SonarQube 무료와 유료 에디션 룰 차이점

이 문서는 SonarQube Community Edition과 Developer Edition 간의 Rule 차이에 대한 정보를 공유하기 위해 작성되었습니다.

유료 버전 룰 차이

다음은 SonarQube ver10 기준의 룰 차이 입니다. 룰은 지속적으로 업데이트 되고 있습니다. 다음은 2023년 7월 기준입니다.

- Java의 경우, 22개의 보안 취약성과 관련된 룰 추가
- Javascript의 경우, 14개의 보안 취약성과 관련된 룰 추가
- C#은 16개의 보안취약성 룰 추가
- Python은 16개의 보안취약성 룰이 추가

	Java	Bug	Vulnerability	Code Smell	Secutiry Hotspot	합계
1	Community - Java	149	33	403	37	622
2	Developer - Java	154	55	403	38	650
3	**Javascript**	**Bug**	**Vulnerability**	**Code Smell**	**Secutiry Hotspot**	**합계**
4	Community - Javascript	74	16	163	61	314
5	Developer - Javasciprt	74	30	163	62	329
6	**C#**	**Bug**	**Vulnerability**	**Code Smell**	**Secutiry Hotspot**	**합계**
7	Community - C#	78	18	288	28	412
8	Developer - C#	78	34	288	29	429
9	**Python**	**Bug**	**Vulnerability**	**Code Smell**	**Secutiry Hotspot**	**합계**
10	Community - Python	59	16	106	43	224
11	Developer - Python	66	32	107	44	249

유료 버전 보안 취약성 룰 요약

유료 버전에는 커뮤니티 버전에는 없는 보안 취약점과 관련된 룰들이 존재합니다.

Developer and Enterprise 버전에 있는 룰들은 다음과 같이 CWE, OWASP, SANS-Top에서 알려져 있는 보안 취약성과 관련하여 취약한 코드를 검출합니다.

- 세션 쿠키 삽입 공격에 취약한 코드 검출
- 데이터베이스 쿼리에 악성 코드 삽입 공격에 취약한 코드 검출
- 공격자가 역직렬화된 객체에 악성 코드를 삽입할 수 있는 유형의 공격에 취약한 코드 검출
- 공격자가 악성 코드를 삽입하여 코드를 실행할 수 있는 취약점을 검출
- 공격자가 신뢰할 수 없는 입력을 사용하여 사용자의 브라우저에 악성 코드를 삽입할 수 있는 코드 검출
- HTTP 요청 재지정 위조 공격에 취약한 코드 검출
- HTTP 응답 헤더 주입 공격에 취약한 코드 검출
- I/O 함수 호출 경로 주입 공격에 취약한 코드 검출
- 로깅 주입 공격에 취약한 코드 검출

보안 취약성 룰 설명

세션 쿠키 삽입 공격에 취약한 코드 검출

- 세션 쿠키 삽입 공격은 공격자가 신뢰할 수 없는 입력으로부터 세션 쿠키를 생성할 때 발생할 수 있는 공격입니다. 신뢰할 수 없는 입력에는 사용자 이름, 암호 및 기타 개인 정보와 같은 사용자의 데이터가 포함될 수 있습니다. 이 데이터가 세션 쿠키를 생성하는 데 사용되면 공격자가 사용자의 세션을 가로채고 계정에 액세스하는 데 사용할 수 있습니다.

데이터베이스 쿼리에 악성 코드 삽입 공격에 취약한 코드 검출

- 데이터베이스 쿼리는 삽입 공격에 취약하지 않아야 합니다. 삽입 공격은 공격자가 데이터베이스 쿼리에 악성 코드를 삽입할 수 있는 유형의 공격입니다. 이를 통해 공격자는 무단으로 데이터에 액세스하고 데이터를 수정하거나 데이터베이스를 제어할 수 있습니다.

공격자가 역직렬화된 객체에 악성 코드를 삽입할 수 있는 유형의 공격에 취약한 코드 검출

- 역직렬화는 공격자가 역직렬화된 객체에 악성 코드를 삽입할 수 있는 유형의 공격입니다. 이를 통해 공격자는 대상 시스템에서 임의의 코드를 실행할 수 있습니다.

공격자가 악성 코드를 삽입하여 코드를 실행할 수 있는 취약점을 검출

- 동적 코드 실행은 공격자가 악성 코드를 삽입하여 코드를 실행할 수 있는 취약점이 없어야 합니다. 동적 코드 실행 취약점은 공격자가 신뢰할 수 없는 입력을 사용하여 애플리케이션이 실행할 코드를 조작할 수 있는 경우 발생합니다. 이로 인해 공격자는 애플리케이션을 손상시키거나 제어할 수 있습니다.

공격자가 신뢰할 수 없는 입력을 사용하여 사용자의 브라우저에 악성 코드를 삽입할 수 있는 코드 검출

- 엔드포인트는 반사 XSS 공격에 취약하지 않아야 합니다. 반사 XSS 공격은 공격자가 신뢰할 수 없는 입력을 사용하여 사용자의 브라우저에 악성 코드를 삽입하는 것을 의미합니다. 이로 인해 공격자는 사용자를 사기성 웹사이트로 유도하거나 사용자의 세션을 가로채거나 기타 악의적인 작업을 수행할 수 있습니다.

HTTP 요청 재지정 위조 공격에 취약한 코드 검출

- HTTP 요청 재지정은 위조 공격에 취약하지 않아야 합니다. 위조 공격은 공격자가 신뢰할 수 없는 입력을 사용하여 사용자를 악성 웹사이트로 유도하는 것을 의미합니다. 이로 인해 공격자는 사용자의 개인 정보를 훔치거나 악성 코드를 실행할 수 있습니다.

HTTP 응답 헤더 주입 공격에 취약한 코드 검출

- HTTP 응답 헤더는 주입 공격에 취약하지 않아야 합니다. 주입 공격은 공격자가 신뢰할 수 없는 입력을 사용하여 HTTP 응답 헤더를 조작하여 악의적인 콘텐츠를 삽입하는 것을 의미합니다. 이로 인해 공격자는 사용자를 사기성 웹사이트로 유도하거나 사용자의 세션을 가로채거나 기타 악의적인 작업을 수행할 수 있습니다.

I/O 함수 호출 경로 주입 공격에 취약한 코드 검출

- I/O 함수 호출은 경로 주입 공격에 취약하지 않아야 합니다. 경로 주입 공격은 공격자가 신뢰할 수 없는 입력을 사용하여 I/O 함수의 경로 매개변수를 조작하여 악의적인 파일을 열거나 실행하는 것을 의미합니다. 이로 인해 공격자는 시스템을 손상시키거나 악성 코드를 실행할 수 있습니다.

로깅 주입 공격에 취약한 코드 검출

- 로깅은 주입 공격에 취약하지 않아야 합니다. 주입 공격은 공격자가 신뢰할 수 없는 입력을 사용하여 로깅 메시지를 조작하여 악의적인 콘텐츠를 삽입하는 것을 의미합니다. 이로 인해 공격자는 보안 취약점을 악용하거나 민감한 정보를 훔칠 수 있습니다.