링을 적용한다.**

**3.3 SQL Injection**

**(**[**http://normaltic.com:9191/search\_result.php**](http://normaltic.com:9191/search_result.php)**)**

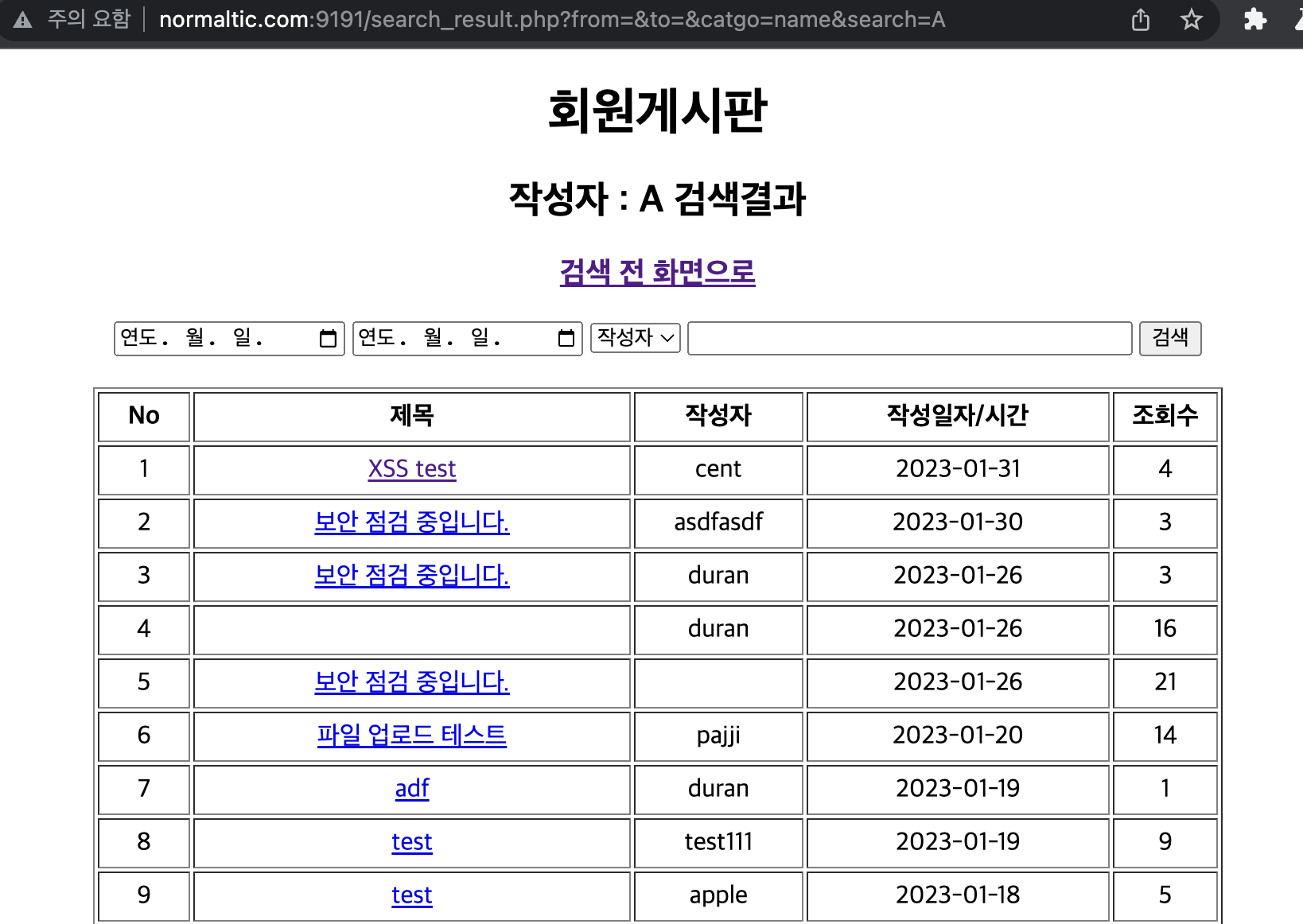
**(http://normaltic.com:9191/3\_mypage\_view.php)**

**3.3.1 SQL Injection**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **(**[**http://normaltic.com:9191/search\_result.php**](http://normaltic.com:9191/search_result.php)**)**  **(http://normaltic.com:9191/3\_mypage\_view.php)** | http://normaltic.com:9191/search\_result.php?from=&to=&catgo=name&search=%f0+%27 | 게시판 글 검색 및  회원 정보 수정 |
| 취약점 상세내역 | 게시판 글 검색 기능에서 SQL Injection에 대한 미약한 필터링,  회원 정보 수정에서 SQL Injection에 대한 필터링이 이루어지지 않아  SQL Injection 취약점이 발생하였습니다. | | |

**3.3.2 취약점 확인**

**1. 게시판 글 검색 기능에 아무 문자나 삽입한다.**

****

**2. search의 파라미터를 %f0+’로 조작한다.**

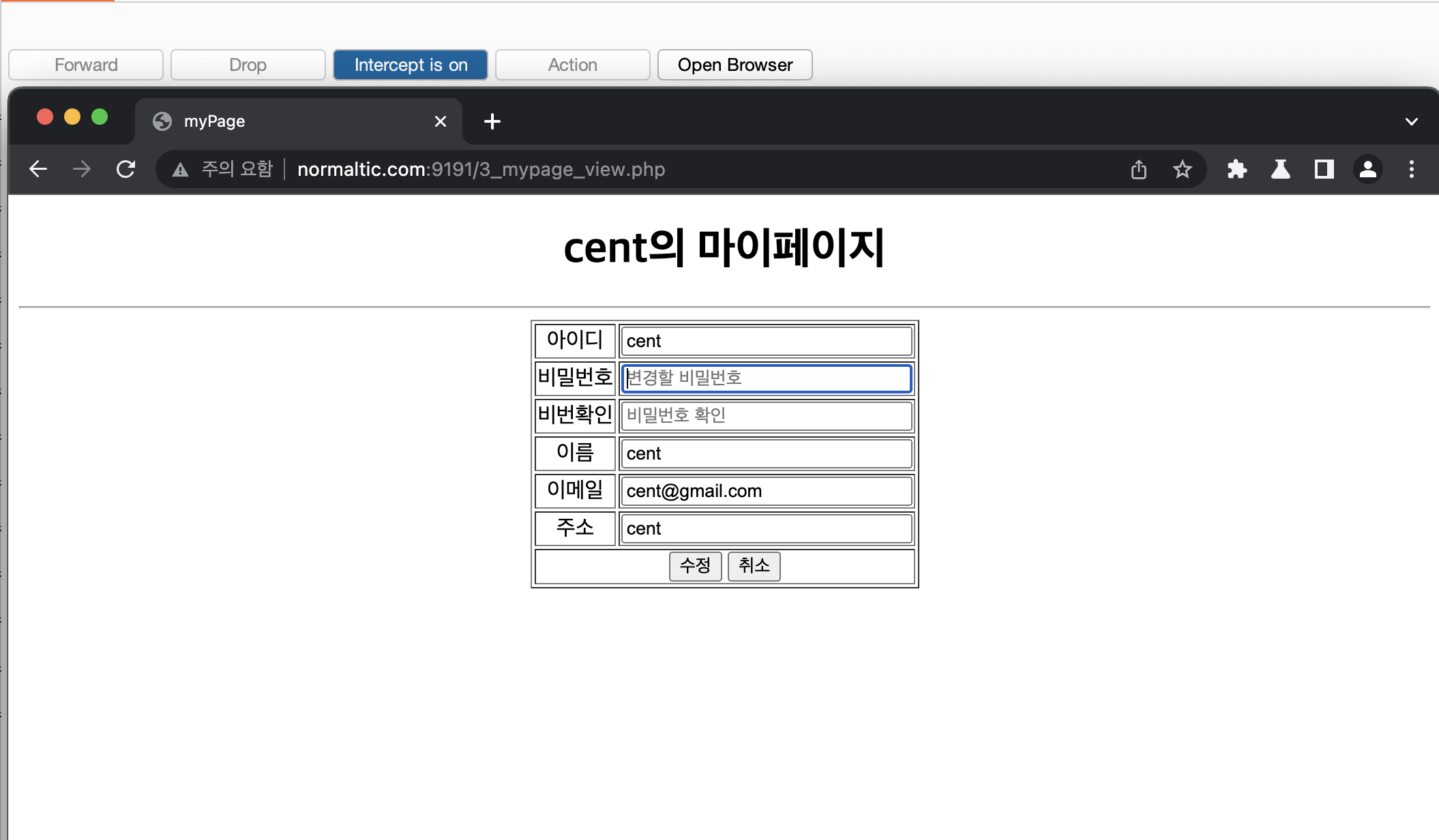
****

**3. ‘는 원래 \’로 필터링 되어야 함에도 필터링 되지 않고 SQL 구문에 삽입된다.**

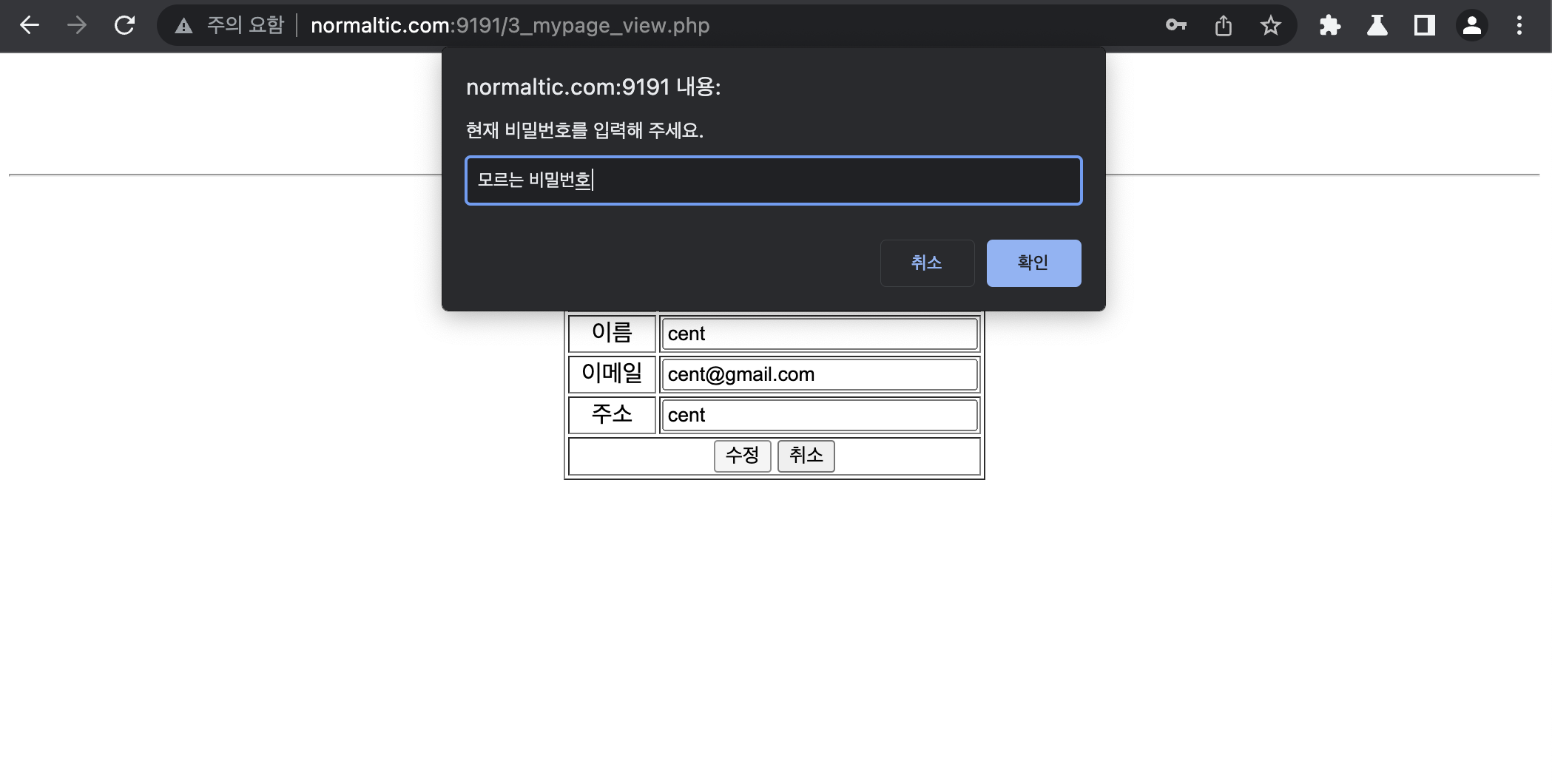
****

**3.3.2 취약점 확인\_2**

**1. 로그인 후 BurpSuite Proxy기능을 켠 상태로 마이페이지에 접속한다.**

****

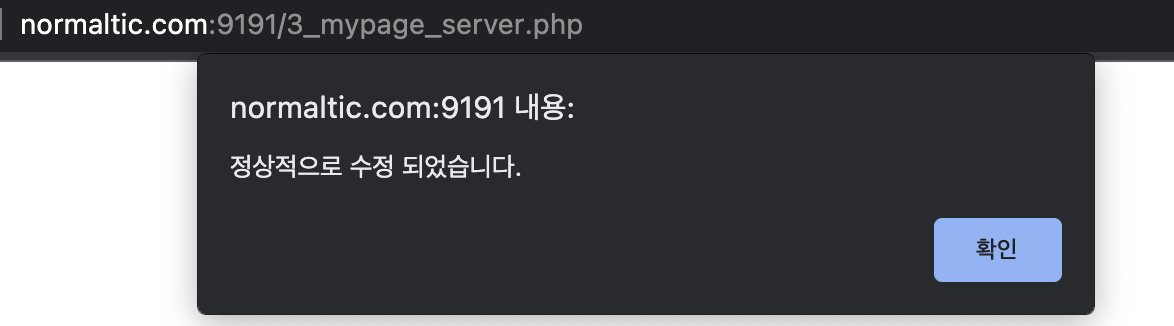
**2. 변경할 비밀번호를 적고 , 현재 비밀번호에 아무 값이나 적은 뒤 전송한다.**

****

**3. m\_id의 변수를 아이디‘#으로 수정한다.**

****

**4.현재 비밀번호가 일치하지 않아도 비밀번호가 정상적으로 수정된다.**

****

**[보안 권고안]**

**게시판 글 검색 기능의Search 파라미터를 GET이 아닌 POST방식으로 넘겨 받는다.**

**회원 정보 변경 페이지에서 넘겨받는 모든 변수의 특수문자를 필터링하고,**

**Prepared Statement를 적용한다.**

**3.4 디렉터리 인덱싱**

**3.4.1 디렉터리 인덱싱**

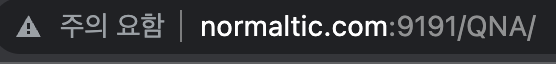
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/QNA/** | http://normaltic.com:9191/QNA/ | 비회원 문의 게시판 |
| 취약점 상세내역 | 비회원 문의 게시판 경로인 /QNA/안에 어떤 파일이 존재하는지 그대로 드러난다. | | |

**3.4.2 취약점 확인**

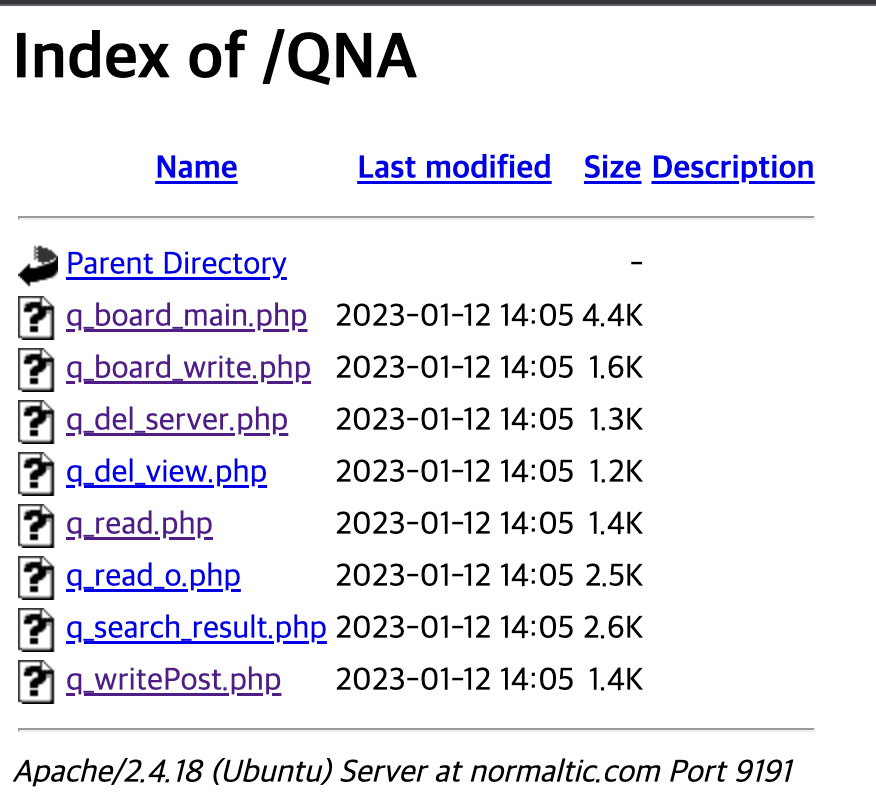
**1. 비회원 문의 게시판에 접속한다.**

****

**2. URL에서 q\_board\_main.php를 제거한다.**

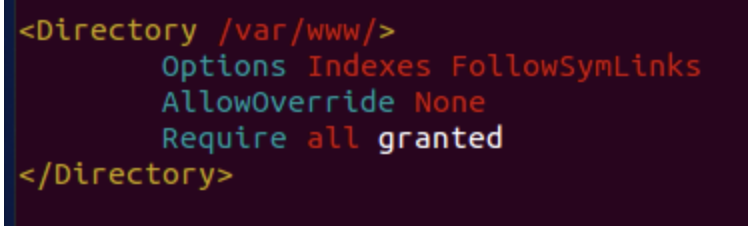
****

**3. 디렉터리 인덱싱 취약점 확인**

****

**[보안 권고안]**

**Vim 편집기를 통해 /etc/apache2/apache2.conf에서**

****

**해당 옵션의 Indexes를 제거한다.**

**3.5 정보누출**

**3.5.1 정보누출**

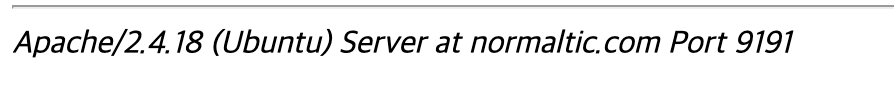
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/x** | **http://normaltic.com:9191/x** | 아무 경로 |
| 취약점 상세내역 | 서버 주소에 접속하고, 존재하지 않는 파일을 요청하면 서버의 정보가 그대로  드러난다. | | |

**3.5.2 취약점 확인**

**1. 사이트에 접속 후, URL을 통해 존재하지 않는 파일을 요청한다.**

****

**2. 서버의 정보가 그대로 드러난다.**

****

**[보안 권고안]**

**Httpd.conf파일에서 ServerSignature 옵션을 off로 변경하고**

**ServerTokens Prod를 적용시킨다.**

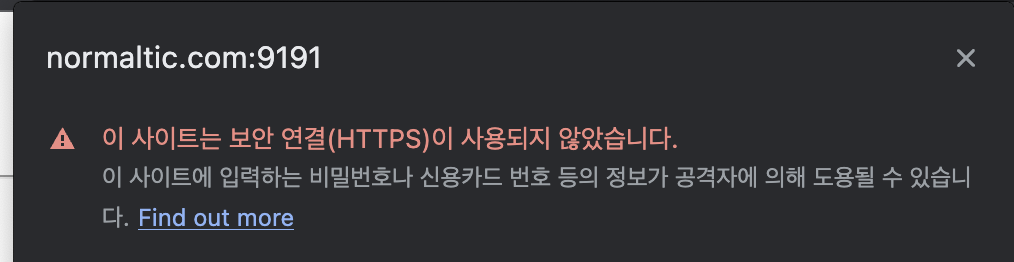
**3.6 데이터 평문 전송**

**3.6.1 데이터 평문 전송**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/** | **x** | 서버 전체 |
| 취약점 상세내역 | 서버에서 전송되는 모든 값들이 함호화되지 않고 평문으로 전송된다. | | |

**3.6.2 취약점 확인**

**1. 서버가 http를 통해 통신하는 것을 확인할 수 있다.**

****

**[보안 권고안**

**서버 통신 프로토콜을 HTTPS로 변경한다.**

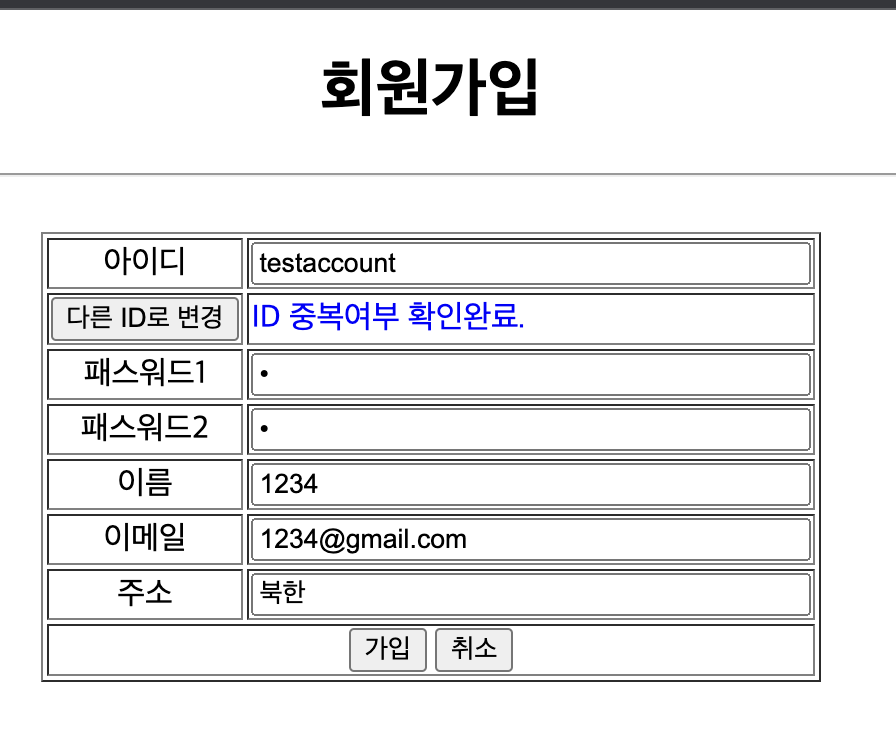
**3.7 약한 문자열 강도**

**3.7.1 약한 문자열 강도**

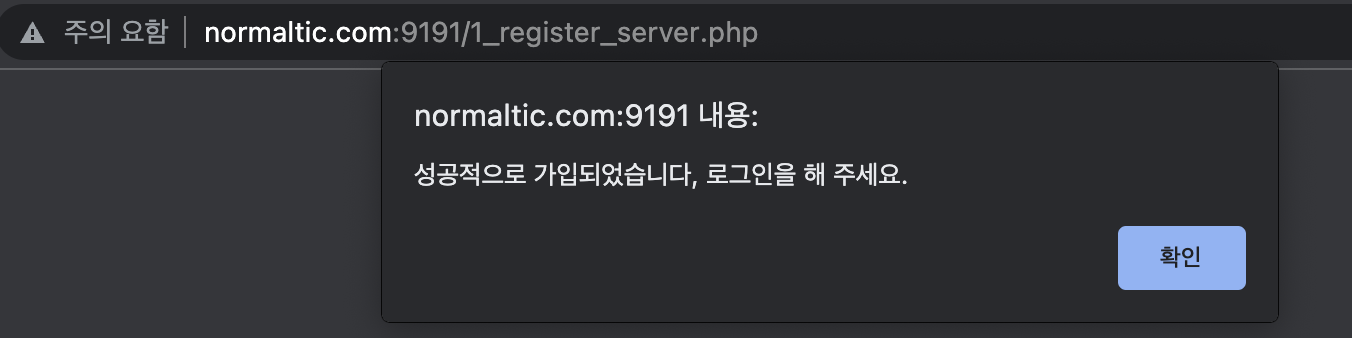
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/1\_register\_view.php** | **x** | 회원가입 페이지 |
| 취약점 상세내역 | 비밀번호 보안강도를 매우 약하게 설정할 수 있다. | | |

**3.7.2 취약점 확인**

**1. 회원가입 시 비밀번호를 매우 짧게 설정한다.**

****

**2. 회원가입에 성공한다.**

****

**[보안 권고안]**

**패스워드 조건을 보안강도가 강하도록 설정한다.**

**일정 보안 강도를 만족시키지 못하면 회원가입을 하지 못하도록 한다.**

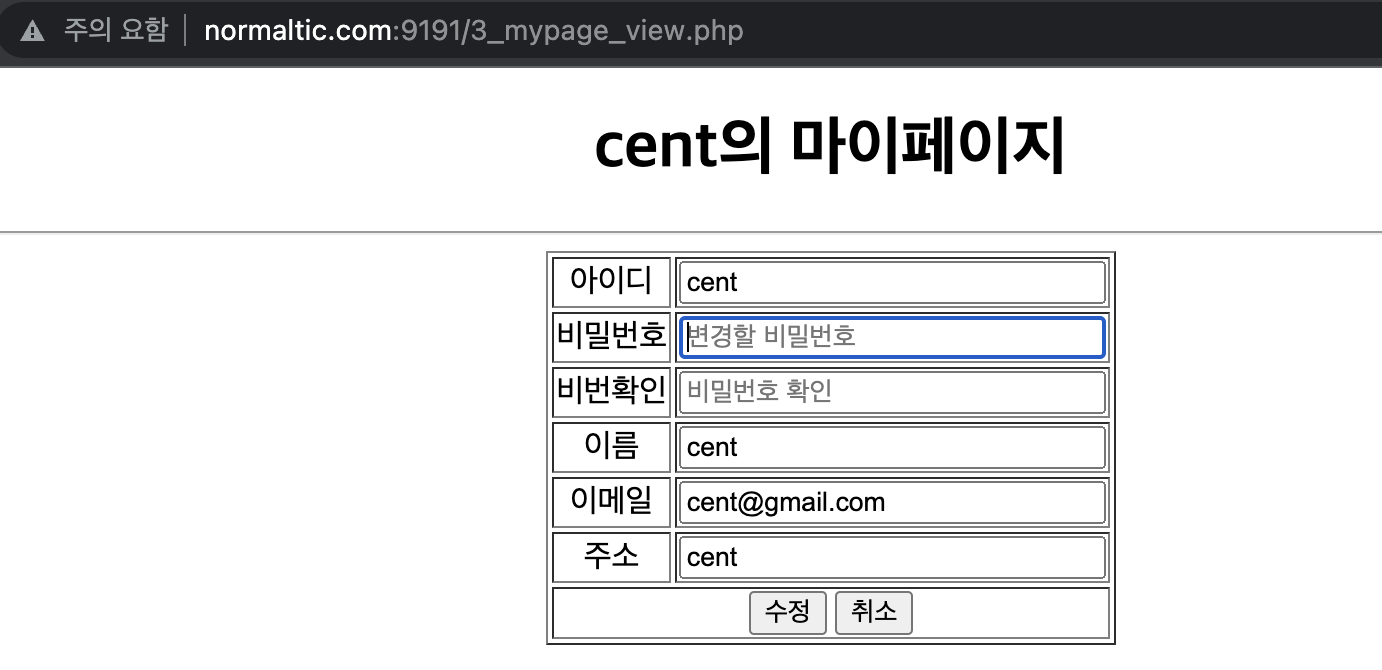
**3.8 크로스 사이트 리퀘스트 변조(CSRF)**

**3.8.1 크로스 사이트 리퀘스트 변조(CSRF)**

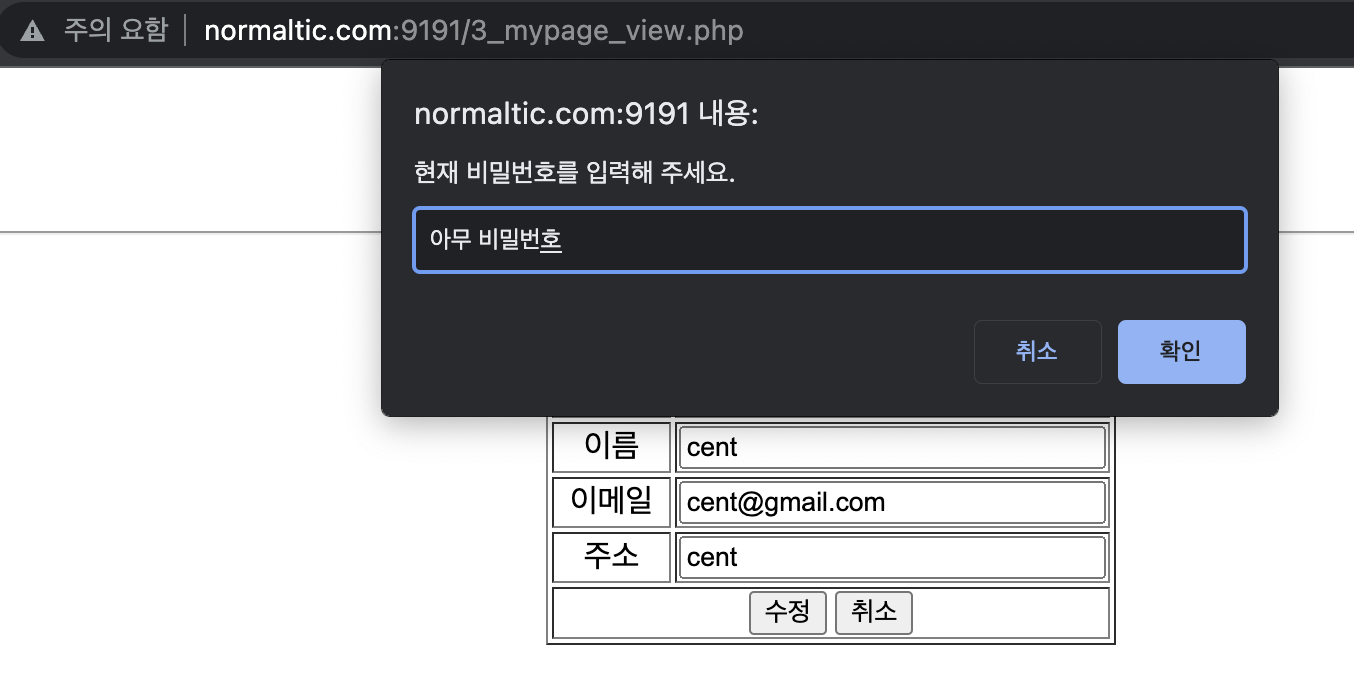
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/3\_mypage\_server.php** | **x** | 회원 정보 수정  페이지 |
| 취약점 상세내역 | 내가 아닌 타인의 회원 정보를 수정할 수 있다. | | |

**3.8.2 취약점 확인**

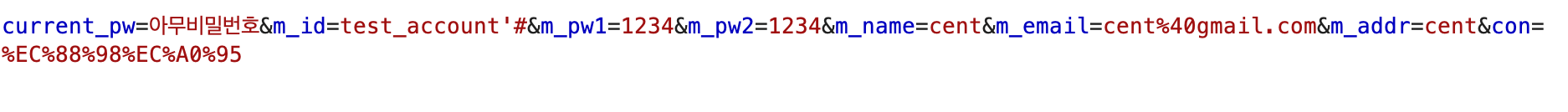
**1. BurpSuite proxy를 켠 상태로 로그인 후, 마이페이지에 접속한다.**

****

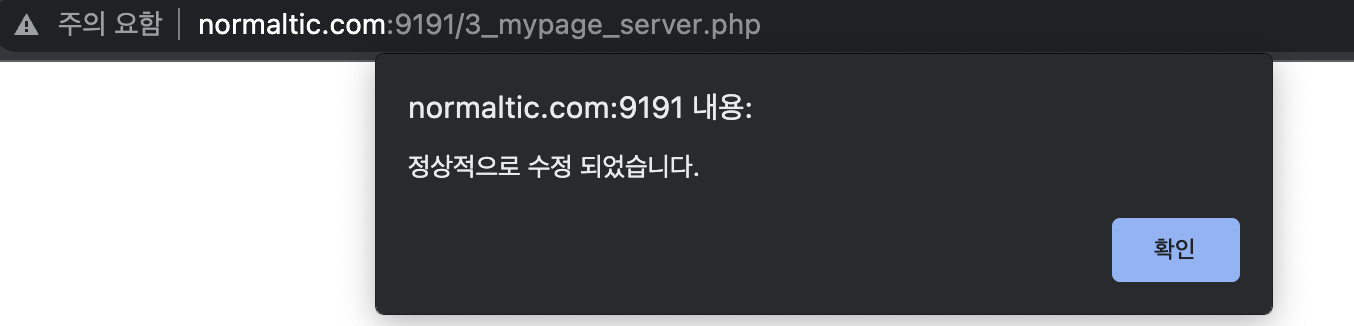
**2. 변경하고 싶은 항목들을 삽입하고, 수정을 누른 후 아무 비밀번호나 입력한다.**

****

**3. m\_id 파라미터를 바꾸고 싶은 계정의 아이디’#으로 수정한 후 값을 전송한다.**

****

**4. 현재 비밀번호가 틀렸음에도 회원정보가 정상적으로 수정된다.**

****

**[보안 권고안]**

**회원 정보 수정 페이지에서 넘겨받는 모든 파라미터에 대한 특수문자 필터링을 적용시킨다.**

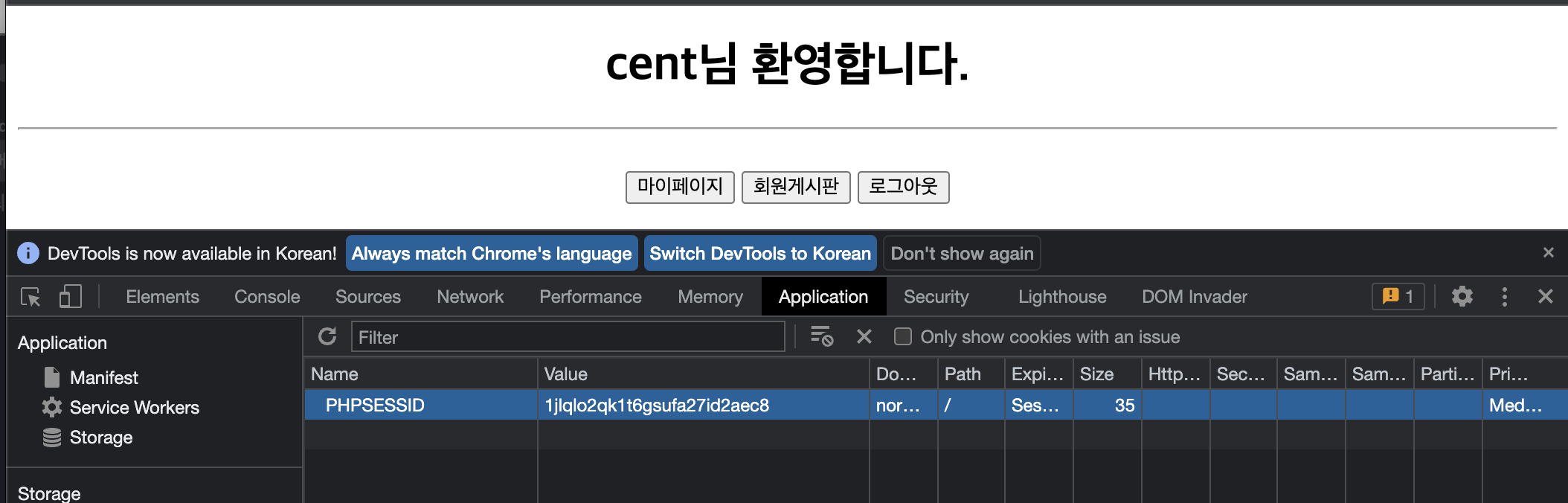
**3.9 세션 고정**

**3.9.1 세션 고정**

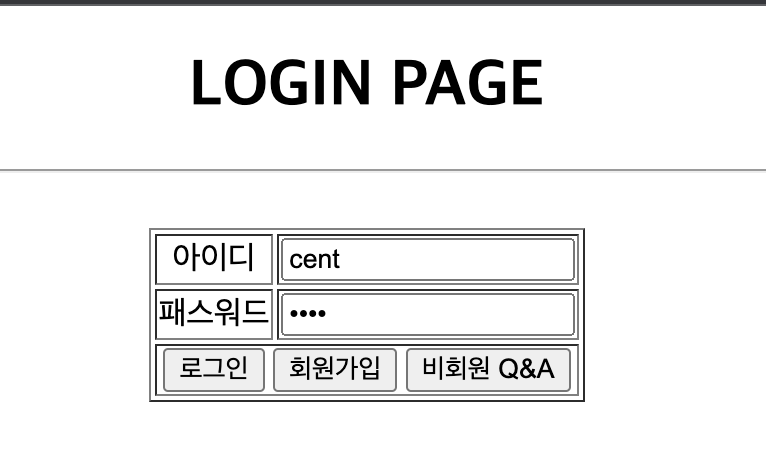
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/** | **x** | 서버 전체 |
| 취약점 상세내역 | 서버에서 발급 받는 세션이 고정된다. | | |

**3.9.2 취약점 확인**

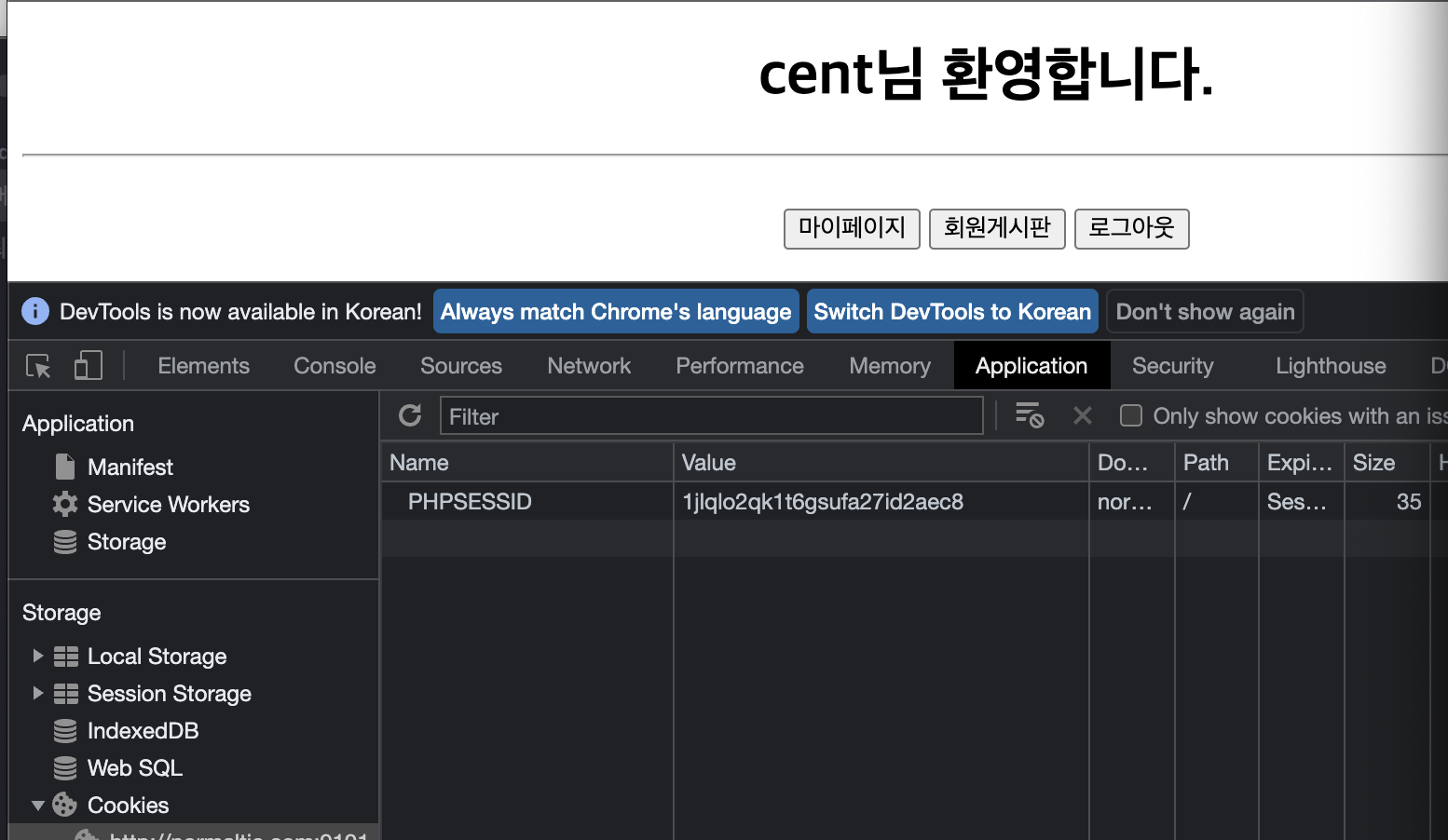
**1. 로그인을 한 후 세션을 확인한다.**

****

**2. 로그아웃 후 재로그인을 한다.**

****

**3. 세션이 고정된 것을 확인할 수 있다.**

****

**[보안 권고안]**

**로그인을 새로 할 때마다 매번 새로운 세션ID를 발급하도록 한다.**

**3.10 파일 업로드**

**3.10.1 파일 업로드**

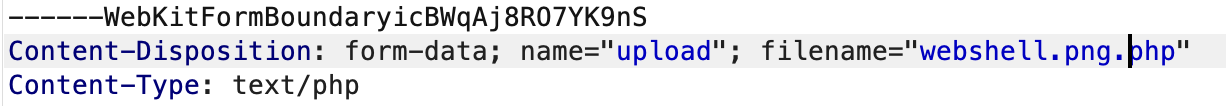
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/4\_board\_write.php** | **x** | 게시판 글 작성 -> 파일 업로드 기능 |
| 취약점 상세내역 | 서버에 확장자와 관계 없이 아무 파일이나 업로드가 가능하다. | | |

**3.10.2 취약점 확인**

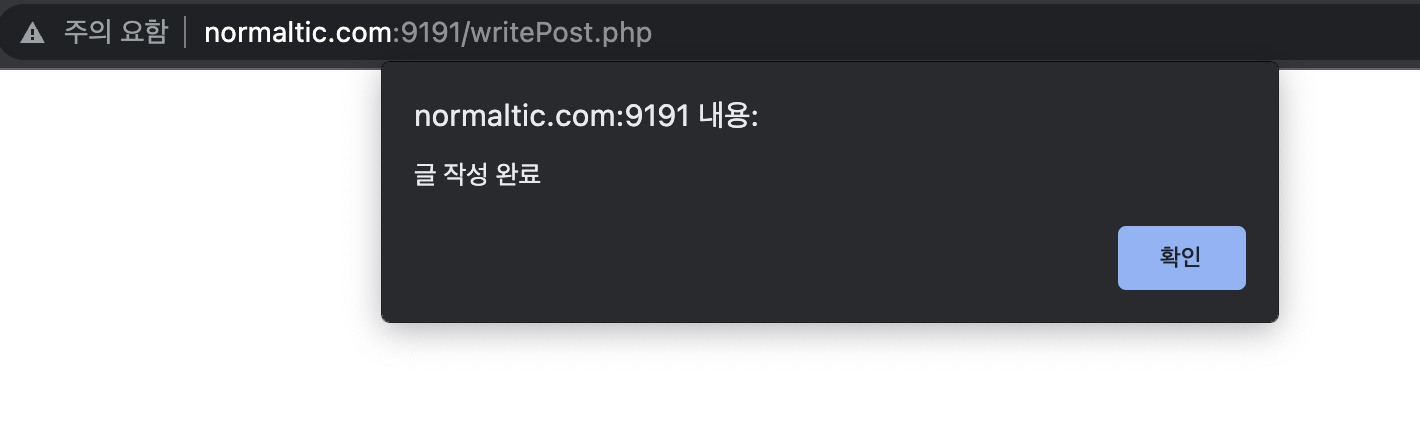
**1. BurpSuite의 proxy기능을 켠 상태로 php파일을 첨부해 글을 작성한다.**

****

**2. 파일 이름을 파일명.png.php로 수정한다.**

****

**3. webshell 파일이 업로드 된다.**

****

**[보안 권고안]**

**파일명을 파라미터로 받아 jpg, png등의 화이트리스트를 만들어 이외의 다른 확장자는 모두 거부하도록 한다.**

**3.11 악성 콘텐츠**

**3.11.1 악성 콘텐츠**

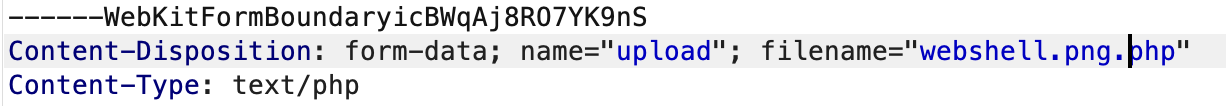
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/4\_board\_write.php** | **x** | 게시판 글 작성 -> 파일 업로드 기능 |
| 취약점 상세내역 | 서버에 악성 콘텐츠를 업로드 할 수 있다. | | |

**3.11.2 취약점 확인**

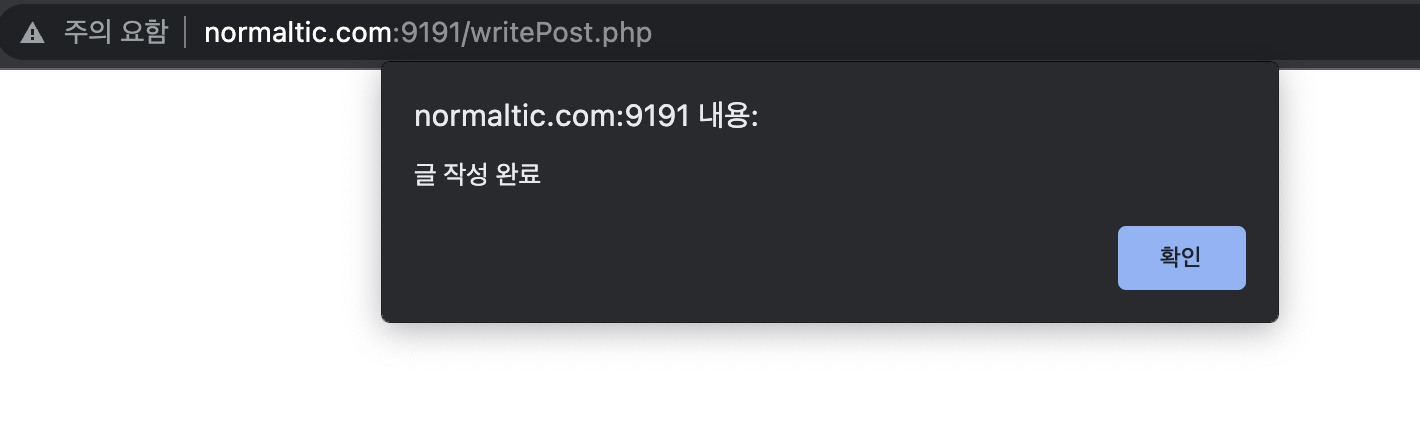
**1. BurpSuite의 proxy기능을 켠 상태로 php파일을 첨부해 글을 작성한다.**

****

**2. 파일 이름을 파일명.png.php로 수정한다.**

****

**3. 악성 콘텐츠 파일이 업로드 된다.**

****

**이외에도 같은 방법을 이용해 exe , cmd , bat등의 파일을 올릴 수 있다.**

**[보안 권고안]**

**파일명을 파라미터로 받아 jpg, png등의 화이트리스트를 만들어 이외의 다른 확장자는 모두 거부하도록 한다.**

**3.12 불충분한 세션 만료**

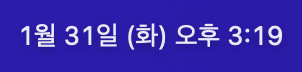
**3.12.1 불충분한 세션 만료**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/** | **x** | 서버 전체 |
| 취약점 상세내역 | 한번 발급받은 세션이 긴 시간이 지나도 사라지지 않는다 | | |

**3.12.2 취약점 확인**

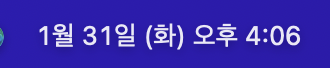
**1. 로그인을 해 세션을 발급 받는다.**

****

****

**2. 30분 정도가 지나도 세션이 사라지지 않는 것을 확인할 수 있다.**

****

****

**3.13 자동화 공격**

**3.13.1 자동화 공격**

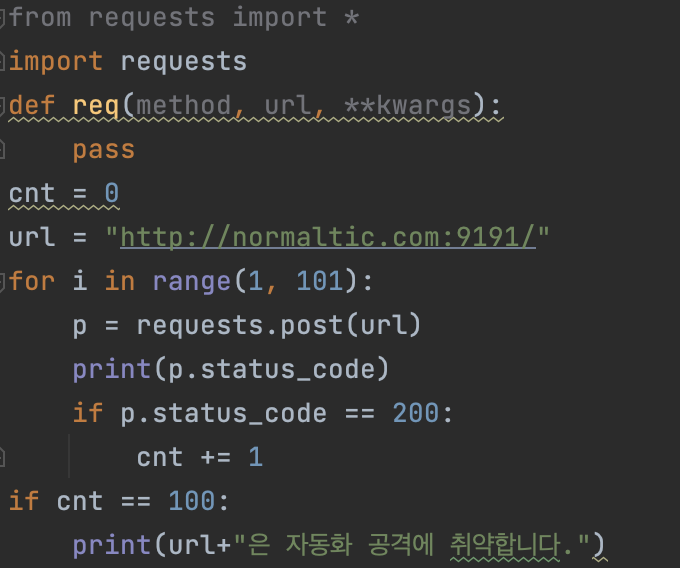
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9191/** | **x** | 서버 전체 |
| 취약점 상세내역 | 서버에 대한 자동화 공격이 가능하다. | | |

**판단 기준 : 반복적인 웹 페이지 요청에 대한 대응**

**3.12.2 취약점 확인**

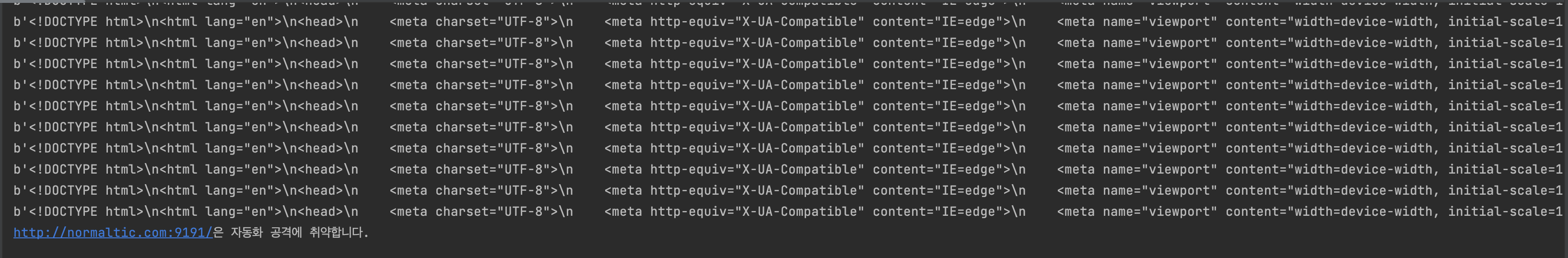
**1. 연속적인 요청을 날리는 Poc코드를 설계해서 웹 페이지에 대량의 요청을**

**보낸다.**

****

**2. 100번의 연속적인 요청에도 불구하고, 계속해서 동일한 페이지를 보내주는**

**것을 확인할 수 있다.**

****

**[보안 권고안]**

**동일한 ip주소에 같은 요청이 일정 시간안에 과도하게 들어오면, 해당 ip가 봇인지 사람인지 검사를 하는 로직을 설계하도록 한다.**

1. 보안 권고안

**4.1 File Download**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | File Download | |
| 취약점 개요 | 파일 다운로드 경로가 그대로 URL에 표시되는 웹 사이트의 경우,  URL을 수정하여 간단하게 서버 코드 파일을 다운로드할 수 있으므로,  URL로 넘겨받는 파라미터에도 필터링을 적용시켜야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법  1. URL로 넘겨받은 파라미터에 대해 .. | ../ | /를 필터링한다. | |

**4.2 크로스 사이트 스크립팅**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 크로스 사이트 스크립팅 | |
| 취약점 개요 | 사용자가 입력한 값이 그대로 다시 화면에 표기되는 웹 페이지는 크로스 사이트 스크립팅 공격이 발생할 확률이 매우 높으므로, 반드시 필터링이 이루어져야  한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 사용자로부터 파라미터를 넘겨받을 때, 모든 파라미터에 대해 Htmlspecialchars를 통해 특수문자 처리를 해준다.  2. 특수문자 처리가 된 파라미터를 다시 서버로 넘겨준다. | |

**4.3 SQL Injection**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | SQL Injection | |
| 취약점 개요 | 사용자가 입력한 값이 그대로 서버의 SQL Query문에 삽입되는 경우, 매우 높은  확률로 SQL Injection이 발생할 수 있다. 그러므로 사용자로부터 넘겨받은 변수를  먼저 특수문자 필터링을 해준 후 SQL Query문에 대입하거나, 또는 SQL Query문을 먼저 컴파일 해준 이후 값을 삽입해야 하는 Prepared Statement기법을 사용해야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 사용자가 입력한 파라미터 값을 mysqli\_real\_escape\_string 함수를 통해 string escapte처리를 해준다.  2. string escape 처리가 완료된 변수를 SQL Query문에 삽입한다. | |

**4.4 디렉터리 인덱싱**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 디렉토리 인덱싱 | |
| 취약점 개요 | 사용자가 특정 디렉토리에 접속 후 아무런 파일을 요청하지 않으면, 서버는  해당 디렉토리 안에 존재하는 모든 파일을 불러오게 된다. 서버가 디렉토리 내부의 파일정보를 넘겨 주지 않도록 설정해야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1.Vim 편집기를 통해 /etc/apache2/apache2.conf를 편집 모드로 연다.  2. Directory /var/www/의 Options에서 Indexes를 제거한다. | |

**4.5 정보누출**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 정보누출 | |
| 취약점 개요 | 서버가 사용자의 브라우저에 오류를 뱉어낼 때 , 따로 설정을 해주지 않으면  서버는 서버의 버전 정보, 서버가 돌아가는 OS 등등 여러 정보를 넘겨주게  된다. 서버가 오류를 뱉어낼 때 이러한 정보를 넘기지 않도록 조취해야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1.Httpd.conf파일을 vim을 통해 편집 모드로 열어준다.  2. ServerSignature 옵션을 off로 변경한다.  3. 그 아래에 ServerTokens Prod를 써준다. | |

**4.6 데이터 평문 전송**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 데이터 평문 전송 | |
| 취약점 개요 | 서버가 HTTP 통신을 사용하면, 서버와 브라우저가 통신할 때 이동하는  데이터들이 모두 평문으로 표기되어 공격자들에게 그대로 드러나게 된다.  서버와 브라우저 사이 이동하는 데이터를 암호화 해야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1.서버가 https 통신을 사용하도록 한다. | |

**4.7 약한 문자열 강도**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 약한 문자열 강도 | |
| 취약점 개요 | 사용자가 계정 패스워드 등을 생성할 때, 보안 강도가 약한 문자열도  허용해주게 되면, Brute Forcing 기법을 통해 사용자 계정이 해킹당할 위험이  있다. 때문에 패스워드 보안 강도를 높게 설정해야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 유저가 회원가입을 할 때, 특정 보안 강도 이상의 패스워드를 생성하지  않으면 회원가입하지 못하도록 한다. | |

**4.8 크로스 사이트 리퀘스트 변조(CSRF)**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 크로스 사이트 리퀘스트 변조(CSRF) | |
| 취약점 개요 | 사용자의 회원 정보를 수정하거나 할 때, 충분한 인증 절차가 이루어지지 않거나  인증 절차를 우회할 수 있으면, 공격자가 서버에 대한 요청을 위조하기 쉽다.  떄문에 사용자의 요청에 대한 인증 절차를 강화해야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 서버가 유저에 대한 요청을 받을 때, 실제 유저만 알 수 있는 정보를 통해  유저가 보낸 정보가 맞는 지 검증을 한다.  2. 유저에게서 받은 파라미터를 모두 mysqli\_real\_escape\_string 함수를 통해  String escape 처리를 해준다. | |

**4.9 세션 고정**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 세션 고정 | |
| 취약점 개요 | 사용자에게 발급받는 세션이 고정되어 있으면, 공격자에게 세션이 드러났을 때,  해당 세션을 통해 서버에 지속적인 공격이 가해질 수 있다. 세션은 로그인 할 때마다 새롭게 부여해야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 유저가 로그인할 때마다 늘 새로운 세션을 발급하도록 한다. | |

**4.10 파일 업로드**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 파일 업로드 | |
| 취약점 개요 | 유저가 업로드 하는 파일에 대한 충분한 검증 절차가 이루어지지 않으면,  Webshell등의 위험한 파일들이 서버에 업로드 되거나, 서버의 파일을 덮어씌울 수 있다. 그러므로 유저가 올리는 파일에 대한 충분한 검증이 이루어져야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 유저가 업로드하는 파일의 확장자를 검사해 jpg, png등의 허용된 확장자가  아니면 파일 업로드 자체를 거부하도록 한다. | |

**4.11 악성 콘텐츠**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 파일 업로드 | |
| 취약점 개요 | 유저가 업로드 하는 파일에 대한 충분한 검증 절차가 이루어지지 않으면,  랜섬웨어와 같은 악성 파일, 악성 콘텐츠가 서버에 업로드 될 수 있다.  그러므로 유저가 올리는 파일에 대한 충분한 검증이 이루어져야 한다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 유저가 업로드하는 파일의 확장자를 검사해 jpg, png등의 허용된 확장자가  아니면 파일 업로드 자체를 거부하도록 한다. | |

**4.12 불충분한 세션 만료**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 불충분한 세션 만료 | |
| 취약점 개요 | 유저에게 발급한 세션이 소멸하지 않고 계속 살아있게 되면, 공격자에게 세션을 도용당할 수 있고, 또한 서버의 리소스가 낭비되게 된다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 유저가 로그아웃하면 세션이 자동으로 파기되도록 하고,  특정 시간이 지나면 세션을 새로 발급하도록 한다. | |

**4.13 자동화 공격**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 자동화 공격 | |
| 취약점 개요 | 서버에 대한 특정 요청이 반복적으로 들어와도 아무런 검증 절차가 이루어지지 못하면, Dos / DDos 공격 등에 취약해진다. 또한 스팸성 계정에 대해서도 취약할 수 있다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 동일한 ip로부터 특정 요청이 반복적으로 날아오게 되면,  해당 ip에 대한 요청이 실제 사람이 보내고 있는 게 맞는지 검증 절차를  추가한다. | |