점검 항목별 성능 지표

1. YourCode-X 서비스 개요

▫ YourCode-X는 로컬, 개발, 스테이징 서버를 대상으로 실질적인 서비스 운영 전 단계라고 볼 수 있는 서버들을 **정적 • 동적 분석**하여 **중요 정보나 취약한 부분들이 외부에 노출되지 않게 예방해주는 서비스**

▫ 웹 취약점 도구는 다양하게 많으나 보안 관련 종사자가 아닌 이상 개발자들은 쉽게 보안적 측면에 접근이 불가하기에 이를 해결하고자 함

2. YourCode-X 점검 평가 항목

▫ **[X]: 위험, [Δ]: 주의, [O]: 양호 기준으로 분석 결과가 도출 [단계별 세분화 진행 예정(1~5단계)]**

▫ KISA 주요통신기반시설 상세 가이드(2021), OWASP TOP 10(2021, 2017)
, Reflectiz OWASP TOP 10(2023)

점검 항목	목록	상세 내용
SQL 인젝션 (SQL Injection)	**설명**	악의적인 사용자가 웹 애플리케이션의 데이터베이스 쿼리에 임의의 SQL 코드를 삽입하는 공격. 이를 통해 공격자는 데이터베이스의 정보를 임의로 조회, 수정, 삭제 등을 할 수 있습니다.
	원인	1. SQL 인젝션(SQL Injection) 공격은 주로 사용자 입력값을 쿼리문에 직접 삽입할 때 발생합니다. 2. 웹 애플리케이션에서 사용자 입력 값에 대한 적절한 검증 및 처리가 이루어지지 않을 경우 발생할 수 있습니다.
	위험도	[X] 임의로 작성된 SQL 쿼리 입력에 대한 검증이 미흡하여 사용자의 정보(쿠키, 세션 등)를 탈취하거나 자동으로 비정상적인 기능이 실행, 조작의 가능한 경우 [Δ] 임의로 작성된 SQL 쿼리 입력에 대한 검증이 미흡한 경우 [O] 임의로 작성된 SQL 쿼리 입력에 대한 검증이 안전하게 이루어진 경우
크로스사이트 스크립팅 (XSS)	**설명**	웹 사이트에 악성 스크립트를 삽입하는 공격. 이를 통해 웹 페이지를 방문하는 사용자의 브라우저에서 사용자의 정보를 탈취하거나 사용자를 다른 웹 사이트로 리다이렉트하는 등의 행동을 할 수 있습니다.
	원인	1. 크로스사이트 스크립팅(XSS) 공격은 주로 웹 애플리케이션에서 사용자의 입력 값을 적절하게 검증하거나 필터링하지 않았을 때 발생합니다. 이 경우, 공격자는 악성 스크립트를 웹 페이지에 삽입할 수 있습니다. 2. 웹 애플리케이션에서 사용자의 입력을 그대로 웹 페이지에 반영하거나, HTML 태그나 자바스크립트 코드를 필터링하지 않을 때 XSS 공격에 취약해집니다. 3.사용자가 웹 브라우저를 통해 직접 접근할 수 있는 URL, HTTP 헤더, 쿠키 등에도 XSS 공격이 발생할 수 있습니다.
	위험도	[X] 사용자 입력 값에 대한 검증 및 필터링이 미흡하여 데이터 유출 및 변조 외에도 서버 파일을 쓰거나 읽을 수 있으며 직접 임의의 명령 실행이 가능한 경우 [Δ] 사용자 입력 값에 대한 검증 및 필터링이 미흡한 경우 [O] 사용자 입력 값에 대한 검증 및 필터링이 안전하게 이루어진 경우

디렉토리 트레버설 (Directory Traversal)	**설명**	공격자가 웹 애플리케이션에서 의도치 않게 파일 시스템에 접근하거나, 허용되지 않은 파일을 읽거나 쓰는 것을 목표로 하는 공격. 이를 통해 공격자는 서버의 중요한 정보를 열람하거나, 시스템을 조작할 수 있습니다.
	원인	1. 디렉토리 트레버설(Directory Traversal) 공격은 주로 웹 애플리케이션에서 사용자의 입력값을 파일의 경로나 이름으로 사용할 때 발생합니다. 이 경우, 공격자는 "../"와 같은 상대 경로를 사용하여 웹 애플리케이션의 루트 디렉토리를 벗어나 시스템의 임의의 파일에 접근할 수 있습니다. 2. 웹 애플리케이션에서 사용자의 입력 값에 대한 적절한 검증 또는 필터링이 이루어지지 않았을 때, 공격자는 이러한 취약점을 이용해 서버의 중요한 파일을 읽거나 변조할 수 있습니다. 3. 사용자가 웹 브라우저를 통해 직접 접근할 수 있는 URL이나 HTTP 헤더에도 이러한 공격이 발생할 수 있습니다.
	위험도	[X] 사용자 입력 값에 대한 검증 및 필터링이 미흡하여 시스템의 중요한 파일에 접근이 가능하며, 서버의 파일을 읽거나 쓰는 등의 공격이 가능한 경우 [Δ] 사용자 입력 값에 대한 검증 및 필터링이 미흡하여 특정 파일에 접근이 가능한 경우 [O] 사용자 입력 값에 대한 검증 및 필터링이 안전하게 이루어진 경우
파일 업로드 (File Upload)	**설명**	공격자가 악성 코드가 포함된 파일을 업로드하거나, 서버의 취약점을 이용하여 허가되지 않은 위치에 파일을 업로드하는 공격. 이로 인해 서버가 공격당하거나, 사용자의 정보가 유출될 수 있습니다.
	원인	1.파일 업로드(File Upload) 공격은 주로 웹 애플리케이션에서 파일의 종류, 크기, 업로드 위치 등을 적절하게 제한하지 않을 때 발생합니다. 이 경우, 공격자는 악성 파일을 업로드하거나, 업로드된 파일을 이용해 서버를 조작하거나, 다른 사용자의 시스템을 공격할 수 있습니다. 2. 웹 애플리케이션에서 업로드된 파일의 실행을 허용할 경우, 공격자는 이를 이용해 웹 서버 내에서 악성 코드를 실행시켜 서버의 제어권을 공격자에게 넘어갈 수 있습니다. 3.사용자가 업로드하는 파일에 대한 적절한 검증 또는 필터링이 이루어지지 않았을 때, 공격자는 이러한 취약점을 이용해 서버의 중요한 파일을 변조하거나 시스템을 공격할 수 있습니다.
	위험도	[X] 업로드 파일에 대한 확장자 검증이 미흡하여 공격자에게 서버 노출 및 제어권 제공이 된 경우 [Δ] 업로드 파일에 대한 확장자 검증이 미흡한 경우 [O] 업로드 파일에 대한 확장자 검증이 안전하게 이루어진 경우

파일 다운로드 (File Download)	**설명**	공격자가 서버의 중요한 파일을 다운로드하거나, 사용자에게 악성 파일을 다운로드하게 하는 공격. 이로 인해 서버의 중요 정보가 유출되거나, 사용자의 컴퓨터가 공격당할 수 있습니다.
	원인	1. 파일 다운로드(File Download) 공격은 주로 웹 애플리케이션에서 다운로드할 파일의 선택이나 접근 권한 등을 적절하게 제한하지 않을 때 발생합니다. 이 경우, 공격자는 서버에 저장된 중요한 파일을 다운로드할 수 있습니다. 2. 웹 애플리케이션에서 사용자의 입력 값을 파일의 경로나 이름으로 사용할 때도 파일 다운로드 취약점이 발생할 수 있습니다. 공격자는 이를 이용해 서버의 임의의 파일에 접근하거나, 해당 파일을 다운로드할 수 있습니다. 3. 사용자가 다운로드하는 파일에 대한 적절한 검증 또는 필터링이 이루어지지 않았을 때, 공격자는 이러한 취약점을 이용해 서버의 중요한 파일을 변조하거나 시스템을 공격할 수 있습니다.
	위험도	[X] 다운로드 파일이 저장된 디렉터리 이외에 접근 가능한 경우 주요 서비스 및 서버 정보 유출 가능성이 존재할 경우 [△] 다운로드 파일이 저장된 디렉터리 이외에 접근이 가능한 경우 [O] 다운로드 파일이 저장된 디렉터리 접근이 불가능한 경우
서버사이트 리퀘스트 변조 (SSRF)	**설명**	공격자가 서버를 이용하여 내부 네트워크에 접근하거나 다른 시스템에 요청을 보내는 공격. SSRF를 통해 공격자는 외부에서 접근할 수 없는 시스템에 요청을 보내 정보를 획득하거나, 서버를 악용하여 다른 시스템을 공격할 수 있습니다.
	원인	1. 서버사이드 리퀘스트 변조(SSRF) 공격은 주로 웹 애플리케이션에서 서버가 외부 시스템에 요청을 보내는 기능을 적절하게 제한하거나 검증하지 않을 때 발생합니다. 이 경우, 공격자는 서버를 이용하여 내부 네트워크에 접근하거나 다른 시스템에 요청을 보낼 수 있습니다. 2. 웹 애플리케이션에서 사용자의 입력 값을 외부 시스템에 대한 요청으로 사용할 때도 SSRF 공격이 발생할 수 있습니다. 공격자는 이를 이용해 외부에서 접근할 수 없는 시스템에 요청을 보내 정보를 획득하거나, 서버를 악용하여 다른 시스템을 공격할 수 있습니다. 3. 사용자가 요청하는 URL이나 파라미터에 대한 적절한 검증 또는 필터링이 이루어지지 않았을 때, 공격자는 이러한 취약점을 이용해 서버의 중요한 정보를 획득하거나 시스템을 공격할 수 있습니다.
	위험도	[X] 서버측에서 이루어진 요청 값이 변조되는 가능성 [O] 서버측에서 이루어진 요청 값이 정상 동작하는 경우

크로스사이트 리퀘스트 변조 (CSRF)	**설명**	공격자가 사용자의 권한을 악용하여 서버에 요청을 보내는 공격. 이를 통해 공격자는 사용자가 의도하지 않은 행동을 수행하게 할 수 있습니다.
	원인	1. 크로스사이트 리퀘스트 변조(CSRF) 공격은 주로 웹 애플리케이션에서 사용자의 요청을 적절하게 검증하거나 사용자의 세션을 관리하지 않을 때 발생합니다. 이 경우, 공격자는 사용자의 권한을 악용하여 서버에 요청을 보낼 수 있습니다. 2. 웹 애플리케이션에서 사용자의 입력 값을 그대로 요청에 반영하거나, 쿠키를 이용한 인증 정보를 자동으로 요청에 포함할 때 CSRF 공격이 발생할 수 있습니다. 공격자는 이를 이용해 사용자가 의도하지 않은 행동을 수행하게 할 수 있습니다. 3.사용자가 방문하는 웹 페이지에 악성 스크립트가 삽입되어 있거나, 사용자의 클릭 등의 행동에 따라 자동으로 악성 요청이 발생하는 경우에도 CSRF 공격이 가능합니다. 이러한 경우 사용자는 자신의 의지와는 무관하게 공격자에게 원하지 않는 요청을 보내게 됩니다.
	위험도	[X] 사용자 입력 값에 대한 필터링이 이루어지지 않고, 권한 탈취가 가능한 경우 [Δ] 사용자 입력 값에 대한 필터링이 이루어지지 않으며, HTML 코드(또는 스크립트)를 입력하여 실행되는 경우 [O] 사용자 입력 값에 대한 검증 및 필터링이 안전하게 이루어진 경우
커맨드 인젝션 (Command Injection)	**설명**	공격자가 악의적인 명령어를 시스템에 주입하는 공격. 웹 애플리케이션에서 사용자의 입력 값을 시스템 명령어의 일부로 사용할 경우 발생하며, 공격자는 이를 이용해 서버를 제어하거나 중요한 데이터를 유출시킬 수 있습니다.
	원인	1.커맨드 인젝션(Command Injection) 공격은 주로 웹 애플리케이션에서 사용자의 입력값을 시스템 명령어의 일부로 사용할 때 발생합니다. 이 경우, 공격자는 악의적인 명령어를 시스템에 주입할 수 있습니다. 2. 또한, 웹 애플리케이션에서 사용자의 입력값에 대한 적절한 검증 또는 필터링이 이루어지지 않았을 때, 공격자는 이러한 취약점을 이용해 서버를 제어하거나 중요한 데이터를 유출시킬 수 있습니다. 3. 사용자의 입력 값을 그대로 시스템 명령어로 사용하거나, 명령어 실행 결과를 사용자에게 제공하는 경우에도 커맨드 인젝션 공격이 가능합니다. 이러한 경우 사용자는 자신의 의지와는 무관하게 공격자에게 원하지 않는 명령을 실행시키게 됩니다.
	위험도	[X] 임의로 작성된 명령어 입력에 대한 검증이 미흡하여 시스템 명령어를 실행하거나 비정상적인 기능이 자동으로 실행, 조작 가능한 경우 [Δ] 임의로 작성된 명령어 입력에 대한 검증이 미흡한 경우 [O] 임의로 작성된 명령어 입력에 대한 검증이 안전하게 이루어진 경우

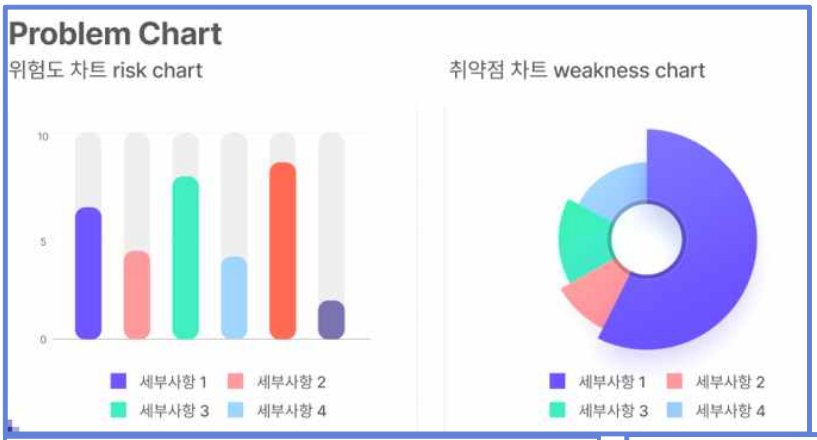
약한 문자열 강도 (Weak String Strength)	**설명**	비밀번호나 암호화 키 등의 보안을 위한 문자열이 쉽게 예측 가능하거나 복잡도가 낮아 공격자에 의해 쉽게 뚫릴 수 있는 상태. 이로 인해 공격자는 브루트포스 공격 등을 이용해 비밀번호를 쉽게 획득하거나, 암호화 통신을 해독할 수 있습니다.
	원인	1. 약한 문자열 강도(Weak String Strength)는 주로 사용자가 간단하거나 예측 가능한 비밀번호를 사용할 때 발생합니다. 이 경우, 공격자는 브루트포스 공격 등을 이용해 비밀번호를 쉽게 획득할 수 있습니다. 2. 웹 애플리케이션에서 비밀번호의 복잡도나 길이를 적절하게 제한하지 않았을 때, 사용자는 간단한 비밀번호를 설정하게 되어 약한 문자열 강도 문제가 발생할 수 있습니다. 3. 암호화 키의 생성 규칙이 공격자에게 알려졌거나, 키의 복잡도가 충분히 높지 않은 경우에도 약한 문자열 강도 문제가 발생합니다. 이 경우, 공격자는 암호화 키를 예측하거나 브루트포스 공격을 통해 키를 획득하고, 암호화 통신을 해독할 수 있습니다.
	위험도	[X] 계정 및 패스워드가 유추하기 쉬운 값으로 설정되어 있으며, 일정 횟수 이상 인증 실패 시 로그인을 제한하고 있지 않아 공격자가 사용자 계정의 자격 증명을 탈취하고, 민감한 정보에 접근하거나 악의적인 활동이 가능한 경우 [Δ] 계정 및 패스워드가 유추하기 쉬운 값으로 설정되어 있으며, 일정 횟수 이상 인증 실패 시 로그인을 제한하고 있지 않은 경우 [O] 계정 및 패스워드가 유추하기 어려운 값으로 설정되어 있으며, 일정 횟수 이상 인증 실패 시 로그인을 제한하고 있는 경우
불충분한 인증 (Insufficient Authorization)	**설명**	시스템이 사용자의 권한을 제대로 확인하지 않아 발생하는 보안 취약점. 이로 인해 공격자는 권한 없이 시스템의 중요한 기능을 사용하거나, 타인의 개인정보를 열람하고 수정할 수 있습니다.
	원인	1. 불충분한 인증(Insufficient Authorization)은 주로 웹 애플리케이션에서 사용자의 권한을 제대로 확인하지 않을 때 발생합니다. 이 경우, 공격자는 권한 없이 시스템의 중요한 기능을 사용할 수 있습니다. 2. 웹 애플리케이션에서 각 요청에 대한 권한 검사를 누락하거나, 권한 검사가 부정확하게 이루어졌을 때에도 불충분한 인증 문제가 발생할 수 있습니다. 이럴 경우, 공격자는 자신의 권한 범위를 벗어나는 행동을 할 수 있습니다. 3. 사용자의 세션을 제대로 관리하지 않거나, 세션에 대한 검증이 충분하지 않은 경우에도 불충분한 인증 문제가 발생합니다. 공격자는 이를 이용해 타인의 세션을 가로채거나, 세션을 통해 권한을 부여받을 수 있습니다.
	위험도	[X] 중요 정보 페이지 접근에 대한 추가 인증을 하지 않는 경우, 권한이 없는 사용자가 중요 정보 페이지에 접근하여 정보를 유출하거나 변조할 가능성 [Δ] 중요 정보 페이지 접근에 대한 추가 인증을 하지 않는 경우 [O] 중요 정보 페이지 접근 시 추가 인증을 하는 경우

불충분한 세션 만료 (Insufficient session expiration)	**설명**	웹 애플리케이션에서 사용자의 세션을 제때 종료하지 않아 발생하는 보안 취약점. 이로 인해 공격자는 사용자의 세션을 탈취하거나, 사용자가 로그아웃한 후에도 계속해서 사용자의 권한으로 서버에 요청을 보낼 수 있습니다.
	원인	1. 불충분한 세션 만료(Insufficient session expiration)는 웹 애플리케이션에서 사용자의 세션을 적절하게 관리하지 않거나, 세션 만료 시간이 너무 길게 설정되어 있을 때 발생합니다. 이 경우, 공격자는 사용자의 세션을 탈취하거나, 사용자가 로그아웃한 후에도 계속해서 사용자의 권한으로 서버에 요청을 보낼 수 있습니다. 2. 웹 애플리케이션에서 사용자의 활동을 통해 세션의 만료 시간을 연장하지 않거나, 사용자의 로그아웃 요청에 따라 세션을 즉시 종료하지 않는 경우에도 불충분한 세션 만료 문제가 발생할 수 있습니다.
	위험도	[X] 세션 종료 시간이 설정되어 있지 않아 세션 재사용이 가능하여 각종 정보 탈취 및 변조가 가능한 경우 [Δ] 세션 종료 시간이 설정되어 있지 않거나 재사용 가능성이 존재할 경우 [O] 세션 종료 시간 및 세션 재사용이 불가능하게 설정되어 있는 경우
세션고정 (Session Fixation)	**설명**	공격자가 특정 세션 ID를 사용자에게 강제로 사용하게 하여, 사용자가 로그인한 후에도 공격자가 해당 세션 ID를 이용해 사용자의 권한으로 서버에 요청을 보내는 공격. 이로 인해 공격자는 사용자의 개인정보를 열람하거나, 사용자의 권한으로 서버의 기능을 사용할 수 있습니다.
	원인	1. 세션 고정(Session Fixation) 공격은 웹 애플리케이션에서 사용자가 로그인할 때마다 새로운 세션 ID를 부여하지 않을 경우 발생합니다. 이 경우, 공격자는 특정 세션 ID를 사용자에게 강제로 사용하게 하여, 사용자가 로그인한 후에도 해당 세션 ID를 이용해 사용자의 권한으로 서버에 요청을 보낼 수 있습니다. 2. 웹 애플리케이션에서 세션 ID의 생성 규칙이 예측 가능하거나, 세션 ID를 안전하게 전송하지 않는 경우에도 세션 고정 공격이 가능합니다. 이 경우, 공격자는 예측 가능한 세션 ID를 이용해 공격을 수행하거나, 사용자의 세션 ID를 탈취할 수 있습니다.
	위험도	[X] 로그인 세션 ID가 고정되어 사용되거나 새로운 세션 ID가 발행되지만 예측 가능한 패턴으로 발행될 경우 [O] 로그인할 때마다 예측 불가능한 새로운 세션 ID가 발행되고, 기존 세션 ID는 파기될 경우

관리자 페이지 노출 (Administrator Page Exposure)	**설명**	관리자 인터페이스가 공격자에게 노출되어 있는 상태. 이러한 상태에서 공격자는 관리자 페이지를 통해 시스템을 조작하거나 중요한 정보를 획득할 수 있음. 또한, 관리자 페이지는 일반 사용자에게는 필요 없는 고급 기능을 제공하기 때문에, 이를 악용하면 시스템에 심각한 피해를 줄 수 있습니다.
	원인	1. 관리자 페이지 노출(Administator Page Exposure)은 주로 웹 애플리케이션에서 관리자 인터페이스에 대한 접근 제어가 충분히 이루어지지 않았을 때 발생합니다. 이 경우, 공격자는 관리자 인터페이스를 통해 시스템을 조작하거나 중요한 정보를 획득할 수 있습니다. 2. 웹 애플리케이션에서 관리자 인터페이스의 위치나 접근 방법이 예측 가능하거나, 관리자 인터페이스에 대한 정보가 공개되어 있을 때에도 관리자 페이지 노출 문제가 발생할 수 있습니다. 이 경우, 공격자는 이러한 정보를 이용해 관리자 인터페이스에 접근할 수 있습니다. 3. 사용자 인증 절차가 취약하거나, 인증 절차가 없는 경우에도 관리자 페이지 노출 문제가 발생합니다. 이 경우, 공격자는 인증 절차를 우회하거나 인증을 필요로 하지 않는 관리자 인터페이스에 접근할 수 있습니다.
	위험도	[X] 유추하기 쉬운 URL로 관리자 페이지 접근이 가능하여 웹 관리자의 권한이 노출된 경우 [O] 유추하기 어려운 URL로 관리자 페이지 접근이 거의 불가능한 경우
데이터 평문 전송 (Plain Text Transmission)	**설명**	암호화되지 않은 상태로 데이터를 전송하는 것을 의미. 이는 네트워크를 통해 전송되는 데이터가 도청되거나 변조될 위험이 있음. 특히, 개인정보나 비밀번호 등의 민감한 정보가 평문으로 전송되면, 이를 도청하는 공격자에게 쉽게 정보가 노출될 수 있습니다.
	원인	1. 데이터 평문 전송(Plain Text Transmission)은 주로 웹 애플리케이션에서 통신을 암호화하지 않거나, 암호화가 적절하게 이루어지지 않았을 때 발생합니다. 이 경우, 네트워크를 통해 전송되는 데이터가 도청되거나 변조될 위험이 있습니다. 2. 웹 애플리케이션에서 사용자의 민감한 정보를 평문으로 전송하거나, 사용자의 요청에 대한 응답을 평문으로 전송하는 경우에도 데이터 평문 전송 문제가 발생할 수 있습니다. 이 경우, 공격자는 도청을 통해 사용자의 민감한 정보를 쉽게 획득할 수 있습니다. 3. HTTPS와 같은 보안 프로토콜을 사용하지 않거나, 보안 프로토콜의 설정이 적절하지 않은 경우에도 데이터 평문 전송 문제가 발생합니다. 이 경우, 공격자는 네트워크 트래픽을 도청하거나 변조하여 사용자의 정보를 획득하거나, 사용자의 요청이 변조될 수 있습니다.
	위험도	[X] 중요 정보 전송 구간에 암호화 통신이 이루어지지 않아 간단한 도청만으로도 공격자가 민감한 정보를 탈취 및 도용할 수 있는 경우 [Δ] 중요 정보 전송 구간에 일부만 암호화 통신이 적용되어 일부 민감한 정보가 여전히 평문으로 전송되는 경우 [O] 모든 중요 정보 전송 구간에 안전한 암호화 통신이 적용된 경우

3. YourCode-X 성능 지표

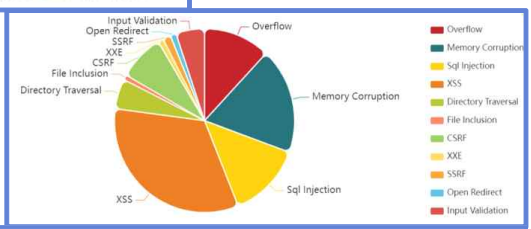
3-1. 성능 지표 결과



Problem Chart (위험도 차트)

▫ 취약점이 발견된 파일 경로 개수와 비례하여 결과가 Bar Chart에 반영

▫ 위험, 주의, 양호 기준으로 분석 결과 반영



Weakness Chart (취약점 차트)

▫ 취약점이 발견된 케이스들의 개수를 측정하여 결과가 Pie Chart에 반영

▫ 2013년부터~ 취약점 현황 데이터가 반영되는 cvedetails.com에서 연도별 결과와 함께 비교 가능

Problem List (취약점 세부 목록)

▫ 취약점 항목별 내부 점검 결과가 테이블 형태로 반영됨

▫ 양호를 제외한(위험, 주의) RISK 열은 진단 세부사항으로 결과를 받을 수 있음

[단계별 세분화 진행 예정(1~5단계)]

▫ 동적 분석 결과에 따른 결과가 정적 분석에서 신뢰성과 정확성을 높여 도움을 받을 수 있게 해줌

| 점검 범위

※ 점검 항목에 제시되어 있는 부분으로 검사 진행되며 필요시 추가 예정

<웹 취약점 점검 항목>	<데이터베이스>	<데이터(파라미터) 전송 방식>
- SQL 인젝션 - XSS - Directory Indexing - File Upload - File Download - …	- RDBMS	- GET - POST

3-2. 결과 도출 기준

| 세부 기준

<웹 취약점 점검 세부 기준>

[O: 구현이 체계화 됨, △: 구현이 미흡, X: 구현이 명확히 적용되지 않음]

1. SQL 인젝션(SQL Injection)

No.	참고	내용	평가
1.	해당 함수별 전송 방식	GET/POST 방식 중 어떤 파라미터로 데이터가 전송되는지 파악	
2.	func classicSQLI	기본적인 SQL Injection 공격 기법을 사용하여 해당 Exploit Code가 동작될 경우 심도 있는 공격이 이루어짐	
3.	func errorBasedSQLI	의도적으로 잘못된 SQL 쿼리문을 DB에 요청하여 서버로부터 리턴받은 에러 메시지를 통해 DB의 정보를 파악하는 공격으로 이루어짐	
4.	func unionBasedSQLI	여러개의 SQL문을 한번에 실행하는 공격 기법을 삽입하여 공격자가 원하는 정보를 유출시키는 공격으로 이루어짐	
5.	func blindSQLI	외부 입력 쿼리에 참, 거짓을 서버로부터 리턴받고 DB의 내용을 추측하는 공격으로 이루어짐	
6.	func outOfBandSQLI	쿼리의 결과를 다른 외부 채널을 통해 전달하여 실행되는 서버에서 캡처 패킷 확인이 가능하도록 이루어짐	
7.	func secondOrderSQLI	사용자가 제공한 데이터가 애플리케이션에 의해 저장된 후로 안전하지 않은 방식의 SQL 쿼리가 통합되는 경우로 이루어짐	

2. 크로스사이트스크립트(XSS)

No.	참고	내용	평가
1.	해당 함수별 전송 방식	GET/POST 방식 중 어떤 파라미터로 데이터가 전송되는지 파악	
2.	func storedXSS	웹 애플리케이션 취약점이 있는 웹 서버에 악성 스크립트를 영속적으로 저장해 놓는 공격으로 이루어짐	
3.	func reflectedXSS	웹 애플리케이션의 지정된 파라미터를 사용할 때 발생하는 취약점을 이용하는 공격으로 이루어짐	
4.	func domBasedXSS	브라우저에서 렌더링할 때 스크립트가 실행되어 DOM 문서 내 계층적으로 구성된 객체에 접근하여 읽고 쓰게 되면서 웹 페이지의 컨텐츠가 변경되는 공격으로 이루어짐	

3. 디렉토리 트레버설(Directory Traversal)

No.	참고	내용	평가
1.	해당 함수별 전송 방식	GET/POST 방식 중 어떤 파라미터로 데이터가 전송되는지 파악	
2.	func basicDT	기본적인 디렉토리 트레버설 공격을 사용하여 상위 디렉토리로 이동하고 시스템 파일에 접근할 수 있는 공격으로 이루어짐	
3.	func nullByteDT	NULL 바이트를 이용하여 파일 경로를 조작하고, 확장자 검증을 우회하여 시스템 파일에 접근할 수 있는 공격으로 이루어짐	
4.	func encodingDT	URL 인코딩을 활용하여 외부 입력을 조작하고, 디렉토리 경로를 우회적으로 탐색하여 시스템 파일에 접근할 수 있는 공격으로 이루어짐	
5.	func doubleEncodingDT	이중 URL 인코딩을 활용하여 외부 입력을 조작하고, 디렉토리 경로를 우회적으로 탐색하여 시스템 파일에 접근할 수 있는 공격으로 이루어짐	
6.	func unicodeEncodingDT	유니코드 인코딩을 이용하여 파일 경로를 조작하고, 외부 입력에 대한 인코딩을 우회하여 시스템 파일에 접근할 수 있는 공격으로 이루어짐	

4. 파일 업로드(File Upload)

No.	참고	내용	평가
1.	해당 함수별 전송 방식	GET/POST 방식 중 어떤 파라미터로 데이터가 전송되는지 파악	
2.	func extensionBypassFU	확장자를 변조하여 웹 애플리케이션의 보안 메커니즘을 우회하고 악성 코드를 실행시킬 수 있는 공격으로 이루어짐	
3.	func capacityFU	대량의 파일을 업로드하여 파일 용량 제한의 유무를 판별하고 없다면 웹 애플리케이션 리소스를 고갈시키는 서비스 거부 공격으로 이루어짐	

5. 파일 다운로드(File Download)

No.	참고	내용	평가
1.	해당 함수별 전송 방식	GET/POST 방식 중 어떤 파라미터로 데이터가 전송되는지 파악	
2.	func pathManipulationFD	다운로드 경로를 조작하여 서버에 저장된 임의의 파일을 다운로드하는 공격으로 이루어짐	