

TIFFANYINDEX

1. SQL Injection

2. 위험한 형식 파일 업로드

3. 메모리 버퍼 오버플로우

4. 사용자 하드디스크에 저장되는
쿠키를 통한 정보 노출

5. 경로 조작 및 자원 삽입

6. 오류 메시지를 통한 정보 노출

7. 적절한 인증 없는 중요기능 허용

8. 반복된 인증시도 제한 기능 부재

9. 약한 문자열 강도

10. CSRF
(Cross-Site Request Forgery)

11. Null pointer 역참조

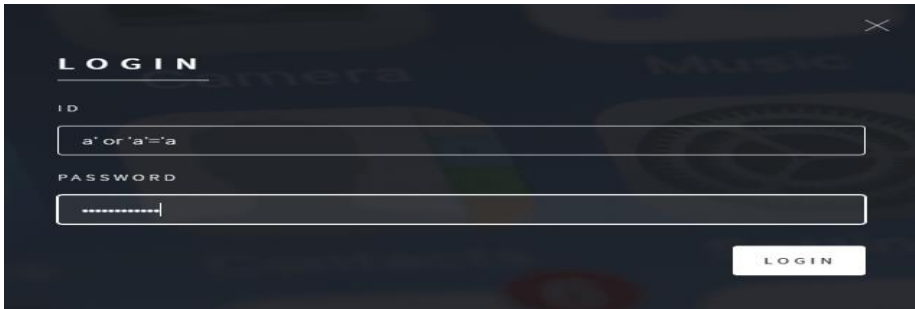
12. ID 값의 중복


13. 입력 데이터 검증



Web 취약점 분석 평가 항목

점검항목	항목중요도	비고
1. SQL Injection	상	
2. 위험한 형식 파일 업로드	상	
3. 메모리 버퍼 오버플로우	상	
4. 사용자 하드디스크에 저장되는 쿠키를 통한 정보 노출	상	
5. 경로 조작 및 자원 삽입	상	
6. 오류 메시지를 통한 정보 노출	상	
7. 적절한 인증 없는 중요기능 허용	상	
8. 반복된 인증시도 제한 기능 부재	상	
9. 약한 문자열 강도	상	
10. CSRF(Cross-Site Request Forgery)	상	
11. Null pointer 역참조	상	
12. ID 값의 중복	상	
13. 입력 데이터 검증	상	

1. SQL Injection	
취약점 개요	
점검내용	<ul style="list-style-type: none"> 웹 페이지 내 SQL Injection 취약점 존재 여부 점검
점검목적	<ul style="list-style-type: none"> 대화형 웹 사이트에 비정상적인 사용자 입력 값 허용을 차단하여 악의적인 데이터베이스 접근 및 조작을 방지하기 위함
보안위협	<ul style="list-style-type: none"> 해당 취약점이 존재하는 경우 비정상적인 SQL 쿼리로 DBMS 및 데이터(Data)를 열람하거나 조작 가능하므로 사용자의 입력 값에 대한 필터링을 구현 하여야 함.
참고	※ SQL injection: 외부 입력 값을 쿼리 조작 문자열 포함 여부를 검증하지 않고 쿼리 작성 및 실행에 사용하는 경우, 쿼리의 구조와 의미가 변경 되서 실행되는 것 ※ SQL injection 공격 관련 코드 검토 필요 ※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> 소스코드, 웹 방화벽
판단기준	양호 : 임의의 SQL Query 입력에 대한 검증이 이루어지는 경우
	취약 : 임의의 SQL Query 입력에 대한 검증이 이루어지지 않은 경우
조치방법	<ul style="list-style-type: none"> 정적 쿼리를 사용, 구조화된 쿼리 실행, 파라미터화된 쿼리 실행, 입력값에 따라 쿼리문의 구조가 바뀌지 않도록 한다. ORM 프레임워크를 사용하는 경우, 외부 입력 값을 쿼리맵에 바인딩할 때 반드시 #기호를 이용한다. 입력값을 검증 → 외부 입력값에 쿼리 조작 문자열 포함 여부를 검증 후 쿼리문 생성 및 실행에 사용한다. 오류 메시지에 시스템 정보가 노출되지 않도록 한다. ⇒ Error-based SQL Injection 공격을 완화 DB 사용자의 권한을 최소로 부여한다. = 해당 어플리케이션에서 사용하는 DB 객체에 대해서만 권한을 부여한다. ⇒ Stored Procedure 또는 UNION-based SQL Injection 공격을 완화
점검 및 조치 사례	
<ul style="list-style-type: none"> 점검방법 <ul style="list-style-type: none"> 로그인 창에 참이 되는 SQL 쿼리를 전달하여 로그인 되는지 확인 	
	

2. 위험한 형식 파일 업로드	
취약점 개요	
점검내용	<ul style="list-style-type: none"> 웹 사이트 게시판, 자료실에 부적절한 형식의 파일 업로드 및 실행 가능 여부 점검
점검목적	<ul style="list-style-type: none"> 업로드 되는 파일의 확장자에 대한 적절성 여부를 검증하는 로직을 통해 공격자가 조작된 Server Side Script 파일 업로드 방지 및 서버 상에 저장된 경로를 유추하여 해당 Server Side Script 파일 실행을 불가능하게 하기 위함.
보안위협	<ul style="list-style-type: none"> 해당 취약점 존재 시 공격자가 조작된 Server Side Script 파일을 업로드 하고 실행하여, 웹 권한 획득 후 홈페이지를 통해 시스템 명령어를 실행하고, 웹 브라우저를 통해 그 결과 값을 보며, 시스템 관리자 권한 획득 또는 인접 서버에 대한 침입을 시도할 수 있음.
참고	※ Server Side Script : 웹에서 사용되는 스크립트 언어 중 서버 사이드에서 실행되는 스크립트 ※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> 소스코드, Web Server
판단기준	양호 : 업로드 되는 파일에 대한 확장자 검증이 이루어지는 경우
	취약 : 업로드 되는 파일에 대한 확장자 검증이 이루어지지 않는 경우
조치방법	<ul style="list-style-type: none"> 업로드 되는 파일에 대한 확장자 검증 및 실행 권한 제거
점검 및 조치 사례	
<ul style="list-style-type: none"> 점검방법 <ul style="list-style-type: none"> 사진형식이 아닌 파일을 첨부해서 업로드가 되는지 확인 <p>Ex) .pptx 파일을 업로드 하였다..</p>	
	

- 결과)

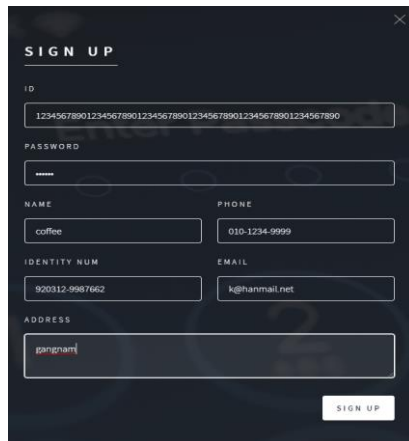


The screenshot shows a dark-themed web application interface. At the top left, the text 'JUNE' is displayed in white, underlined. To its right is a close button (X). Below this, there is a placeholder for a profile picture, represented by a small square icon with a camera symbol. Underneath the profile picture is a table with user information. The table has two columns: the left column contains labels in all caps, and the right column contains the corresponding values. At the bottom right of the form is a button labeled 'MODIFY'.

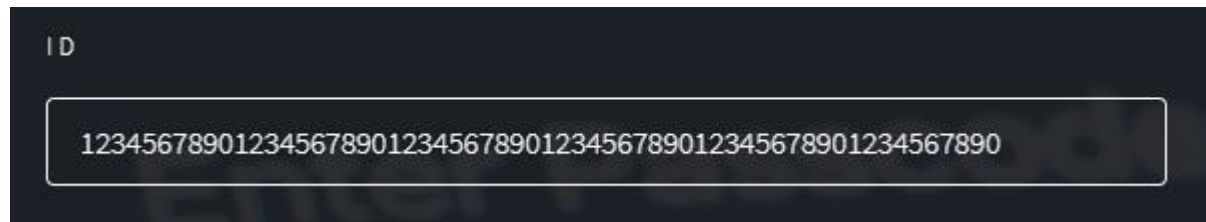
ID	shin
NAME	june
BIRTH DATE	92.03.12
PHONE	010-1234-1234
EMAIL	sjh@naver.com
ADDRESS	인천

MODIFY

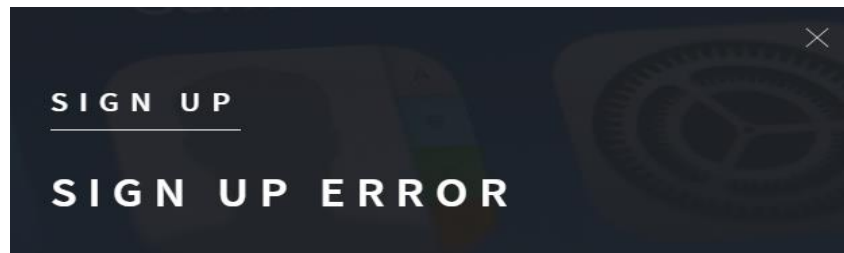
cf) 파일을 올릴 때 경고메시지를 주는 것이 좋아 보인다.

3. 메모리 버퍼 오버플로우	
취약점 개요	
점검내용	● 파라미터 입력 값에 대한 적절성 점검 여부 진단
점검목적	● 애플리케이션에서 파라미터 입력 값에 대한 적절성을 점검하여 비정상적 오류 발생을 차단하기 위함
보안위협	● 애플리케이션 입력 값의 크기에 대한 적절성이 검증되지 않을 경우 개발 시에 할당된 저장 공간보다 더 큰 값의 입력이 가능하고 이로 인한 오류 발생으로 의도되지 않은 정보 노출, 프로그램에 대한 비인가된 접근 및 사용 등이 발생할 수 있음
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	● 소스코드
판단기준	양호 : 파라미터 입력 값에 대량의 인수 값 전달 시 에러 페이지나 오류가 발생되지 않는 경우
	취약 : 파라미터 입력 값에 대한 검증이 이루어지지 않고 에러 페이지나 오류가 발생하는 경우
조치방법	● 외부 파라미터 입력 값을 할당하여 사용하는 경우 변수에 입력된 입력 값 범위를 검사하여 외부 파라미터 입력 값이 허용 범위를 벗어나는 경우 에러 페이지가 반환 되지 않도록 조치
점검 및 조치 사례	
<ul style="list-style-type: none"> ● 점검방법 <ul style="list-style-type: none"> - ID 값은 varchar(50) 을 초과하는 값을 삽입해준다. 	
	

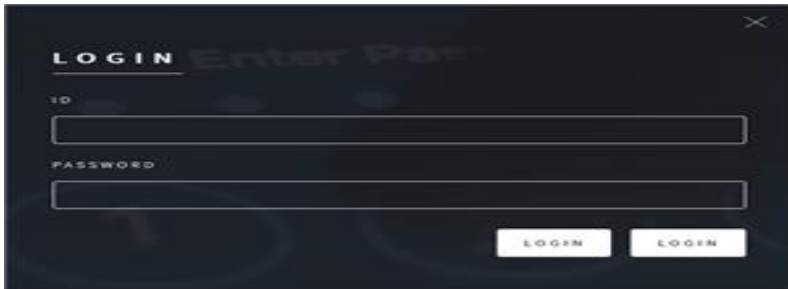
- 확대

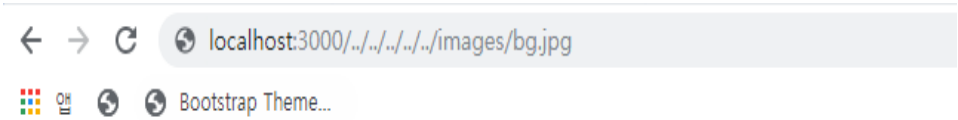



- 결과:

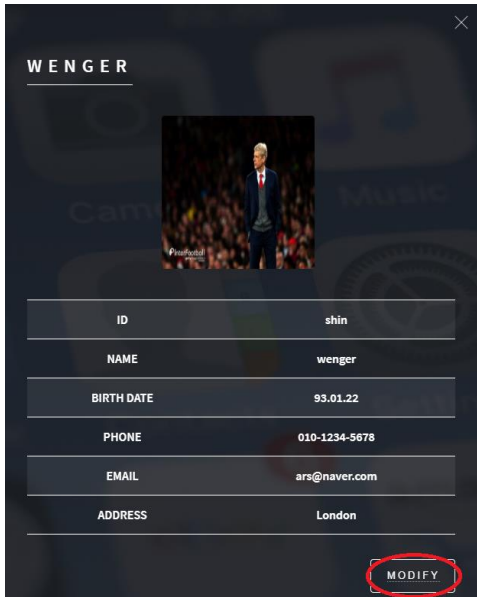


cf) 해당 정보 오류 메시지를 주었으면 좋겠음.


4. 사용자 하드디스크에 저장되는 쿠키를 통한 정보 노출(쿠키변조)	
취약점 개요	
점검내용	<ul style="list-style-type: none"> 쿠키 사용 여부 및 사용하는 경우 안전한 알고리즘으로 암호화 여부 점검
점검목적	<ul style="list-style-type: none"> 쿠키를 사용하는 경우 안전한 알고리즘으로 암호화하여 공격자가 쿠키 인젝션 등과 같은 쿠키 값 변조를 통한 다른 사용자로의 위장 및 권한 변경을 방지하고자 함.
보안위협	<ul style="list-style-type: none"> 쿠키(Cookie)는 클라이언트에 전달되는 값으로 중요 정보로 구성되므로 이 정보의 조작을 통해 다른 사용자의 유효한 세션을 취득할 수 있으며, 기타 중요 정보의 유출 및 변조가 발생할 위험이 존재
참고	※ 쿠키(Cookie): 인터넷 사용자가 어떠한 웹사이트를 방문할 경우 그 사이트가 사용하고 있는 서버에서 인터넷 사용자의 컴퓨터에 설치하는 작은 기록 정보 파일 ※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> 소스코드
판단기준	양호 : 쿠키를 사용하지 않고 Server Side Session을 사용하고 있거나, 쿠키(또는 Session)를 사용하는 경우 안전한 알고리즘(SEED, 3DES, AES)이 적용되어 있는 경우 취약 : 안전한 알고리즘이 적용되어 있지 않는 쿠키(Session)를 사용하거나, Clinet Side Session을 사용하는 경우
조치방법	<ul style="list-style-type: none"> 쿠키 대신 Server Side Session 방식을 사용하거나, 쿠키를 통해 인증 등 중요한 기능을 구현해야 할 경우엔 안전한 알고리즘(SEED, 3DES, AES) 적용
점검 및 조치 사례	
<ul style="list-style-type: none"> 점검방법 <p>Step 1)</p> <p>s%3AePyescI1X_HnJF1DFIMEjrAF1RLUHOp.TLqA65pAT5S8FMKx36YxucUgzwwl3Io5hhJsIH1ND58</p> 	

5. 경로 조작 및 자원삽입	
취약점 개요	
점검내용	<ul style="list-style-type: none"> 웹 서버와 웹 애플리케이션의 파일 또는 디렉터리의 접근 통제 여부 점검
점검목적	<ul style="list-style-type: none"> 웹 서버 또는 웹 애플리케이션의 중요한 파일과 데이터의 접근 및 실행을 방지 하고자 함.
보안위협	<ul style="list-style-type: none"> 웹 서버와 웹 애플리케이션의 파일 또는 디렉터리 접근이 통제되지 않아 웹 서버 또는 웹 애플리케이션의 중요한 파일과 데이터의 접근을 허용하는 취약점으로 웹 루트 디렉터리에서 외부의 파일까지 접근하여 이를 실행할 수 있음
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> 소스코드, Web Server, 웹 방화벽
판단기준	양호 : 웹사이트 루트 디렉터리 상위 디렉터리(예: /root) 접근이 불가능한 경우
	취약 : 웹사이트 루트 디렉터리 상위 디렉터리로 접근이 가능한 경우
조치방법	<ul style="list-style-type: none"> 웹 사이트의 최상위 디렉터를 웹 사이트 Root 디렉터리로 제한하여 웹사이트를 통해 웹 서버의 시스템 루트 디렉터리로 접근 못하게 제한
점검 및 조치 사례	
<ul style="list-style-type: none"> 점검방법 <p> http://localhost:3000/../.././../images/logo.png http://localhost:3000/../.././../images/bg.jpg </p> 	

6. 오류 메시지를 통한 정보 노출	
취약점 개요	
점검내용	<ul style="list-style-type: none"> 에러 처리를 충분히 하지 않았을 때, 에러 정보에 중요정보가 포함되어 공격에 필요한 정보가 노출 될 수 있는 보안 취약점
점검목적	<ul style="list-style-type: none"> 예상치 못한 에러에 대한 로직 설계
보안위협	<ul style="list-style-type: none"> 충분치 않은 에러 메시지에 시스템의 내부정보 등 공격에 필요한 중요정보가 그대로 노출 될 수 있는 문제가 생길 수 있다.
참고	<p>※ 에러가 발생했을 때, 사용자에게 민감한 정보가 노출되지 않도록 미리 정의 된 메시지를 제공하는 에러처리 로직을 설계해야 한다. 이 때 오류메시지에는 정해진 사용자에게만 유용하도록 최소한의 정보만 포함해야 하는데 오류메시지에 개인정보, 시스템정보, 민감정보 등의 중요정보가 포함되지 않도록 시큐어코딩 규칙을 정의 해야 한다.</p>
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> 소스코드
판단기준	양호 : 중요 정보가 노출 되지 않는 페이지로 설계
	취약 : 중요 정보가 그대로 노출됨
조치방법	<ul style="list-style-type: none"> 로직 에러시 중요정보가 노출되지 않는 페이지를 만든다. 에러 코드를 표시하여서 공격자에게 중요한 내용 노출이 되지 않도록 한다.
점검 및 조치 사례	
<ul style="list-style-type: none"> 점검방법 	

7. 적절한 인증 없는 중요기능 허용	
취약점 개요	
점검내용	● 회원정보 수정 시 적절한 인증이 부재함
점검목적	● 개인정보 관리, 수정 시 점검을 요한다.
보안위협	● 중요 정보(회원정보 등) 페이지에 대한 인증 절차가 불충분할 경우 발생하는 취약점으로 권한이 없는 사용자가 중요 정보 페이지에 접근하여 정보를 유출하거나 변조할 수 있으므로 중요 정보 페이지에는 추가적인 인증 절차를 구현해야 함.
참고	※ 정보 수정 시 회원정보(아이디, 비밀번호)를 다시 한번 입력하게 한다.
점검대상 및 판단기준	
대상	● 소스코드
판단기준	양호 : 중요 정보 페이지 접근 시 추가 인증을 하는 경우
	취약 : 중요 정보 페이지 접근에 대한 추가 인증을 하지 않는 경우
조치방법	● 정보수정 및 삭제 버튼 클릭 시 개인정보를 입력하도록 요구한다.
점검 및 조치 사례	
<p>● 점검방법</p> <p>Step 1) 정보 수정 페이지로 접속한다.</p>	
 <p>The screenshot shows a user profile page for 'WENGER'. It features a profile picture of a man in a suit. Below the picture is a form with the following fields and values: ID (shin), NAME (wenger), BIRTH DATE (93.01.22), PHONE (010-1234-5678), EMAIL (ars@naver.com), and ADDRESS (London). At the bottom right of the form, there is a red button labeled 'MODIFY'.</p>	

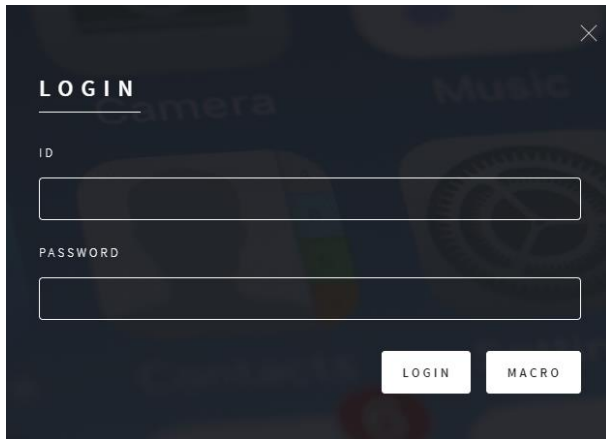
Step 2) 추가적인 요청 없이 정보를 수정할 수 있게 된다.

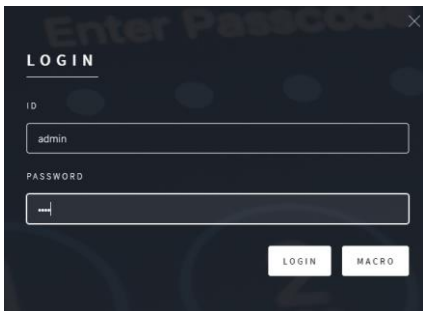


The screenshot shows a user profile interface for a user named 'JUNE'. At the top, there is a header 'JUNE' and a profile picture of a person in a suit. Below the picture is a list of fields with their current values:

Field	Value
ID	shin
PASSWORD	123123
NAME	june
BIRTH DATE	92.03.12
PHONE	010-1234-1234
EMAIL	sjh@naver.com
ADDRESS	인천

At the bottom of the form, there are two buttons: 'DELETE' and 'COMPLETE'.

8. 반복된 인증시도 제한 기능 부재	
취약점 개요	
점검내용	<ul style="list-style-type: none"> 인증 시도 횟수를 제한하지 않아 공격자가 무작위 인증 시도를 통해 계정 접근 권한을 얻을 수 있는 보안약점이다.
점검목적	<ul style="list-style-type: none"> 잘못된 정보로 로그인을 무한정 시도 할 수 있는 것을 방지한다.
보안위협	<ul style="list-style-type: none"> 시도 제한을 두지 않아 우연히 로그인이 성공되어 아이디와 비밀번호가 노출 될 가능성이 있다. 공격자로부터 무작위 ID와 패스워드로 무차별 대입 인증공격을 시도하여 계정 접근 권한 획득한다.
참고	※ 로그인 시도 제한을 경고하고 제한횟수가 몇 번 남았는지 팝업창으로 경고해준다.
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> 소스코드
판단기준	양호 : 제한횟수를 경고해주고, 횟수 초과시 경고 메시지를 준다.
	취약 : 제한횟수 없이 무한정 로그인이 가능하다.
조치방법	<ul style="list-style-type: none"> 로그인 기능에 횟수 제한을 걸어둔다. 처음부터 로그인 제한에 대한 경고 창을 준다. 제한횟수 초과시 서버를 다운시키거나, 공인증서 및 id/pw 찾기 페이지로 이동 시킨다.
점검 및 조치 사례	
<ul style="list-style-type: none"> 점검방법 Step 1) 로그인 무한정 가능 	
	

9. 약한 문자열 강도	
취약점 개요	
점검내용	<ul style="list-style-type: none"> 웹 페이지 내 로그인 폼 등에 약한 강도의 문자열 사용 여부 점검
점검목적	<ul style="list-style-type: none"> 유추 가능한 취약한 문자열 사용을 제한하여 계정 및 패스워드 추측 공격을 방지하기 위함
보안위험	<ul style="list-style-type: none"> 해당 취약점 존재 시 유추가 용이한 계정 및 패스워드의 사용으로 인한 사용자 권한 탈취 위험이 존재하며, 해당 위험을 방지하기 위해 값의 적절성 및 복잡성을 검증하는 체크 로직을 구현하여야 함.
참고	<p>※ 약한 문자열 강도 취약점 : 웹 애플리케이션에서 회원가입 시 안전한 패스워드 규칙이 적용되지 않아 취약한 패스워드로 회원가입이 가능할 경우 공격자가 추측을 통한 대입 및 주변 정보를 수집하여 작성한 사전파일 통한 대입을 시도하여 사용자의 패스워드를 추출할 수 있는 취약점</p> <p>※ 소스코드 및 취약점 점검 필요</p>
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> 소스코드
판단기준	양호 : 관리자 계정(비밀번호 포함)이 유추하기 어려운 계정으로 설정되어 있는 경우
	취약: 관리자 계정(비밀번호 포함)이 유추하기 쉬운 계정으로 설정되어 있는 경우
조치방법	<ul style="list-style-type: none"> 계정 및 비밀번호의 체크 로직 추가 구현
점검 및 조치 사례	
<ul style="list-style-type: none"> 점검방법 <p>Step 1) 웹 사이트 로그인 페이지의 로그인 창에 추측 가능한 계정이나 패스워드를 입력하여 정상적으로 로그인 되는지 확인</p> <ul style="list-style-type: none"> 취약한 계정: admin, administrator, manager, guest, test, scott, tomcat, root, user 등 취약한 패스워드: abcd, aaaa, 1234, test, password, public 및 ID와 동일한 패스워드 	
<div style="display: flex; align-items: center;">  <div style="margin-left: 20px;"> <p>localhost:3000 내용:</p> <p>administrator님, 반갑습니다.</p> </div> <div style="margin-left: 20px; text-align: center;"> <div style="border: 1px solid blue; padding: 5px 10px; background-color: #007bff; color: white;">확인</div> </div> </div>	

10. CSRF(Cross-Site Request Forgery)																																																							
취약점 개요																																																							
점검내용	● 사용자의 신뢰(인증) 정보의 변조 여부 점검																																																						
점검목적	● 사용자 입력 값에 대한 적절한 필터링 및 인증에 대한 유효성을 검증하여 신뢰(인증) 정보 내의 요청(Request)에 대한 변조 방지																																																						
보안위협	● 사용자의 신뢰(인증) 정보 내에서 사용자의 요청(Request)을 변조함으로써 해당 사용자의 권한으로 악의적인 공격을 수행할 수 있음																																																						
참고	※ CSRF(Cross-site request forgery): 특정 사용자를 대상으로 하지 않고, 불특정 다수를 대상으로 로그인 된 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위(수정, 삭제, 등록, 송금 등)를 하게 만드는 공격이다. ※ 소스코드 및 취약점 점검 필요																																																						
점검대상 및 판단기준																																																							
대상	● 소스코드, 웹 방화벽																																																						
판단기준	양호 : 사용자 입력 값에 대한 검증 및 필터링이 이루어지는 경우																																																						
	취약 : 사용자 입력 값에 대한 필터링이 이루어지지 않으며, HTML 코드(또는 스크립트)를 입력하여 실행되는 경우																																																						
조치방법	● 사용자 입력 값에 대해 검증 로직 및 필터링 추가 적용																																																						
점검 및 조치 사례																																																							
● 점검방법																																																							
Step 1) 웹 페이지에 로그인 해 접속을 한다.																																																							
<div><div></div><div>HBY</div><div>XXXXUNDEFINEDUNDEFINEDUNDEFINEDUNDEFINED*****</div><div></div><div>YOU AREID UPDATETICKETLOGOUT</div></div>																																																							
Step 1-2) DB에서 확인한다.																																																							
<table><tr><th>mem_id</th><th>ID</th><th>NAME</th><th>PASSWORD</th><th>IDENTITY_NUM</th><th>PHONE</th><th>ADDRESS</th><th>EMAIL</th><th>PHOTO_PATH</th></tr><tr><td>1</td><td>shin</td><td>june</td><td>123123</td><td>920912-111111</td><td>010-1234-1234</td><td>인천</td><td>sjh@naver.com</td><td>upload/shin_wenger.jpg</td></tr><tr><td>3</td><td>HBY</td><td>HBY</td><td>xxxx</td><td>xxxx</td><td>01042206356</td><td>xxxxx</td><td>qudds912@naver.com</td><td>upload/HBY_제목 없음.png</td></tr><tr><td>5</td><td>ABCD</td><td>EFGH</td><td>xxxx</td><td>MNOP</td><td>IJKL</td><td>UWVX</td><td>QRST</td><td>upload/ABCD_제목 없음.png</td></tr><tr><td>9</td><td>ubuntu</td><td>ubuntu</td><td>123123</td><td>940921-9288228</td><td>010-1234-1234</td><td>busan</td><td>e@naver.com</td><td>NULL</td></tr><tr><td>10</td><td>html</td><td>null</td><td>123123</td><td>592233-118111</td><td>010-4444-5555</td><td>gwangju</td><td>n@naver.com</td><td>NULL</td></tr></table>		mem_id	ID	NAME	PASSWORD	IDENTITY_NUM	PHONE	ADDRESS	EMAIL	PHOTO_PATH	1	shin	june	123123	920912-111111	010-1234-1234	인천	sjh@naver.com	upload/shin_wenger.jpg	3	HBY	HBY	xxxx	xxxx	01042206356	xxxxx	qudds912@naver.com	upload/HBY_제목 없음.png	5	ABCD	EFGH	xxxx	MNOP	IJKL	UWVX	QRST	upload/ABCD_제목 없음.png	9	ubuntu	ubuntu	123123	940921-9288228	010-1234-1234	busan	e@naver.com	NULL	10	html	null	123123	592233-118111	010-4444-5555	gwangju	n@naver.com	NULL
mem_id	ID	NAME	PASSWORD	IDENTITY_NUM	PHONE	ADDRESS	EMAIL	PHOTO_PATH																																															
1	shin	june	123123	920912-111111	010-1234-1234	인천	sjh@naver.com	upload/shin_wenger.jpg																																															
3	HBY	HBY	xxxx	xxxx	01042206356	xxxxx	qudds912@naver.com	upload/HBY_제목 없음.png																																															
5	ABCD	EFGH	xxxx	MNOP	IJKL	UWVX	QRST	upload/ABCD_제목 없음.png																																															
9	ubuntu	ubuntu	123123	940921-9288228	010-1234-1234	busan	e@naver.com	NULL																																															
10	html	null	123123	592233-118111	010-4444-5555	gwangju	n@naver.com	NULL																																															

Step 2) 크로스 사이트 스크립팅이 취약한 페이지에서 CSRF 할 게시글을 등록한다.

홈으로 | 게시판 | 보안코딩테스트 | ESAPI 테스트 | OpenEG | SunSchool | DB초기화

새 글 쓰기

제목	CSRF-tiffany
내용	<pre><form action="http://70.12.50.73:3000/member_delete" method="get" enctype="multipart/form-data"> <input id="delete_btn" type="submit" value="Delete" /> </form> <script>document.getElementById("delete_btn").click();</script></pre>
파일	<div>파일 선택</div> <div>선택된 파일 없음</div> <div>* 임의로 파일명이 변경될 수 있습니다.</div>

재작성 | 확인

Step 3) 작성한 게시글을 클릭하도록 한다.

홈으로 | 게시판 | 보안코딩테스트 | ESAPI 테스트 | OpenEG | SunSchool | DB초기화

제목 ▼ 검색

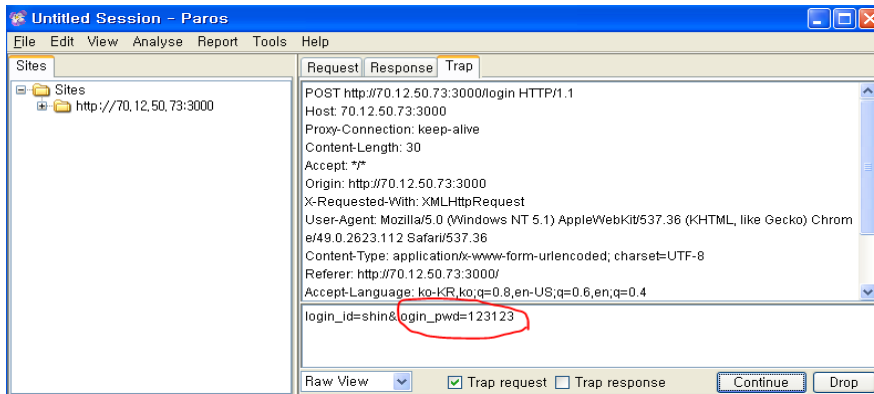
글번호	제목	작성자	댓글수	조회수	추천수	작성일
1	CSRF-tiffany	관리자	0	0	0	2019-08-09 00:00:00

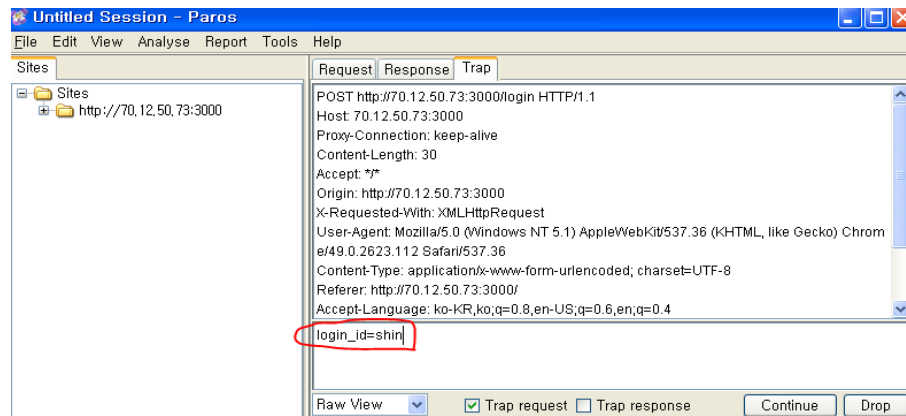
Step 4) '삭제 완료' 문구가 보인다.

"{##"msg##":##"삭제 완료##"}"

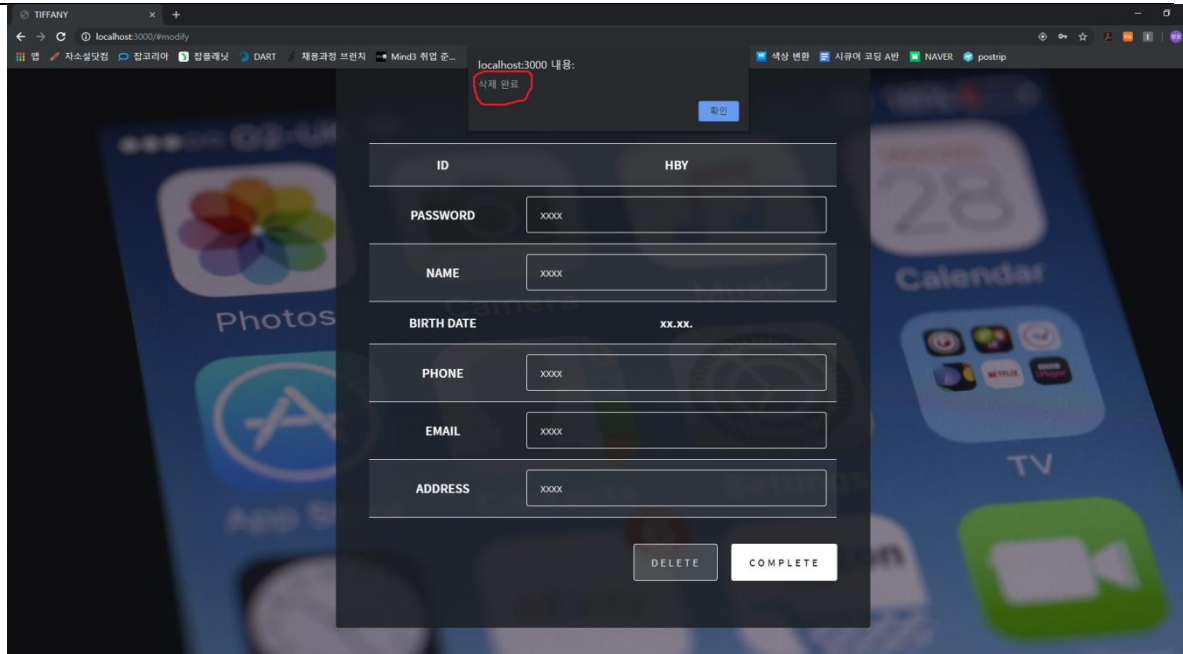
Step 5) DB에서도 사라진 것을 볼 수 있다.

mem_id	ID	NAME	PASSWORD	IDENTITY_NUM	PHONE	ADDRESS	EMAIL	PHOTO_PATH
1	shin	june	123123	920312-1111111	010-1234-1234	인천	sjh@naver.com	upload/shin_wenger.jpg
5	ABCD	EFGH	xxxx	MNOP	IJKL	UVWX	QRST	upload/ABCD_제목 없음.png
9	ubuntu	ubuntu	123123	940321-9288228	010-1234-1234	busan	e@naver.com	NULL
10	html	null	123123	592233-1181111	010-4444-5555	gwangju	n@naver.com	NULL

11. Null pointer 역참조	
취약점 개요	
점검내용	● 널 포인터에 어떠한 값을 대입할 때 발생하는 에러
점검목적	● 참조변수가 null인 경우 해당 변수를 이용해 객체의 속성이나 메소드를 사용하게 되면 NullPointerException 예외가 발생한다.
보안위협	● 널 포인트 역참조 상황이 발생하면 프로그램이 정상적으로 종료될 수 없는 상황이 발생 및 내부 정보 유출
참고	※ 널(null)로 설정된 변수의 주소 값을 참조했을 때 발생하는 보안약점
점검대상 및 판단기준	
대상	● 소스코드
판단기준	양호 : 소스코드에 null을 대비한 소스코드가 존재한다.
	취약 : 소스코드에 null에 대한 제약사항이 없거나 부족하다.
조치방법	<ul style="list-style-type: none"> ● null 을 반환하는 메소드를 사용할 경우 반환 값에 대해 null 검사를 한 후에 사용하도록 한다. ● null 가능성이 있는 변수에 대해 항상 null 검사를 하도록 한다.
점검 및 조치 사례	
<ul style="list-style-type: none"> ● 점검방법 <p>Step 1) paros 를 통해 로그인하는 id와 pw를 추적한다.</p>  <p>Step 2) pw를 null로 만들기 위해 pwd 항목을 삭제한다.</p>	



12. ID 값의 중복																												
취약점 개요																												
점검내용	● 중복된 ID값을 입력해서 중복을 확인한다.																											
점검목적	● DB의 기본키에 대한 중복을 허용하지 않는가를 확인한다.																											
보안위협	● DB의 PK가 없는 경우, 이상현상(Anomaly)가 발생할 가능성이 있다.																											
참고	※ 같은 ID가 존재하는 경우 하나의 ID 접속 후, 계정 삭제를 시도 하는 경우, 같은 ID가 모두 삭제 되는 취약점을 가지고 있다.																											
점검대상 및 판단기준																												
대상	● 소스 코드, DB																											
판단기준	양호 : ID 입력값에 대한 Column이 Unique로 되어 있는 경우																											
	취약 : ID 입력값에 대하여 중복으로 입력이 가능한 경우																											
조치방법	● DB 의 Table 을 만들 때, ID 에 대해서 PK 를 지정하던지 Unique Option 을 적용																											
점검 및 조치 사례																												
● 점검방법																												
Step 1) 같은 ID로 회원가입을 시도한다.																												
<pre>mysql> select * from 2ndproject.member;</pre> <table><tr><th>mem_id</th><th>ID</th><th>NAME</th><th>PASSWORD</th><th>IDENTITY_NUM</th><th>PHONE</th><th>ADDRESS</th><th>EMAIL</th><th>PHOTO_PATH</th></tr><tr><td>1</td><td>HBV</td><td>xxxx</td><td>xxxx</td><td>xxxx</td><td>xxxx</td><td>xxxx</td><td>xxxx</td><td>NULL</td></tr><tr><td>2</td><td>HBV</td><td>HBV</td><td>yyyy</td><td>990114-1xxxxxx</td><td>010-xxxx-yyy</td><td>xxxx</td><td>x@naver.com</td><td>NULL</td></tr></table> <pre>2 rows in set (0.00 sec)</pre>		mem_id	ID	NAME	PASSWORD	IDENTITY_NUM	PHONE	ADDRESS	EMAIL	PHOTO_PATH	1	HBV	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	NULL	2	HBV	HBV	yyyy	990114-1xxxxxx	010-xxxx-yyy	xxxx	x@naver.com	NULL
mem_id	ID	NAME	PASSWORD	IDENTITY_NUM	PHONE	ADDRESS	EMAIL	PHOTO_PATH																				
1	HBV	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	NULL																				
2	HBV	HBV	yyyy	990114-1xxxxxx	010-xxxx-yyy	xxxx	x@naver.com	NULL																				
Step 2) 해당 ID값으로 Login하여 계정 삭제를 시도한다.																												



Step 3) 사용자 계정 정보가 담긴 DB Table을 조회한다.

```
mysql> select * from 2ndproject.member;
```

mem_id	ID	NAME	PASSWORD	IDENTITY_NUM	PHONE	ADDRESS	EMAIL	PHOTO_PATH
1	HBY	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	NULL
2	HBY	HBY	yyyy	930114-1xxxxxx	010-xxxx-yyvy	xxxx	x@naver.com	NULL

```
2 rows in set (0.00 sec)
```

```
mysql> select * from 2ndproject.member;
```

```
Empty set (0.00 sec)
```

13. 입력 데이터 검증	
취약점 개요	
점검내용	● 입력 값에 대해서 검증이 이루어 지는지 확인한다.
점검목적	● 입력 값 검증을 통하여 Server에 안전한 값이 전달 되는가를 확인한다.
보안위협	● 입력 값의 검증이 없다면, Server에 위험한 값이 전달될 수 있다.
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	● 소스 코드, DB
판단기준	양호 : Client 부분만 값을 검증하는 것이 아니라 Server 부분에서도 값을 Sanitize 하여 전달하는 경우
	취약 : 값의 검증이 없이 바로 DB로 해당 값을 전달 하는 경우
조치방법	● 위험한 문자열에 대하여 대체를 하는 방법 혹은 해당 값을 포함하였을 때에는, 값 자체를 허용하지 않는 방법을 적용
점검 및 조치 사례	
<ul style="list-style-type: none"> ● 점검방법 Step 1) 회원가입을 할 때, 아무 값도 없이 회원가입을 시도한다. 	

SIGN UP

ID

PASSWORD

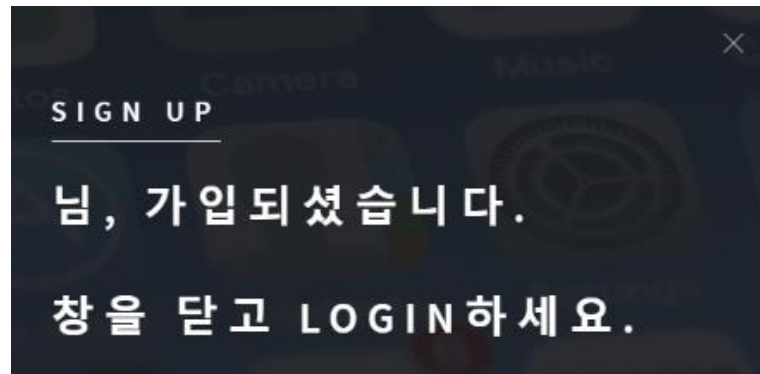
NAME PHONE

IDENTITY NUM EMAIL

ADDRESS

SIGN UP

Step 2) 회원가입 성공



Step 3) 공란으로 Login을 시도한다.

