항목	내용
취약점	버퍼 오버플로우
취약점 설명	메모리를 다루는 데에 오류가 발생하여 잘못된 동작을 하는 프로그램 취약점이다. 컴퓨터 보안과 프로그래밍에서는 프로세스가 데이터를 버퍼에 저장할 때 프로그래머가 지정한 곳 바깥에 저장하는 것을 의미한다.
취약점 발생 원인	웹 사이트에서 사용자가 입력한 파라미터 값의 문자열 길이를 제한하지 않는 경우 개발 시에 할당된 저장 공간보다 더 큰 값의 입력이 가능하고 이로 인한 오류 발생 시 의도되지 않은 정보 노출, 프로그램에 대한 비인가 접근 및 사용 등이 발생할 수 있음
위험 시나리오	
조치 방법	파라미터 값을 외부에서 입력받아 사용하는 경우 입력 값 범위를 제한하며, 허용 범위를 벗어나는 경우 에러 페이지가 반환되지 않도록 조치

항목	내용
취약점	포맷스트링
취약점 설명	포맷스트링과 이것을 사용하는 printf() 함수의 취약점을 이용하여 RET의 위치에 셸 코드의 주소를 읽어 셸을 획득하는 해킹 공격이다.
취약점 발생 원인	C언어로 만드는 프로그램 중 변수의 값을 출력하거나 입력받을 때 입력받은 값을 조작하여 프로그램의 메모리 위치를 반환받아 메모리 주소를 변조하여 시스템의 관리자 권한을 획득할 수 있음
위험 시나리오	
조치 방법	웹 서버 프로그램을 최신 버전으로 업데이트하고 포맷 스트링 버그를 발생시키는 문자열에 대한 검증 로직 구현

항목	내용
취약점	LDAP 인젝션
취약점 설명	LDAP(Lightweight Directory Access Protocol)에 대한 Injection 공격으로 사용자의 입력 값이 LDAP Query에 직접 영향을 끼칠 수 있을 때 이를 통해 비정상적인 LDAP 동작을 유도하는 공격 방법이다.
취약점 발생 원인	응용 프로그램이 사용자 입력 값에 대한 적절한 필터링 및 유효성 검증을 하지 않아 공격자는 로컬 프록시를 사용함으로 LDAP 문의 변조가 가능함
위험 시나리오	
조치 방법	지정된 문자열만 입력 허용하고, 임의의 LDAP 쿼리 입력에 대한 검증 로직 구현

항목	내용

취약점	운영체제 명령 실행
취약점 설명	웹 어플리케이션에서 system(), exec()와 같은 시스템 명령어를 실행 시킬 수 있는 함수를 제공하며 사용자 입력 값에 대한 필터링이 제대로 이루어지지 않을 경우 공격자가 운영체제 시스템 명령어를 호출하여 백도어 설치나 관리자 권한 탈취 등 시스템 보안에 심각한 영향을 미칠 수 있는 취약점이다.
취약점 발생 원인	OS command Injection은 웹 서버에 OS 명령을 실행하기 위해 웹 인터페이스를 사용한다. 사용자는 웹 인터페이스를 통해 OS 명령을 실행하기 위하여 운영체제 명령어를 입력한다. 만일 이 명령어가 제대로 필터링 되지 않는다면 웹 인터페이스는 해당 공격에 취약할 수 있다.
위험 시나리오	
조치 방법	취약한 버전의 웹 서버 및 웹 애플리케이션 서버는 최신 버전으로 업데이트를 적용해야 하며, 애플리케이션은 운영체제로부터 명령어를 직접적으로 호출하지 않도록 구현하는 게 좋지만, 부득이하게 사용해야 할 경우 소스 코드나 웹 방화벽에서 특수문자, 특수 구문에 대한 검증을 할 수 있도록 조치해야 함

항목	내용
취약점	SQL 인젝션
취약점 설명	사용자의 입력 값으로 웹 사이트 SQL 쿼리가 완성되는 약점을 이용하며, 입력 값을 변조하여 비정상적인 SQL 쿼리를 조합하거나 실행하는 공격. 개발자가 생각지 못한 SQL문을 실행되게 함으로써 데이터베이스를 비정상적으로 조작 가능함
취약점 발생 원인	사용자의 입력 값으로 SQL 쿼리가 완성되는 약점을 이용하여, 입력 값을 변조한 후 비정상적인 SQL 쿼리를 조합하거나 실행한다.
위험 시나리오	
조치 방법	소스코드에 SQL 쿼리를 입력 값으로 받는 함수나 코드를 사용할 경우, 임의의 SQL 쿼리 입력에 대한 검증 로직을 구현하여 서버에 검증되지 않는 SQL 쿼리 요청 시 에러 페이지가 아닌 정상 페이지가 반환되도록 필터링 처리하고 웹 방화벽에 SQL 인젝션 관련 룰셋을 적용하여 SQL 인젝션 공격을 차단함

항목	내용
취약점	SSI 인젝션
취약점 설명	HTML 문서 내 입력받은 변수 값을 서버 측에서 처리할 때 부적절한 명령문이 포함 및 실행되어 서버의 데이터가 유출되는 취약점
취약점 발생 원인	공통 SSI 구현은 외부의 파일을 Include 할 수 있는 명령어를 제공하며,

	웹 서버의 CGI 환경 변수를 설정하고 출력할 수 있고, 외부의 CGI 스크립트나 시스템 명령어들을 실행할 수 있으므로 부적절한 명령문이 포함 및 실행될 수 있다.
위험 시나리오	
조치 방법	사용자 입력 값에 대한 검증 로직 추가 구현

항목	내용
취약점	XPath 인젝션
취약점 설명	데이터 베이스와 연동된 웹 어플리케이션에서 XPath 및 XQuery 질의문에 대한 필터링이 제대로 이루어지지 않을 경우 공격자가 입력이 가능한 폼(웹 브라우저 주소입력창 또는 로그인 폼 등)에 조작된 질의문을 삽입하여 인증 우회를 통해 XML 문서로부터 인가되지 않은 데이터를 열람할 수 있는 취약점
취약점 발생 원인	사용자 입력값 검증의 부재
위험 시나리오	
조치 방법	쿼리 입력 값에 대해 검증 로직 추가 구현

항목	내용
취약점	디렉터리 인덱싱
취약점 설명	특정 디렉터리에 초기 페이지 (index.html, home.html, default.asp 등)의 파일이 존재하지 않을 때 자동으로 디렉터리 리스트를 출력하는 취약점
취약점 발생 원인	
위험 시나리오	
조치 방법	웹 서버 설정을 변경하여 디렉터리 파일 리스트가 노출되지 않도록 설정

항목	내용
취약점	정보 누출
취약점 설명	웹 사이트의 민감한 정보가 노출되는 것으로 개발 과정의 코멘트나 에러 메시지 등에서 의도하지 않게 정보가 노출되는 취약점
취약점 발생 원인	
위험 시나리오	
조치 방법	웹 사이트에 노출되는 중요정보는 마스킹을 적용하여야 하며, 발생 가능한 에러에 대해 최소한의 정보 또는 사전에 준비된 메시지만 출력함

항목	내용
취약점	악성 콘텐츠
취약점 설명	악성콘텐츠가 삽입된 페이지에 접속한 사용자는 악성코드 유포 사이트가 자동으로 호출되어 악성코드에 감염될 수 있는 취약점
취약점 발생 원인	사용자 입력값 검증의 부재
위험 시나리오	
조치 방법	사용자 입력 값에 대한 검증 로직 추가 및 실행 제한 설정

항목	내용
취약점	크로스사이트 스크립팅
취약점 설명	악의적인 사용자가 공격하려는 사이트에 스크립트를 넣는 기법으로 공격 방식은 크게 stored 공격 방식과 reflected 공격 방식으로 나누어 짐
취약점 발생 원인	웹 애플리케이션에서 사용자 입력 값에 대한 필터링이 제대로 이루어지지 않을 경우, 공격자는 사용자 입력 값을 받는 게시판, URL 등에 악의적인 스크립트(Javascript, VBScript, ActiveX, Flash 등)를 삽입하여 게시글이나 이메일을 읽는 사용자의 쿠키(세션)를 탈취하여 도용하거나 악성코드 유포 사이트로 Redirect 할 수 있음
위험 시나리오	
조치 방법	웹 사이트의 게시판, 1:1 문의, URL 등에서 사용자 입력 값에 대해 검증 로직을 추가하거나 입력되더라도 실행되지 않게 하고, 부득이하게 웹페이지에서 HTML을 사용하는 경우 HTML 코드 중 필요한 코드에 대해서만 입력되게 설정

항목	내용
취약점	약한 문자열 강도
취약점 설명	웹 사이트에서 취약한 패스워드로 회원가입이 가능할 경우 공격자는 추측 및 주변 정보를 수집하여 작성한 사전 파일로 대입을 시도하여 사용자 계정을 탈취할 수 있는 취약점
취약점 발생 원인	유추가 용이한 계정 및 패스워드의 사용
위험 시나리오	
조치 방법	계정 및 비밀번호의 체크 로직 추가 구현

항목	내용
취약점	불충분한 인증
취약점 설명	중요정보(개인정보 변경 등) 페이지에 대한 인증 절차가 불충분할 경우 권한이 없는 사용자가 중요정보 페이지에 접근하여 정보를 유출하거나 변조할 수 있다.

취약점 발생 원인	중요정보(개인정보 변경 등) 페이지에 대한 인증 절차 불충분
위험 시나리오	
조치 방법	중요정보 페이지에 대한 추가 인증 로직 추가 구현

항목	내용
취약점	취약한 패스워드 복구
취약점 설명	취약한 패스워드 복구 로직(패스워드 찾기 등)으로 인하여 공격자가 불법적으로 다른 사용자의 패스워드를 획득, 변경할 수 있음
취약점 발생 원인	취약한 패스워드 복구 로직
위험 시나리오	
조치 방법	패스워드 복구 로직을 변경하고 인증된 사용자 메일이나 SMS에서만 재설정된 패스워드를 확인할 수 있도록 조치

항목	내용
취약점	크로스사이트 리퀘스트 변조(CSRF)
취약점 설명	사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위(수정, 삭제, 등록 등)를 특정 웹 사이트에 요청하게 하는 공격 유형
취약점 발생 원인	사용자의 신뢰(인증) 정보 내에서 사용자의 요청(Request)을 변조함으로써해당 사용자의 권한으로 악의적인 공격을 수행할 수 있음
위험 시나리오	
조치 방법	사용자 입력 값에 대해 검증 로직 및 필터링 추가 적용

항목	내용
취약점	세션 예측
취약점 설명	숫자가 증가하는 방법이나 일정 패턴 등을 활용하여 공격자가 세션의 ID를 예측하여 세션을 가로챌 수 있는 취약점
취약점 발생 원인	사용자에게 전달하는 세션 ID가 일정한 패턴을 가지고 있는 경우 공격자가 세션 ID를 추측하여 불법적인 접근을 시도할 수 있음
위험 시나리오	
조치 방법	추측 불가능한 세션 ID가 발급되도록 로직 구현

항목	내용
취약점	불충분한 인가
취약점 설명	페이지 접근을 위한 인증기능이 구현되지 않을 경우, 해커나 인가되지 않는 사용자가 페이지에 접근 및 중요 정보의 변조를 할 수 있는 취약점

취약점 발생 원인	접근제어가 필요한 중요 페이지의 통제수단이 미흡한 경우, 비인가자가 URL 파라미터 값 변경 등의 방법으로 중요 페이지에 접근하여 민감한 정보 열람 및 변조 가능함
위험 시나리오	
조치 방법	접근제어가 필요한 모든 페이지에 권한검증 로직 구현

항목	내용
취약점	불충분한 세션 만료
취약점 설명	세션의 값이 계속해서 존재하는 취약점
취약점 발생 원인	세션의 만료 기간을 정하지 않거나, 만료기한을 너무 길게 설정된 경우 악의적인 사용자가 만료되지 않은 세션을 활용하여 불법적인 접근이 가능할 수 있음
위험 시나리오	
조치 방법	세션 종료 시간 설정 또는 자동 로그아웃 기능 구현(세션 종료 시간은 사이트의 특성에 따라 달라질 수 있으므로 사이트의 특성에 맞게 적정 시간 설정)

항목	내용
취약점	세션 고정
취약점 설명	사용자 로그인 시 항상 일정하게 고정된 세션ID가 발급되는 경우 해커의 세션 ID 탈취로부터 비인가자의 접근 및 권한 우회가 가능한 취약점
취약점 발생 원인	사용자 로그인 시 항상 일정하게 고정된 세션 ID가 발행되는 경우 세션 ID를 도용한 비인가자의 접근 및 권한 우회가 가능
위험 시나리오	
조치 방법	사용자가 로그인할 때마다 예측 불가능한 새로운 세션 ID 생성 로직 구현하고 기존 세션 ID는 파기함

항목	내용
취약점	자동화 공격
취약점 설명	웹 애플리케이션에 자동화된 공격을 수행하여 많은 수의 프로세스를 실행 시키는 취약점
취약점 발생 원인	웹 애플리케이션의 특정 프로세스에 대한 반복적인 요청을 통제하지 않을 경우 무차별 대입 공격으로 인해 사용자 계정을 탈취할 수 있고, 자동화 공격으로 게시글 등록 또는 SMS 발송 요청을 반복하여 웹 애플리케이션 자원을 고갈시킬 수 있음
위험 시나리오	
조치 방법	웹 애플리케이션의 특정 프로세스에 대한 대량 사용 통제 로직 구현 및

	웹 방화벽 룰셋 설정을 통해 대량의 불특정 프로세스 요청 차단

항목	내용
취약점	프로세스 검증 누락
취약점 설명	인증이 필요한 페이지에 대해 인가된 인원인지를 확인하는 기능이 존재하지 않는 경우에 해당 정보를 변조하거나 탈취 할 수 있는 취약점
취약점 발생 원인	인증이 필요한 웹 사이트의 중요(관리자 페이지, 회원변경 페이지 등) 페이지에 대한 접근 제어가 미흡할 경우 하위 URL 직접 접근, 스크립트 조작 등의 방법으로 중요한 페이지에 대한 접근이 가능함
위험 시나리오	
조치 방법	인증이 필요한 페이지의 경우 페이지별 권한 체크 로직 구현

항목	내용
취약점	파일 업로드
취약점 설명	파일 업로드 기능이 존재하는 웹 사이트의 확장자 필터링이 미흡할 경우, 공격자가 악성 파일을 업로드하여 시스템을 장악할 수 있는 취약점
취약점 발생 원인	조작된 Server Side Script 파일을 서버에 업로드 및 실행하여 시스템 관리자 권한 획득 또는 인접 서버에 대한 침입을 시도할 수 있음
위험 시나리오	
조치 방법	업로드되는 파일에 대한 확장자 검증 및 실행 권한 제거

항목	내용
취약점	파일 다운로드
취약점 설명	파일 다운로드 기능 사용 시 임의의 문자나 주요 파일의 입력을 통해 웹 서버의 홈 디렉터리를 벗어나 임의의 위치에 있는 파일을 열람하거나 다운 가능한 취약점
취약점 발생 원인	파일 다운로드 시 애플리케이션의 파라미터 값을 조작하여 웹 사이트의 중요한 파일(DB 커넥션 파일, 애플리케이션 파일 등) 또는 웹 서버 루트에 있는 중요한 설정 파일(passwd, shadow 등)을 다운받을 수 있음
위험 시나리오	
조치 방법	다운로드 시 허용된 경로 이외의 디렉터리와 파일에 접근할 수 없도록 구현

항목	내용
취약점	관리자 페이지 노출
취약점 설명	관리자 페이지가 추측 가능한 형태로 구성되어 있을 경우 공격자가 관

	리자 페이지에 쉽게 접근할 수 있으며 무차별 대입 공격을 통해 관리자 권한을 획득할 수 있는 취약점
취약점 발생 원인	유추하기 쉬운 URL로 관리자 페이지 및 메뉴 접근의 가능
위험 시나리오	
조치 방법	유추하기 어려운 이름(포트 번호 변경 포함)으로 관리자 페이지를 변경하여 비인가자가 관리자 페이지에 접근할 수 없도록 하고 근본적인 해결을 위해 지정된 IP만 관리자 페이지에 접근할 수 있도록 제한하여야 함 단, 부득이하게 관리자 페이지를 외부에 노출해야 하는 경우 관리자 페이지 로그인 시 2차 인증(otp, vpn, 인증서 등) 적용 필요함

항목	내용
취약점	경로 추적
취약점 설명	웹 서버와 웹 애플리케이션의 파일 또는 디렉터리 접근이 통제되지 않아 웹 서버 또는 웹 애플리케이션의 중요한 파일과 데이터에 접근을 허용하는 취약점
취약점 발생 원인	
위험 시나리오	
조치 방법	사용자가 임의로 접근할 수 있는 최상위 디렉터리를 웹 루트 디렉터리로 설정하여 웹 서버의 시스템 루트 디렉터리로 접근하지 못하게 제한

항목	내용
취약점	위치 공개
취약점 설명	개발 시 사용한 테스트 파일, 애플리케이션(아파치, IIS, 톰캣 등) 설치 시 기본적으로 설치되는 관리자 페이지, 샘플 페이지 및 매뉴얼 페이지 등을 삭제하지 않아 발생하는 취약점
취약점 발생 원인	폴더나 파일명의 위치가 예측 가능하여 쉽게 노출될 경우 공격자는 이를 악용하여 대상에 대한 정보를 획득하고 민감한 데이터에 접근 가능
위험 시나리오	
조치 방법	웹 루트 디렉터리 이하 모든 불필요한 파일 및 샘플 페이지 삭제

항목	내용
취약점	데이터 평문 전송
취약점 설명	서버와 클라이언트 간 통신 시 중요 정보(계정정보, 주민등록번호, 신용정보 등)가 평문으로 노출되는 취약점
취약점 발생 원인	웹 상의 데이터 통신은 대부분 텍스트 기반으로 이루어지기 때문에 서버와 클라이언트 간에 암호화 프로세스를 구현하지 않으면 간단한 도청(Sniffing)을 통해 정보를 탈취 및 도용할 수 있음

위험 시나리오	
조치 방법	사이트의 중요정보 전송구간(로그인, 회원가입, 회원정보관리, 게시판 등) 암호화 통신(https, 애플리케이션방식) 적용

항목	내용
취약점	쿠키 변조
취약점 설명	보호되지 않은 쿠키를 사용하여 쿠키 인젝션 등과 같은 쿠키 값 변조를 통한 데이터의 위변조가 가능한 취약점
취약점 발생 원인	클라이언트에 전달되는 쿠키에 사용자 식별 값이 평문으로 노출될 경우 쿠키 변조를 통해 다른 사용자의 유효한 세션을 취득할 수 있으며, 기타 중요정보의 유출 및 변조 가능함
위험 시나리오	
조치 방법	쿠키 대신 Server Side Session 방식을 사용하거나, 쿠키를 통해 인증 등 중요한 기능을 구현해야 할 경우엔 안전한 알고리즘(SEED, 3DES, AES 등) 적용