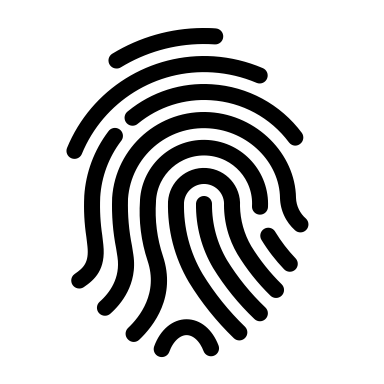
2023년 SB커뮤니티 금취분평 모의해킹 보고서

2023 / 02 /01

SB커뮤니티 금취분평 모의해킹 프로젝트

**모의해킹 결과 보고서 v1.0**

****

목차

1. **모의해킹 수행 정보**
   1. **개요**
   2. **대상**
   3. **수행 기간**
   4. **수행 인력**
2. **모의해킹 결과**

**2.1 총평**

**2.2 점검 항목**

**2.3 결과 요약**

1. **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 SB커뮤니티의 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:9575/** |

* 1. **수행 기간**
* 2023년 1월 30일 ~ 2023년 2월 1일
  1. **수행 인력**

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

1. 모의해킹 결과

**2.1. 총평**

**흔하게 알려진 웹 취약점에 대한 필터링이 존재하지만, 그럼에도 불구하고 기술 / 정책 부문 취약점이 다수 발견되었습니다. 개 중에는 서버에 큰 영향을 끼칠 수 있는 취약점도 존재하기에, 빠른 보안 패치가 시급합니다. 이외에도 서버의 보안을 취약하게 만드는 정책적 취약점도 수정이 필요합니다.**

**2.2 점검 항목**

**금융보안원의 전자금융기반시설 보안 취약점 평가기준에 기반하여 취약점을 점검하였습니다.**

**SQL Injection**

**악성파일 업로드**

**부적절한 이용자 인가 여부**

**이용자 인증정보 재사용**

**고정된 인증정보 이용**

**유추가능한 인증정보 이용**

**유추가능한 초기화 비밀번호 이용**

**단말기 내 중요정보 저장 여부**

**메모리 내 중요정보 노출 여부**

**파일 다운로드**

**외부사이트에 의한 시스템 운영정보 노출 여부**

**유추 가능한 세션ID**

**쿠키변조**

**운영체제 명령실행**

**XML 외부객체 공격 (XXE)**

**리다이렉트 기능을 이용한 피싱 공격**

**LDAP Injection**

**SSI Injection**

**불충분한 이용자 인증**

**화면 내 중요정보 평문노출 여부**

**자동화공격**

**버퍼오버플로우 (Buffer Overflow Attack)**

**포맷스트링 (Format String Attack)**

**단말기 브라우저 영역 내에서의 중요정보 노출**

**앱 소스코드 내 운영정보 노출 여부**

**화면 강제실행에 의한 인증단계 우회**

**데이터 평문전송**

**크로스사이트 요청변조 (CSRF)**

**디렉토리 목록 노출**

**서버 인증서 무결성 검증**

**시스템 운영정보 노출 여부**

**인증 오류 횟수 제한기능 제공 여부**

**불충분한 세션종료 처리**

**취약한 HTTPS 프로토콜 이용**

**취약한 HTTPS 암호 알고리즘 이용**

**취약한 HTTPS 컴포넌트 사용**

**취약한 HTTPS 재협상 허용**

**불필요한 웹 메서드 허용**

**관리자 페이지 노출 여부**

**불필요한 파일 노출 여부**

**크로스 사이트 스크립팅 (XSS)**

**디버그 로그 내 중요정보 노출 여부**

**백그라운드 화면 보호**

**서버 사이드 요청 위조 (SSRF)**

**세션정보 재사용**

**인증수단 소유자 검증 여부**

**모바일 DeepLink 도용 취약점**

**SB커뮤니티에서는 위 항목 중 총 10개의 취약점이**

**발견되었습니다.**

* 1. 결과 요약

|  |  |  |
| --- | --- | --- |
| No | 취약점 | 내용 |
| 1 | **크로스 사이트**  **스크립팅** | **회원/비회원 게시판에서**  **XSS취약점이 발견되었습니다.** |
| 2 | **유추 가능한 인증 정보 이용** | **회원 가입 시 보안 강도가 약한 패스워드의**  **사용이 허용되었습니다.** |
| 3 | **디렉토리 목록 유출** | **서버의 디렉토리 목록이 유출되었습니다.** |
| 4 | **SQL Injection** | **서버에서 SQL Injection 취약점이 발견되었습니다.** |
| 5 | **시스템 운영 정보 노출** | **시스템과 서버 정보가 유출되었습니다.** |
| 6 | **데이터 평문 전송** | **서버와 클라이언트 사이 통신이**  **암호화 되지 않고 평문으로 전송 되고 있습니다.** |
| 7 | **고정된 인증 정보 이용** | **서버가 고정된 인증 정보를 이용합니다.** |
| 8 | **자동화 공격 가능** | **서버에 자동화 공격이 가능합니다.** |
| 9 | **불충분한 세션 종료 처리** | **서버의 세션 종료 처리가 불충분합니다.** |
| 10 | **인증 오류 횟수 제한 기능 제공 여부** | **로그인 시도 시 패스워드 입력 제한 횟수가**  **존재하지 않습니다.** |

1. 취약점 상세 내용

**3.1 크로스 사이트 스크립팅**

**(**[**http://normaltic.com:9575/user\_board/board.php**](http://normaltic.com:9575/user_board/board.php)**)**

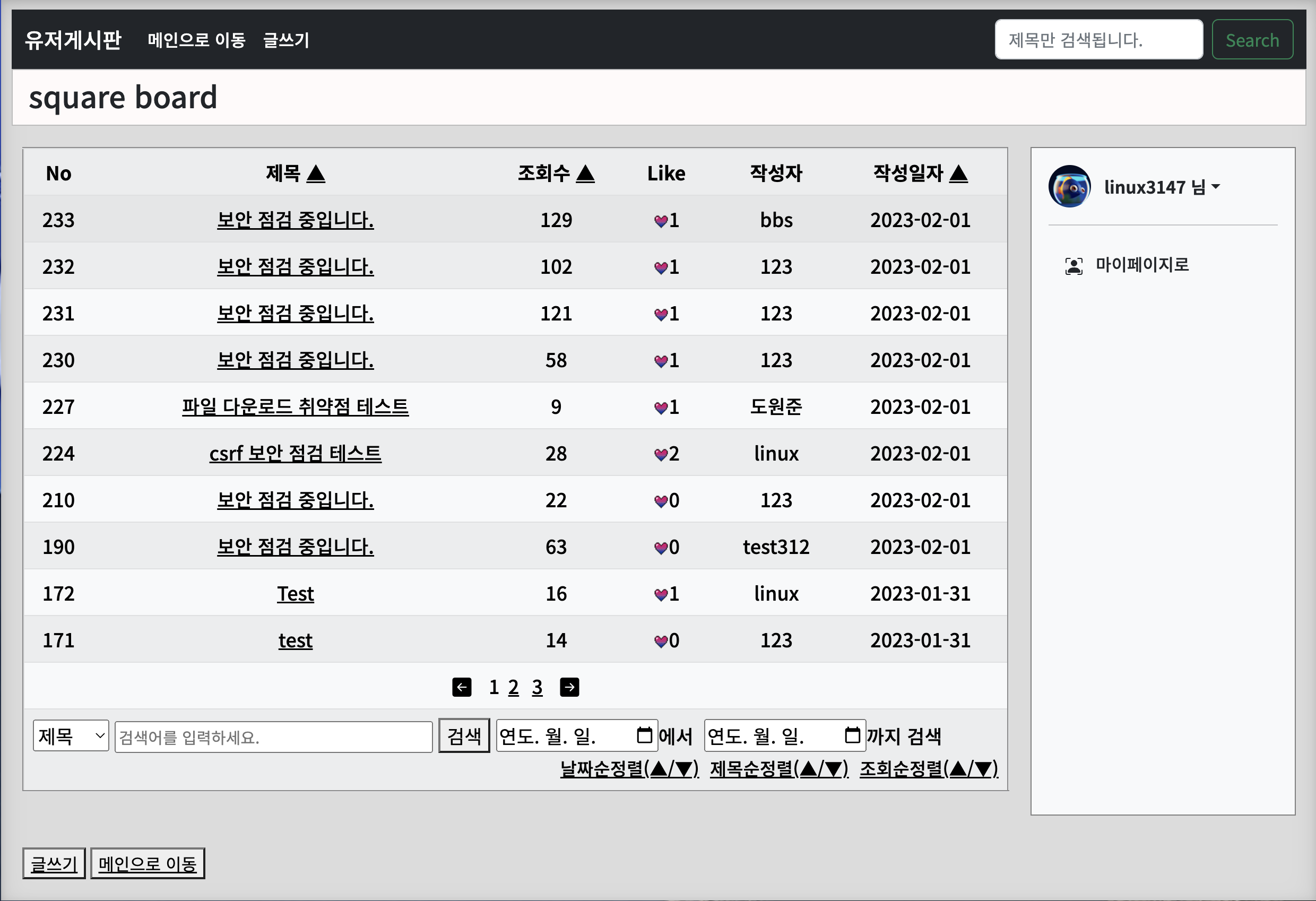
**(http://normaltic.com:9575/ask\_board/ask.php)**

**3.1.1 Cross Site Scripting**

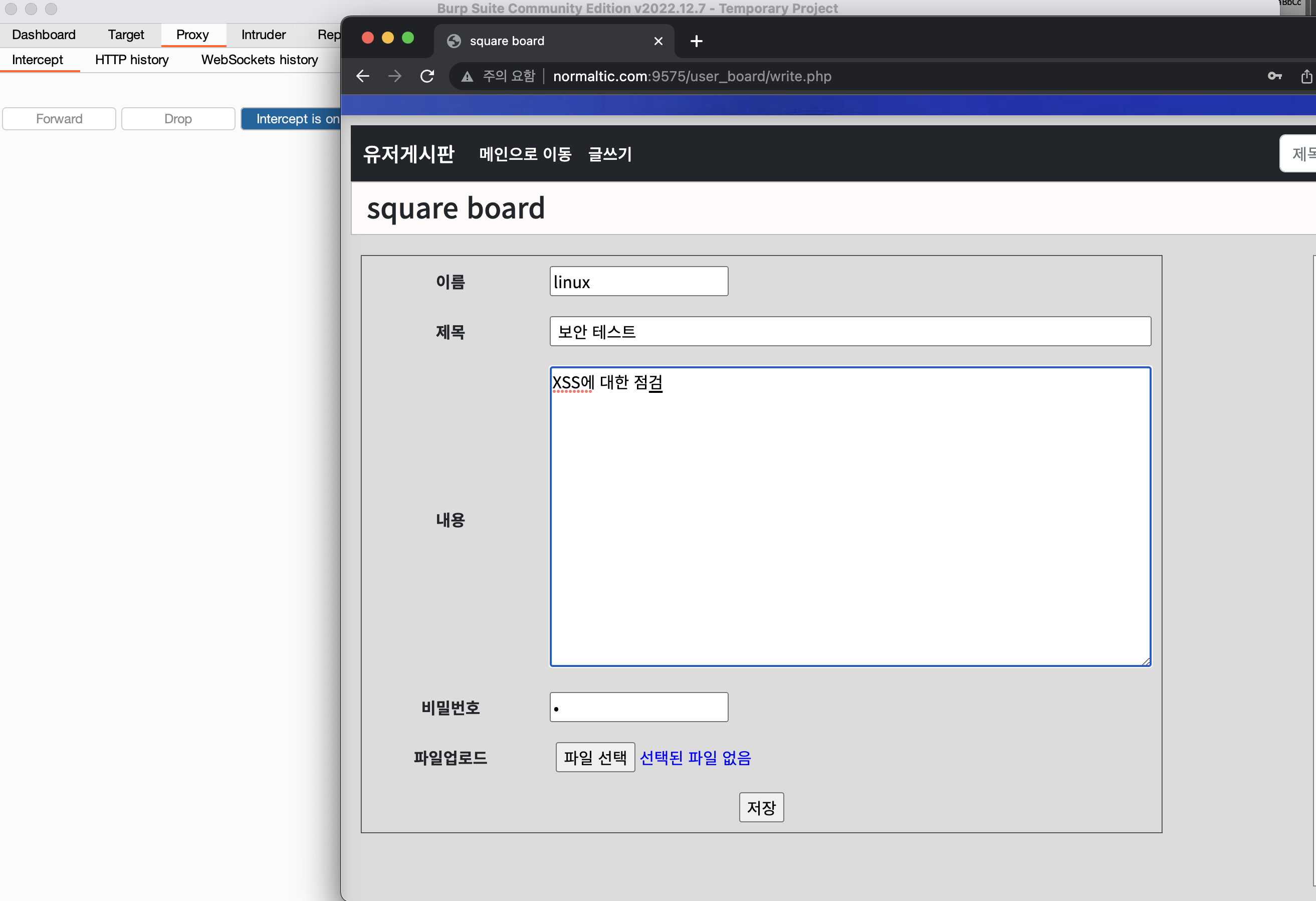
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| [**http://normaltic.com:9575/user\_board/board.php**](http://normaltic.com:9575/user_board/board.php)  **http://normaltic.com:9575/ask\_board/ask.php** | x | 회원 게시판,  문의 게시판 |
| 취약점 상세내역 | 회원 게시판의 글의 인덱스 번호와 작성자에서 xss가 발생하였고,  문의 게시판에선 글의 작성자와 제목에서 xss취약점이 발생하였습니다. | | |

**3.1.2 취약점 확인\_1**

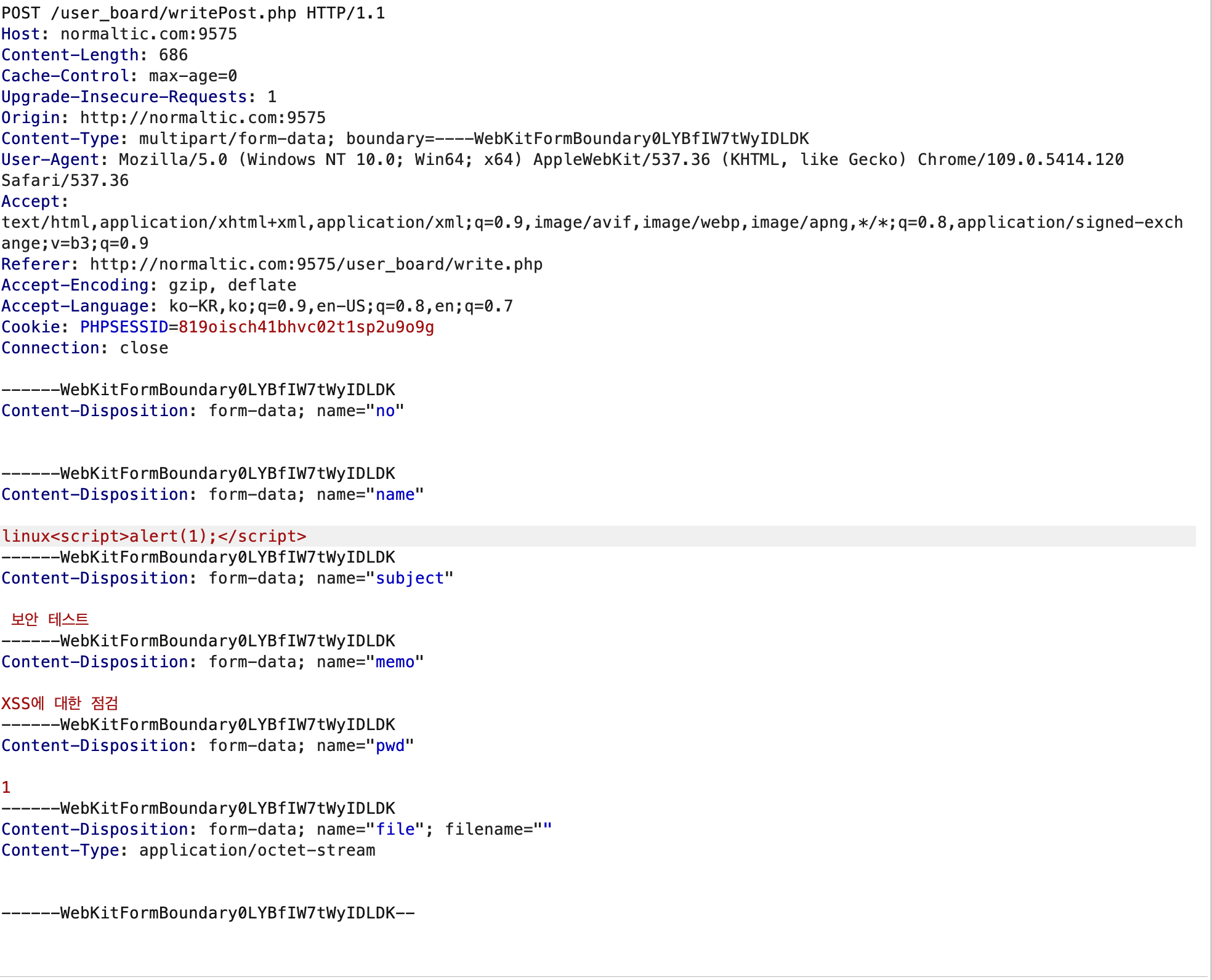
**1. 로그인을 한 채로 게시판에 접속합니다.**

****

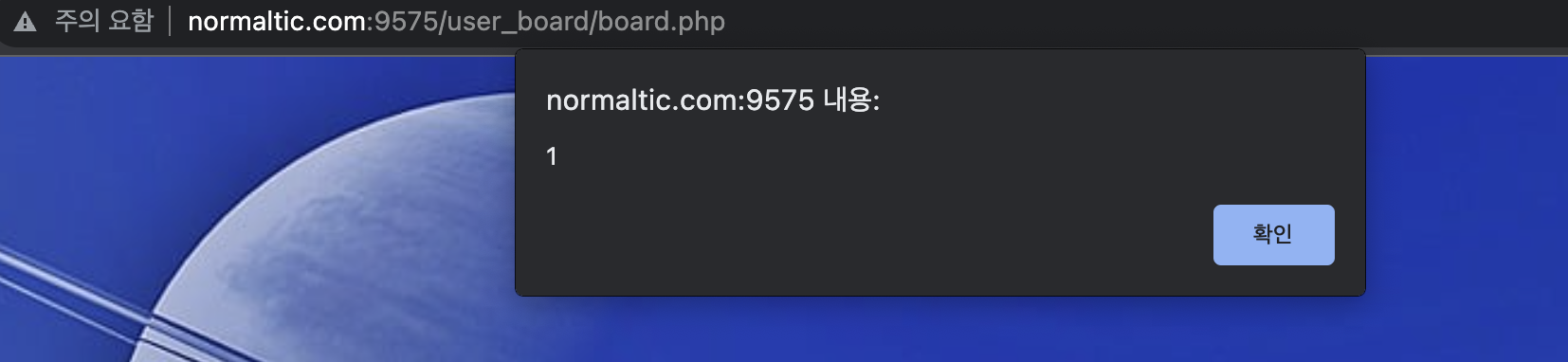
**2. BurpSuite의 proxy기능을 켠 상태로 글을 작성합니다.**



**3. name부분에 xss구문을 삽입합니다.**

****

**4. xss 발생 확인.**

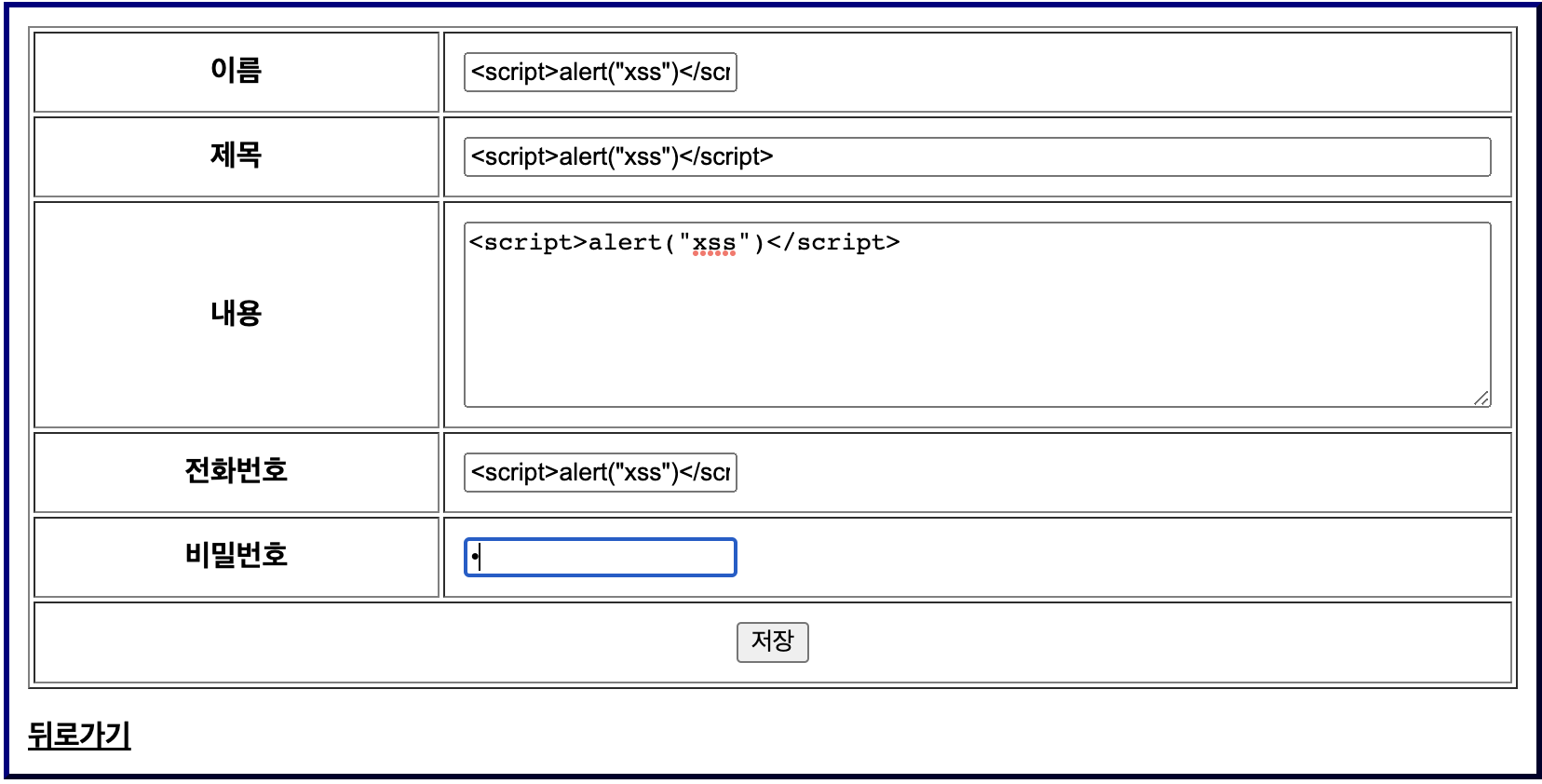
****

**3.1.2 취약점 확인\_2**

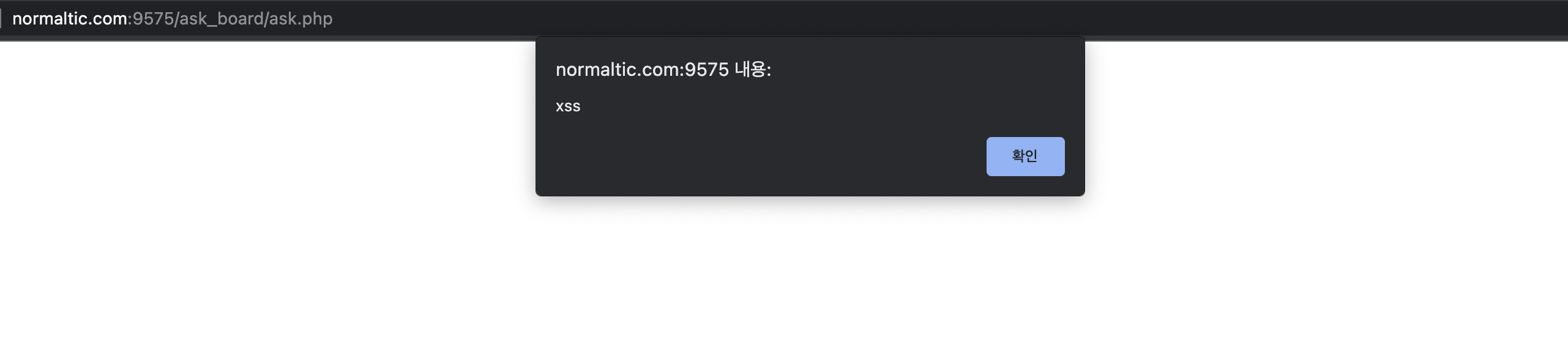
**1. 비회원 문의 게시판에 접속합니다.**

****

**2. 글쓰기 메뉴를 클릭하고, 아래와 같이 값을 삽입합니다.**

****

**3. XSS 취약점 확인.**

**[보안 권고안]**

**글의 내용 뿐만 아니라 화면에 표시되는 모든 값에 대해 특수문자 처리를 해줍니다.**

**3.2 유추 가능한 인증정보 이용**

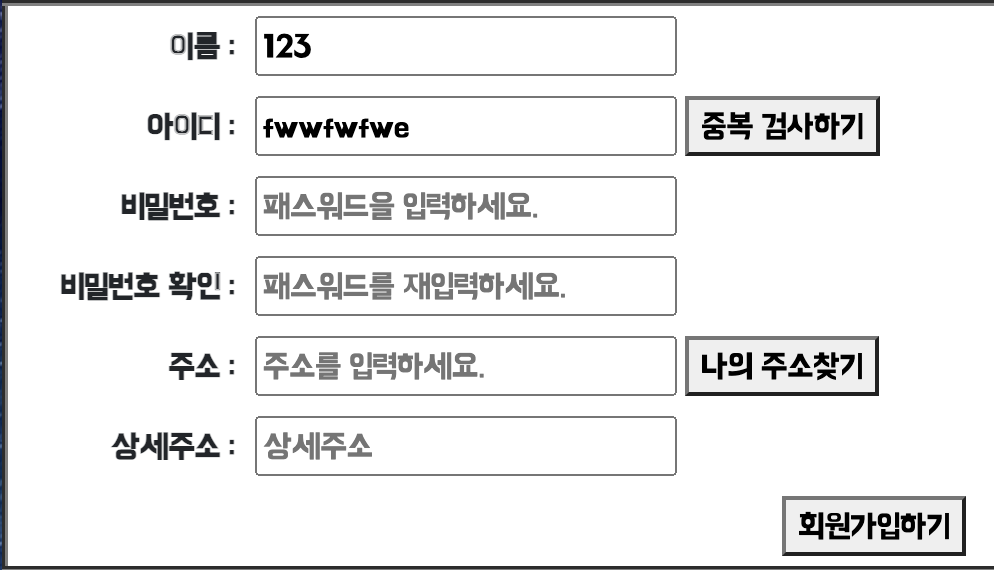
**(http://normaltic.com:9575/register.php)**

**3.2.1 유추 가능한 인증정보 이용**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9575/register.php** | x | 회원가입 |
| 취약점 상세내역 | 사이트에서 회원가입을 할 때, 유추 가능한 인증정보를 대입해도  회원가입이 가능합니다. | | |

**3.2.2 취약점 확인**

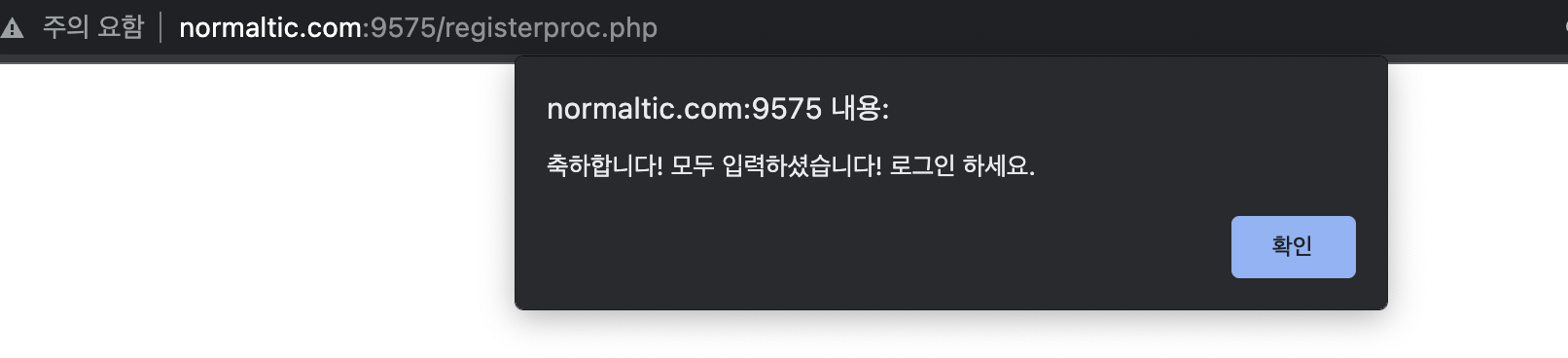
**1. 회원가입 페이지에 접속합니다.**

****

**2. 패스워드를 1로 지정하고 다른 값들을 모두 대입한 후, 회원가입하기를 누릅니다.**

****

**3. 1이라는 보안 강도가 낮은 비밀번호를 사용해도 회원가입이 완료됩니다.**

****

**[보안 권고안]**

**회원가입하기 위해서는 보안 강도가 높은 비밀번호만 사용할 수 있도록 프로그래밍을 합니다.**

**3.3 디렉토리 목록 노출**

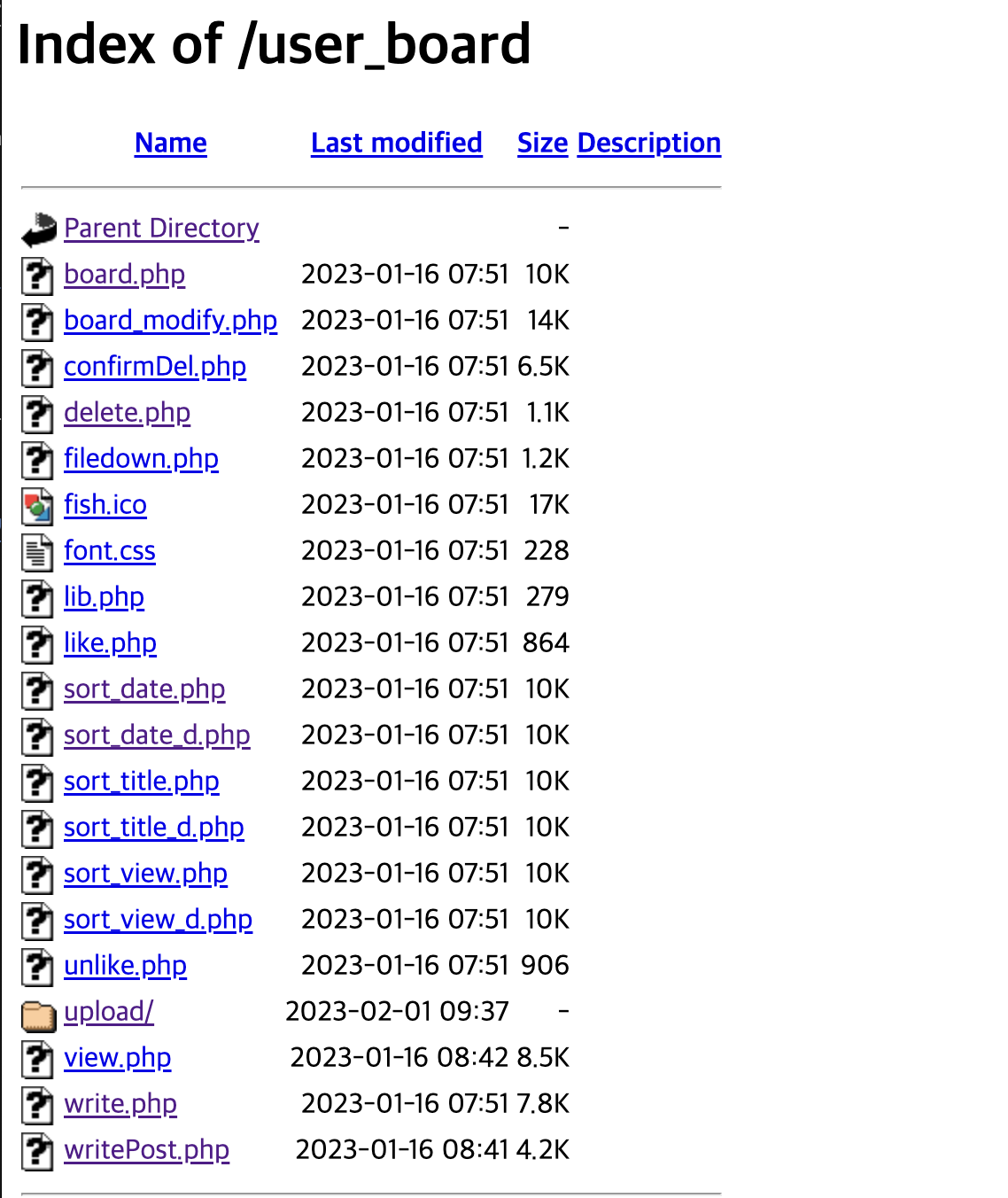
**(http://normaltic.com:9575/**

**3.3.1 디렉토리 목록 노출**

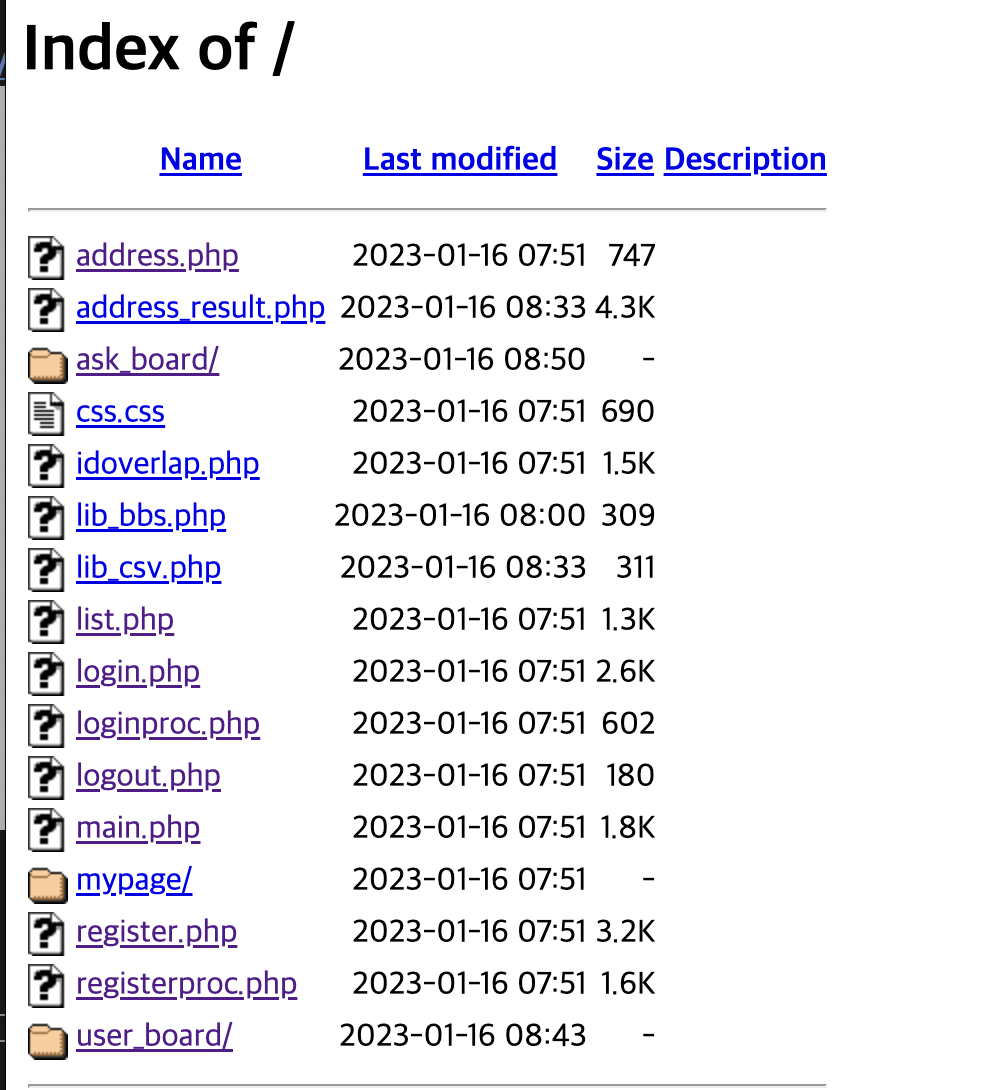
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9575/** | **http://normaltic.com:9575/user\_board/** | 서버 파일의 상위  디렉토리 |
| 취약점 상세내역 | 서버의 디렉토리에서 아무런 파일도 요청하지 않을 시, 해당 디렉토리에 있는  모든 파일이 표기됩니다. | | |

**3.3.2 취약점 확인**

**1. 서버의 URL에** [**http://normaltic.com:9575/user\_board/**](http://normaltic.com:9575/user_board/) **를 입력하여 접속한다.**

****

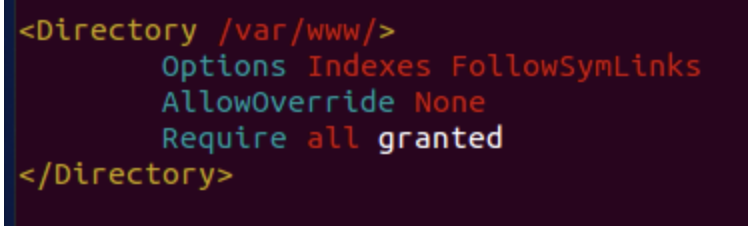
**2. Parent Directory를 누른다.**

****

**3. 서버 전체의 파일을 열람 가능하다.**

**[보안 권고안]**

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

****

**해당 옵션의 Indexes를 제거한다.**

**3.4 SQL Injection**

**(**[**http://normaltic.com:9575/user\_board/board.php**](http://normaltic.com:9575/user_board/board.php)**)**

**(**[**http://normaltic.com:9575/mypage/mypage.php**](http://normaltic.com:9575/mypage/mypage.php)**)**

**(**[**http://normaltic.com:9575/mypage/mod\_addr.php**](http://normaltic.com:9575/mypage/mod_addr.php)**)**

**(http://normaltic.com:9575/register.php)**

**3.4.1 SQL Injection**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **(**[**http://normaltic.com:9575/user\_board/board.php**](http://normaltic.com:9575/user_board/board.php)**)**  **(**[**http://normaltic.com:9575/mypage/mypage.php**](http://normaltic.com:9575/mypage/mypage.php)**)**  **(**[**http://normaltic.com:9575/mypage/mod\_addr.php**](http://normaltic.com:9575/mypage/mod_addr.php)**)**  **(http://normaltic.com:9575/register.php)** | **x** | 게시판 제목 검색 기능,  비밀번호 변경 ,  문의 게시판 제목 검색 기능,  주소 검색 기능,  아이디 중복 체크 기능 |
| 취약점 상세내역 | 서버의 여러 군데에서 SQL injection 취약점이 발생하였습니다. | | |

**3.4.2 취약점 확인\_1**

**1. 게시판에 접속합니다.**

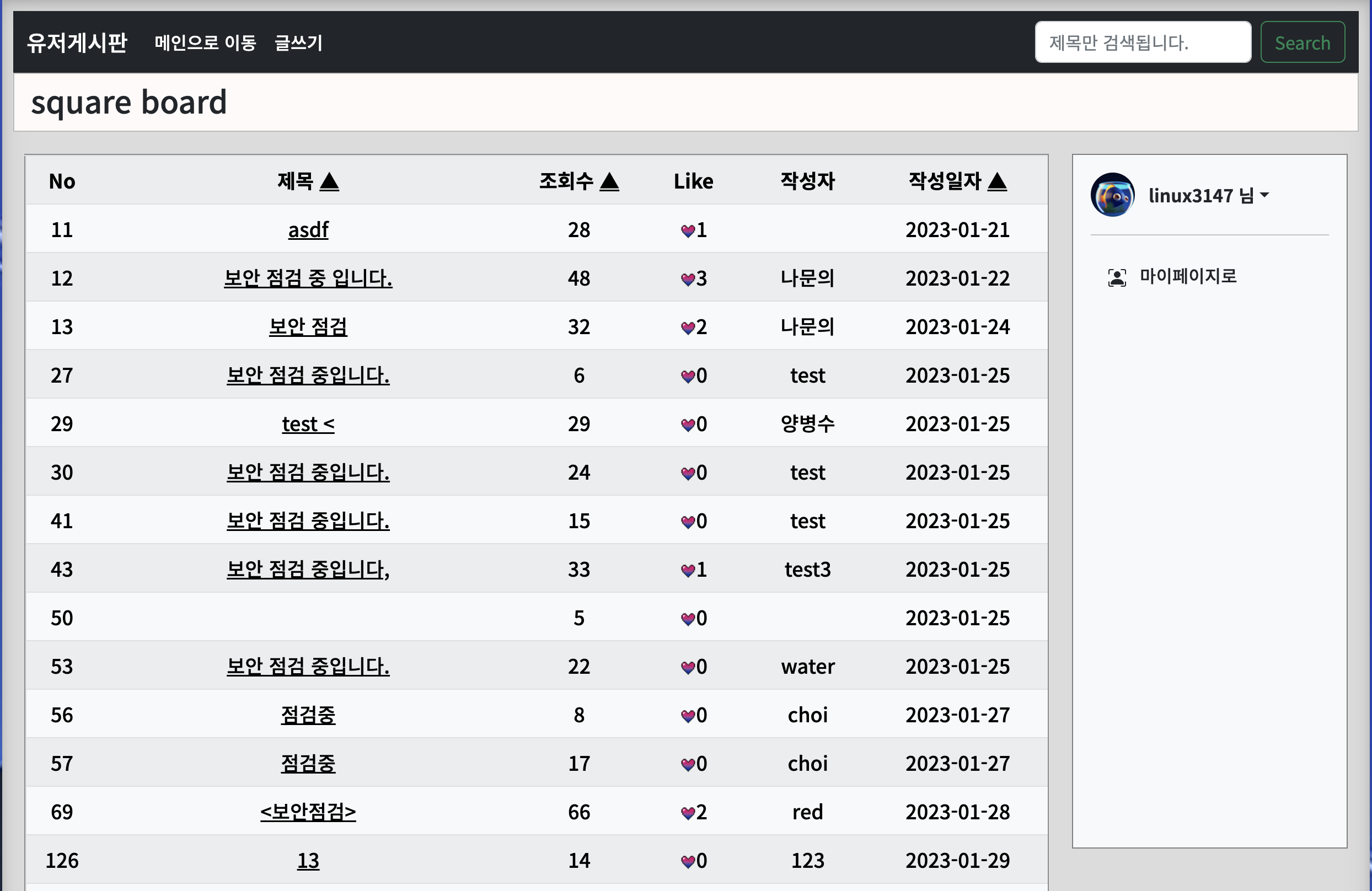
****

**2. 게시판 제목 검색 입력창에**

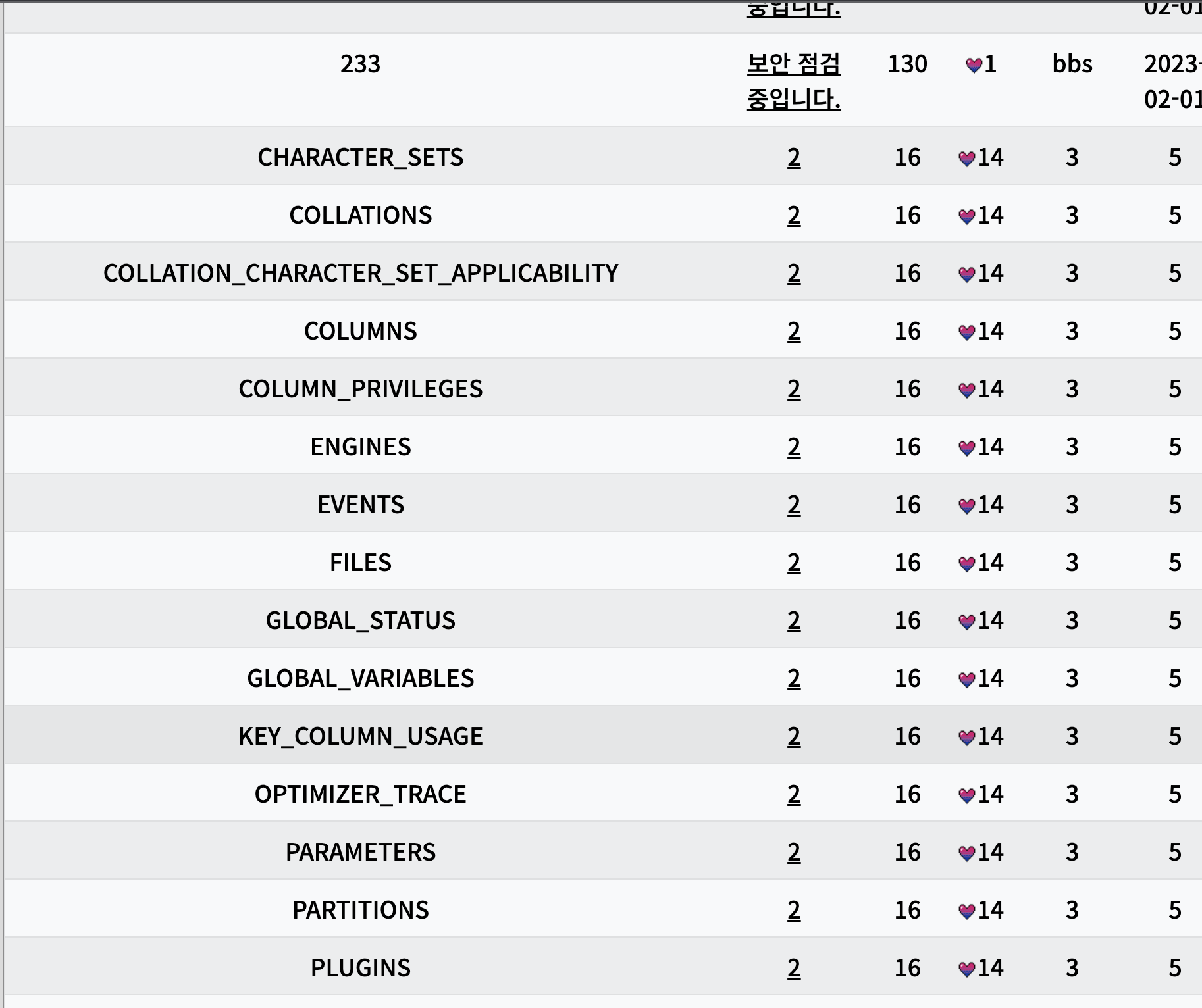
**%' union select table\_name,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 from**

**information\_schema.tables #**

**을 입력합니다.**

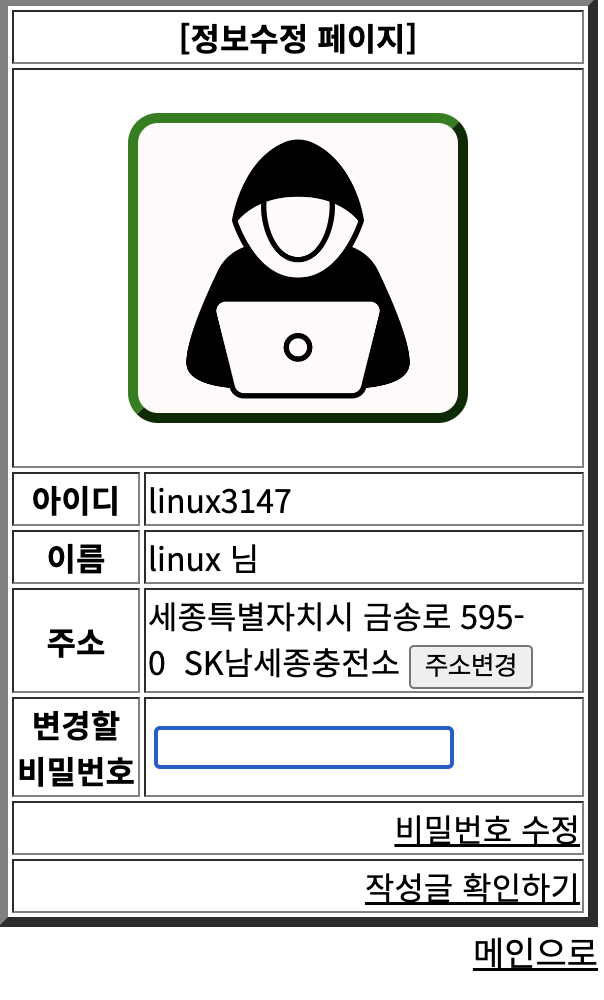
****

**3. 게시판 글 목록에 테이블 이름이 출력된 것을 확인 할 수 있습니다.**

****

**3.4.2 취약점 확인\_2**

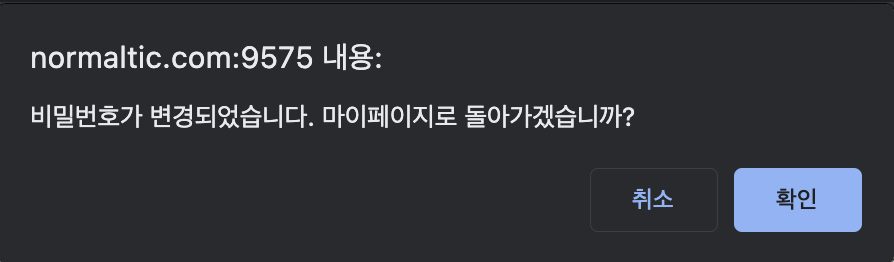
**1. 로그인을 한 후 마이페이지에 접속합니다.**

****

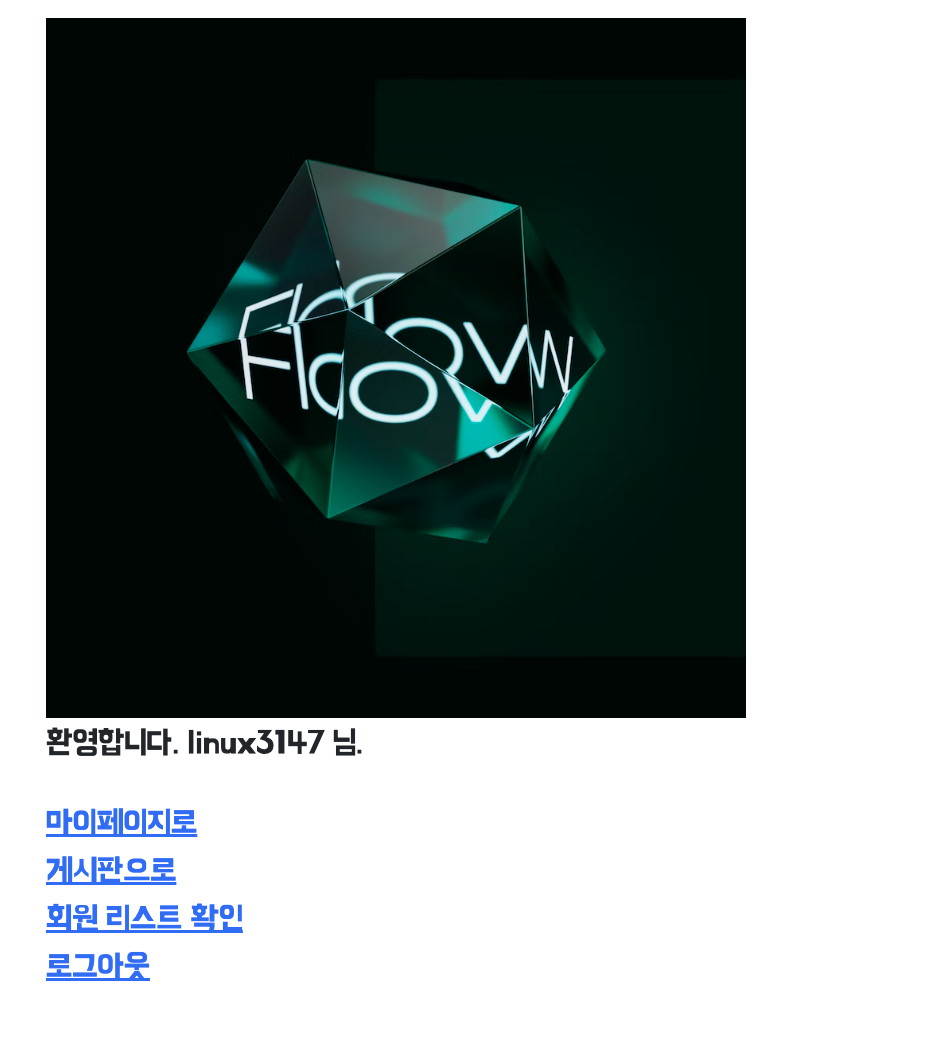
**2. 변경할 비밀번호란에 ‘ or ‘1’=’1 을 입력하고 값을 전송합니다. (띄어쓰기 포함)**

****

**3. 비밀번호 변경 성공 알람창이 표시됩니다.**

****

**4. 로그인창에 아이디를 입력하고, 비밀번호에 1을 입력하면 로그인이 됩니다.**

****

**3.4.2 취약점 확인\_3**

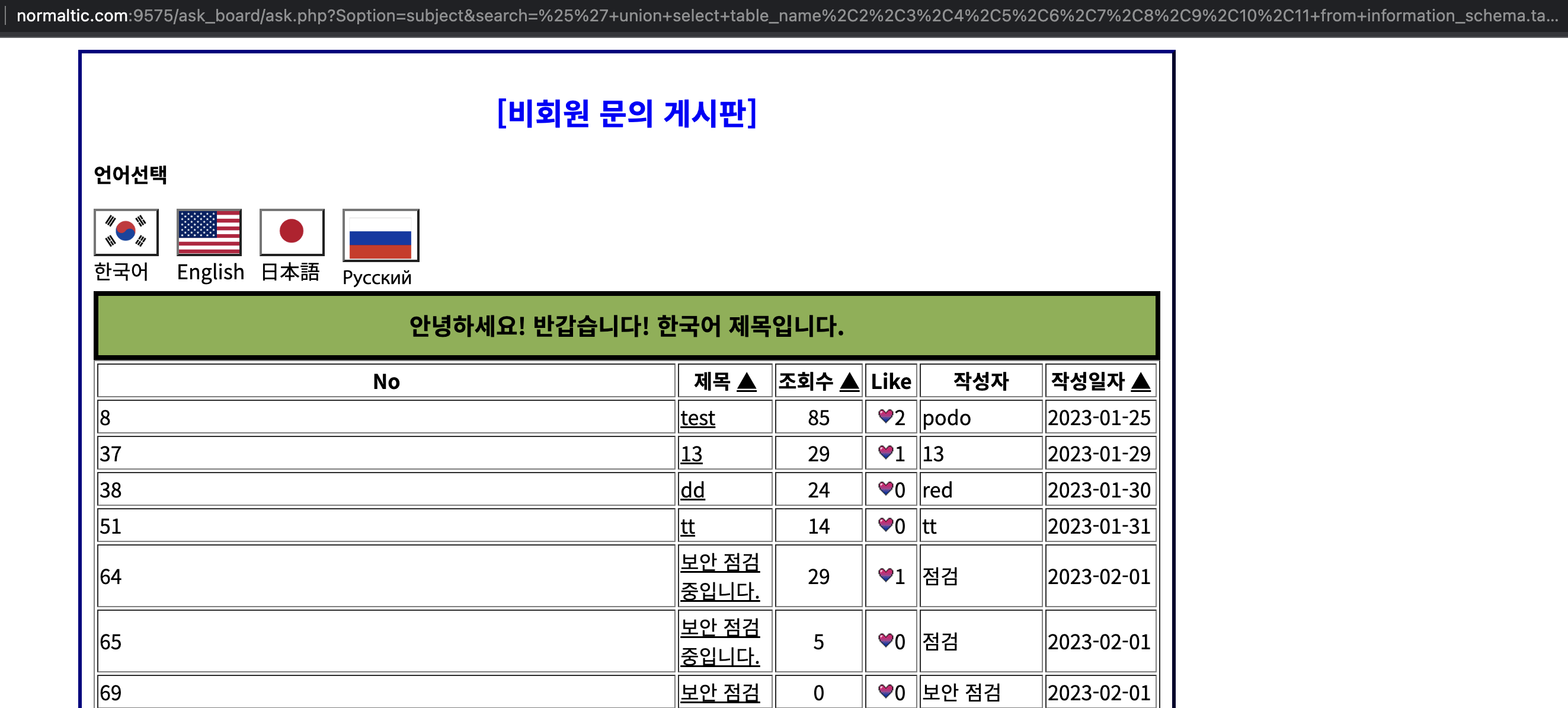
**1. 비회원 문의 게시판에 접속합니다.**

****

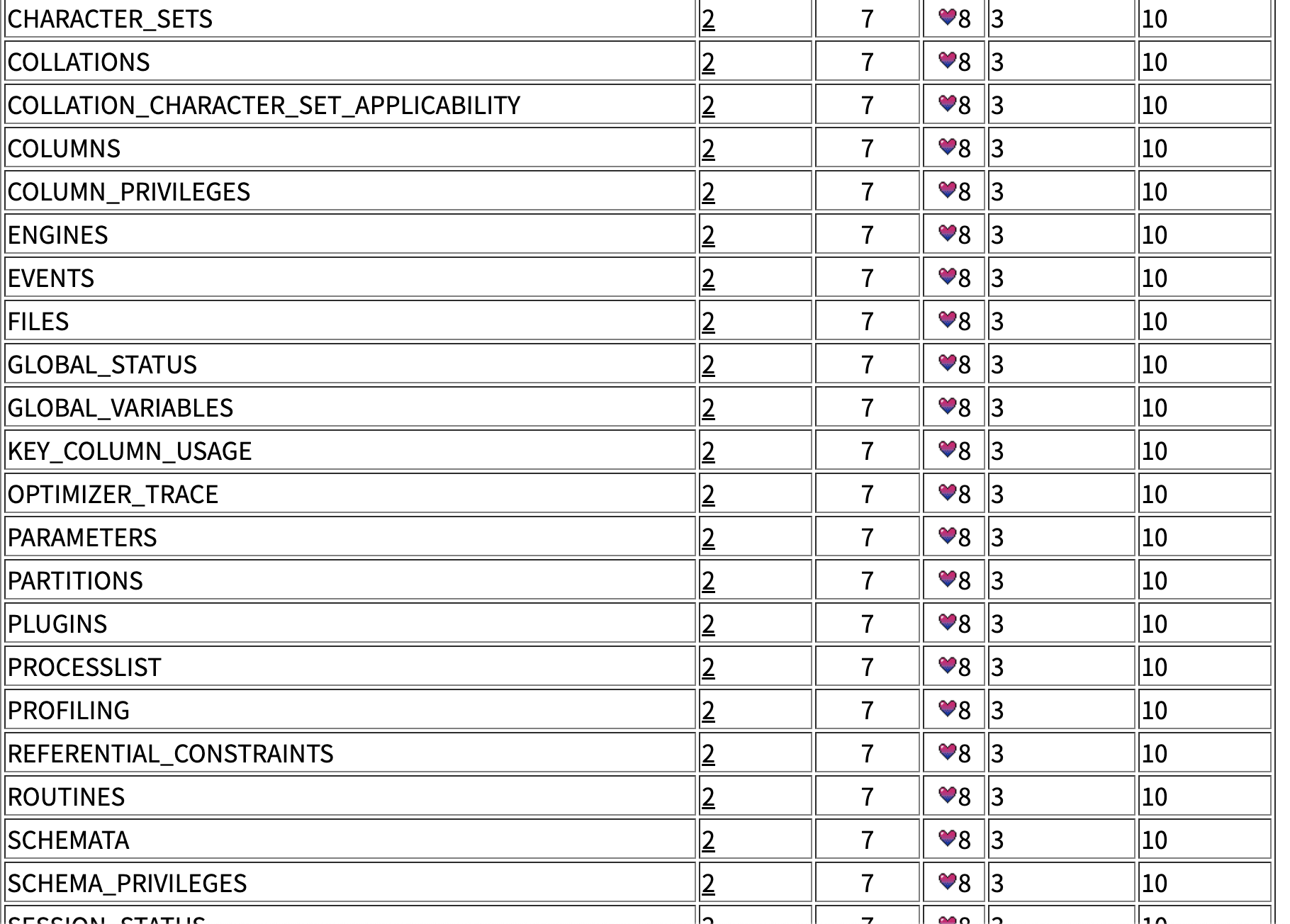
**2. 게시판 글 제목 검색 입력란에**

**%' union select table\_name,2,3,4,5,6,7,8,9,10,11 from information\_schema.tables#**

**다음과 같은 입력 값을 주고 검색을 합니다.**

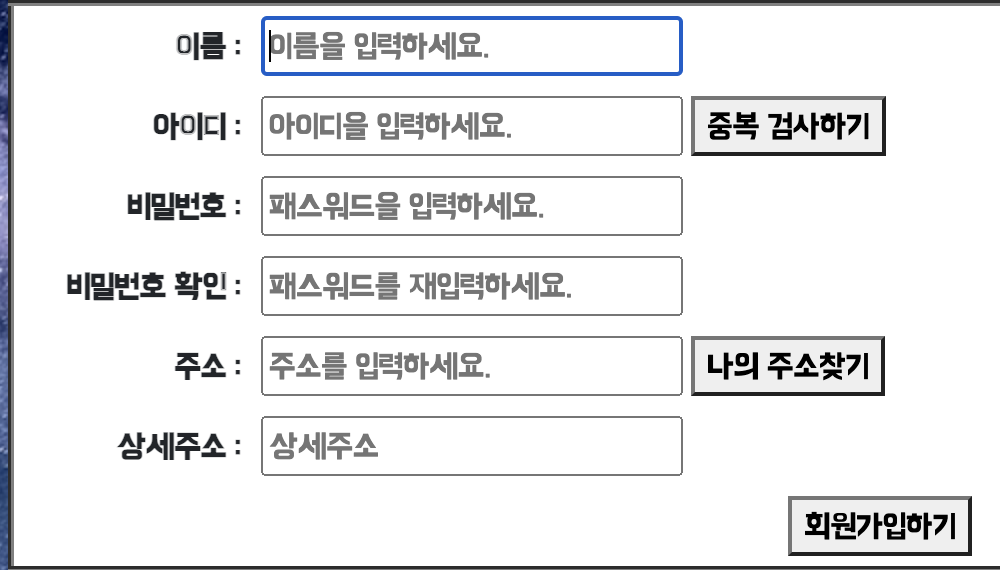
****

**3. 게시판 글 목록에 테이블 이름이 출력된 것을 확인 할 수 있습니다.**

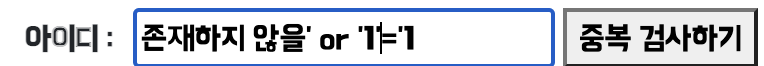
****

**3.4.2 취약점 확인\_4**

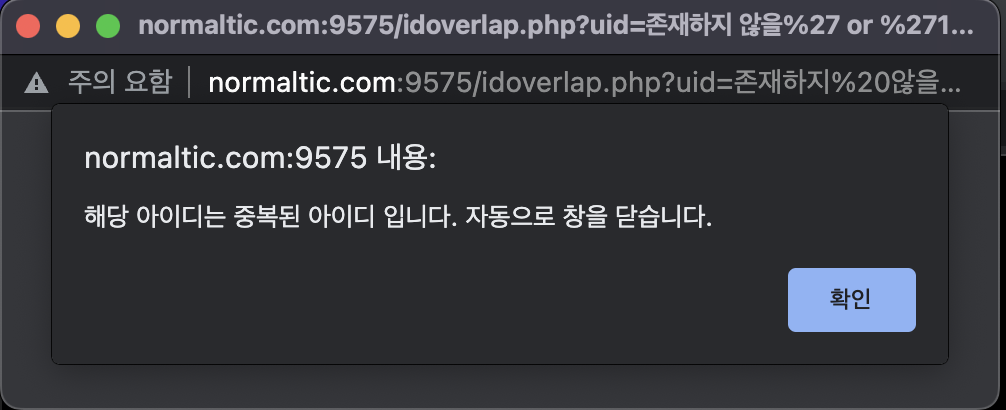
**1. 회원가입 페이지에 접속합니다.**

****

**2. 아이디란에 아무아이디’ or ‘1’=’1을 입력하고 중복 검사를 누릅니다.**

****

**3. ‘ or ‘1’=’1에 의해 쿼리문이 참이 되어 아이디가 중복된다고 표기됩니다.**

****

**[보안 권고안]**

**서버와 클라이언트 사이 이동하는 모든 값에 대해 특수문자 처리를 합니다.**

**3.5 시스템 운영 정보 노출 여부**

**(http://normaltic.com:9575/)**

**3.5.1 시스템 운영 정보 노출 여부**

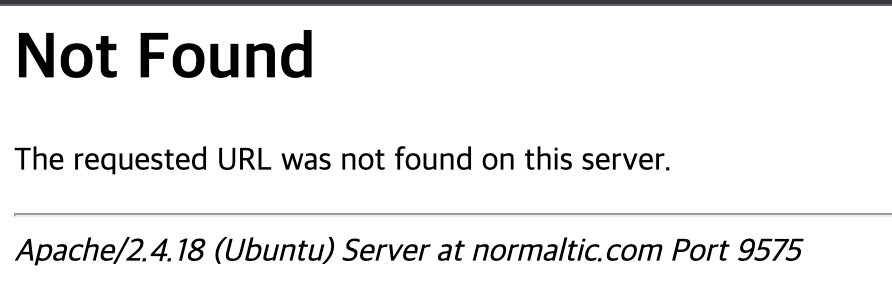
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9575/** | **http://normaltic.com:9575/123** | 서버 전체 |
| 취약점 상세내역 | 서버가 오류를 표기할 때, 서버에서 사용하고 있는 OS와 서버 정보등을 함께  표기합니다. | | |

**3.5.2 취약점 확인**

**1. 서버의 URL에 존재하지 않는 파일을 요청합니다.**

****

**2. 오류가 출력됨과 함께 서버 정보가 표기됩니다.**

****

**[보안 권고안]**

**Httpd.conf파일에서 ServerSignature 옵션을 off로 변경하고**

**ServerTokens Prod를 적용시킨다.**

**3.6 데이터 평문 전송**

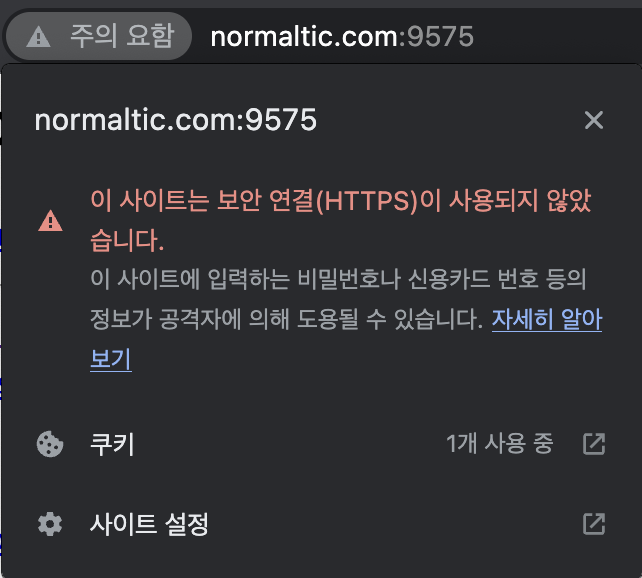
**(http://normaltic.com:9575/)**

**3.6.1 데이터 평문 전송**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9575/** | **x** | 서버 전체 |
| 취약점 상세내역 | 서버가 http 통신을 사용하기에 서버와 클라이언트 사이 이동하는 데이터가  암호화되지 않고 모두 평문으로 노출됩니다. | | |

**3.6.2 취약점 확인**

**1. 서버가 http 통신을 사용 중인 것을 확인할 수 있습니다.**

****

**[보안 권고안]**

**HTTPS 통신을 사용합니다.**

**3.7 고정된 인증정보 이용**

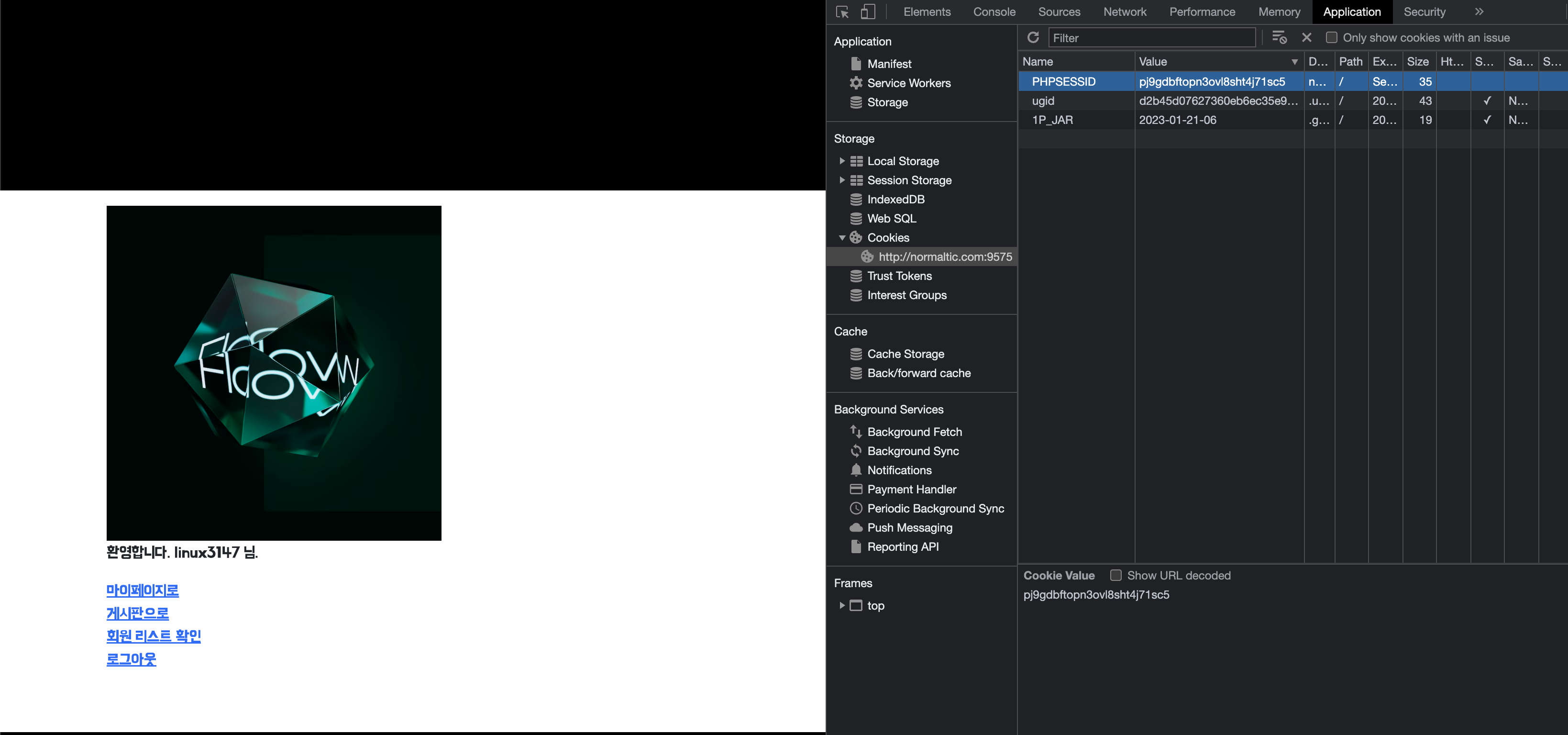
**(http://normaltic.com:9575/)**

**3.7.1 고정된 인증정보 이용**

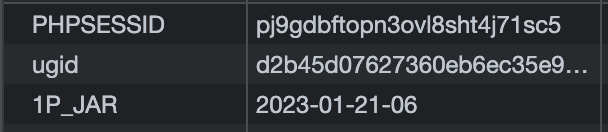
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9575/** | **x** | 서버 전체 |
| 취약점 상세내역 | 서버가 로그인을 할 때마다 같은 세션을 발급합니다. | | |

**3.7.2 취약점 확인**

**1. 로그인 후 쿠키를 확인합니다.**

****

**2. 로그아웃을 했다가 재로그인 후 쿠키를 다시 확인합니다.**

****

**3. 같은 쿠키를 사용 중인 것을 확인 할 수 있다.**

**[보안 권고안]**

**로그아웃 할 때 마다 세션을 파기하고,**

**로그인 할 때 마다 랜덤값으로 세션을 생성하여 발급하도록 한다.**

**3.8 자동화 공격 가능**

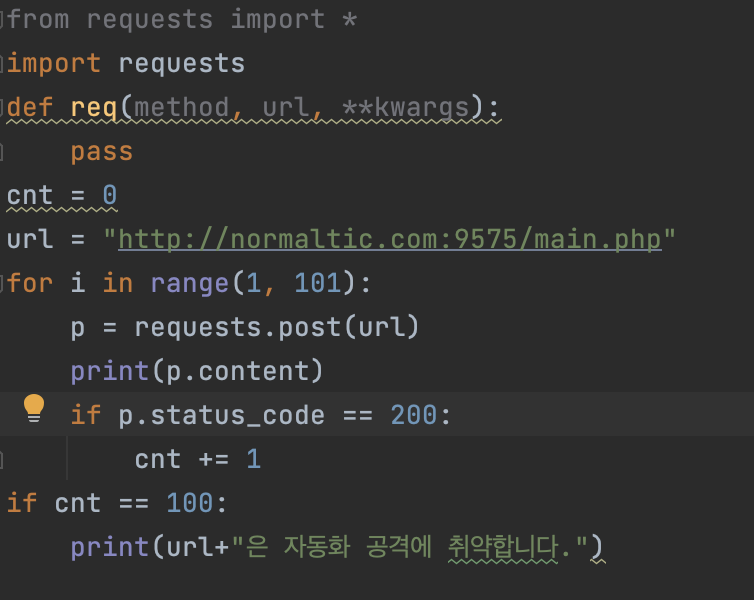
**(http://normaltic.com:9575/)**

**3.8.1 자동화 공격 가능**

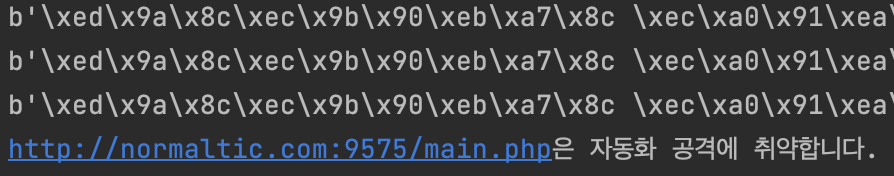
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9575/** | **x** | 서버 전체 |
| 취약점 상세내역 | 서버가 반복적인 동일한 요청에 대해 어떠한 검증 절차도 거치지 않습니다. | | |

**3.8.2 취약점 확인**

**1. 서버에 대해 같은 요청을 반복하는 코드를 설계하여 요청을 보냅니다.**

****

**2. 자동화 공격에 대한 취약성 확인**

****

**[보안 권고안]**

**같은 ip에서 일정 시간안에 같은 요청이 반복적으로 날아오면, 해당 요청이**

**사람이 보내고 있는 게 맞는 지 확인하는 검증하는 코드를 설계합니다.**

**3.9 불충분한 세션 종료 처리**

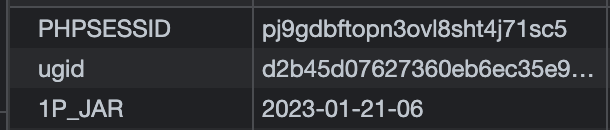
**(http://normaltic.com:9575/)**

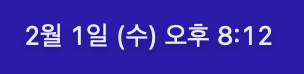
**3.9.1 불충분한 세션 종료 처리**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9575/** | **x** | 서버 전체 |
| 취약점 상세내역 | 서버가 발급한 세션을 일정시간이 지나도 파기하지 않습니다. | | |

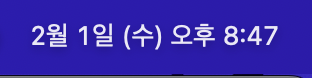
**3.9.2 취약점 확인**

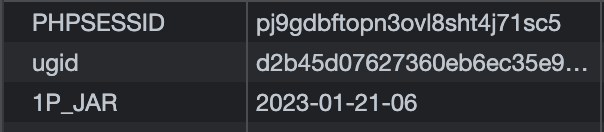
**1. 로그인을 한 후 세션을 확인합니다.**

****

****

**2. 30분 정도의 시간이 지나도 세션이 그대로 남아있습니다.**

****

****

**[보안 권고안]**

**세션이 일정 시간이 지나면 자동으로 파괴되도록 프로그래밍합니다.**

**3.10 인증 오류 횟수 제한 기능 제공 여부**

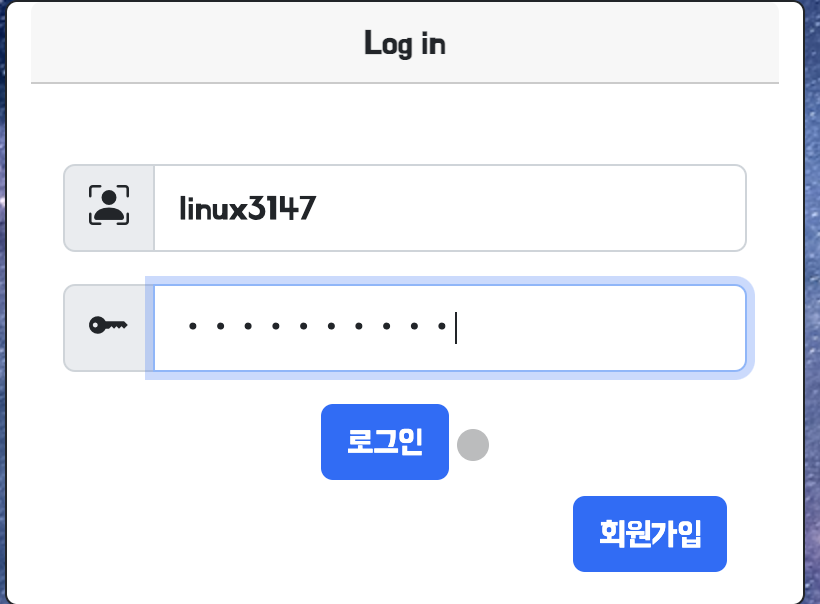
**(http://normaltic.com:9575/login.php)**

**3.9.1 인증 오류 횟수 제한 기능 제공 여부**

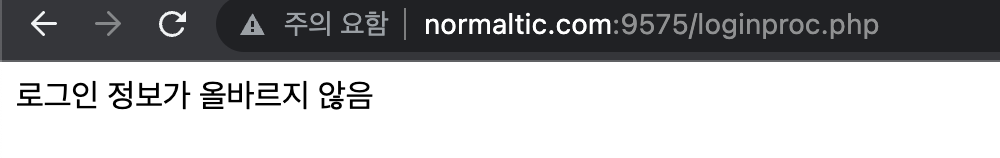
|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 설명 | | |
| 발견URL | **URL** | **파라미터** | **메뉴** |
| **http://normaltic.com:9575/login.php** | **x** | 서버 전체 |
| 취약점 상세내역 | 로그인 비밀번호를 여러 번 틀려도 아무런 제재를 하지 않습니다. | | |

**3.9.2 취약점 확인**

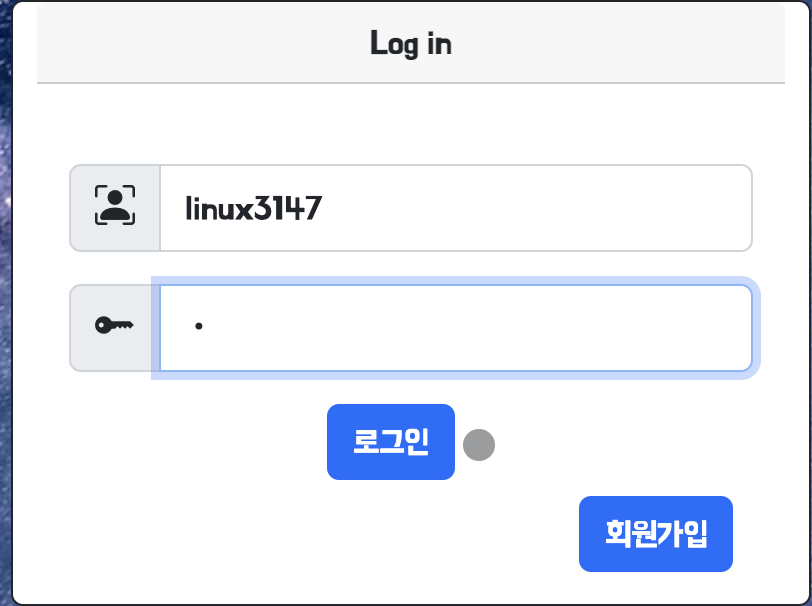
**1. 서버에 아이디와 틀린 비밀번호를 입력하고 로그인 요청을 보냅니다.**

****

**2. 해당 작업을 반복합니다.**

****

**3. 10번 이상 반복했음에도 어떠한 제재도 없음을 확인할 수 있습니다.**

****

**[보안 권고안]**

**잘못된 로그인 시도를 지속적으로 행하면, 로그인 시도를 못하도록 막거나**

**일정 시간 동안 로그인 시도를 못하도록 합니다.**

1. 보안 권고안

**4.1 Cross Site Scripting**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | Cross Site Scripting | |
| 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. | |

**4.2 유추 가능한 인증 정보 이용**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 유추 가능한 인증 정보 이용 | |
| 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. | |

**4.3 디렉토리 목록 노출**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 디렉토리 목록 노출 | |
| 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. | |

**4.4 SQL Injection**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | SQL Injection | |
| 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. 사용자가 입력한 파라미터 값을 mysqli\_real\_escape\_string 함수를 통해 string escapte처리를 해준다.  2. string escape 처리가 완료된 변수를 SQL Query문에 삽입한다. | |

**4.5 시스템 운영 정보 노출 여부**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 시스템 운영 정보 노출 여부 | |
| 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 고정된 인증 정보 이용**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 고정된 인증 정보 이용 | |
| 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. | |

**4.8 자동화 공격 가능**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 자동화 공격 가능 | |
| 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 검증하는 절차를  추가한다. | |

**4.9 불충분한 세션 종료 처리**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 불충분한 세션 종료 처리 | |
| 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. | |

**4.10 인증 오류 횟수 제한 기능 제공 여부**

|  |  |  |
| --- | --- | --- |
| 취약점 항목 | 인증 오류 횟수 제한 기능 제공 여부 | |
| 취약점 개요 | 잘못된 로그인 요청이 계속 되어도 아무런 조치가 취해지지 않으면,  서버가 브루트 포싱 공격에 취약해질 수 있습니다. | |
| 보안 조치 방법 | | |
| * 보안 조치 방법   1. 같은 ip에서 잘못된 로그인 시도 요청이 계속되면,  로그인을 일정시간동안 시도하지 못하도록 차단한다. | |