

**INFORME SONARQUBE**

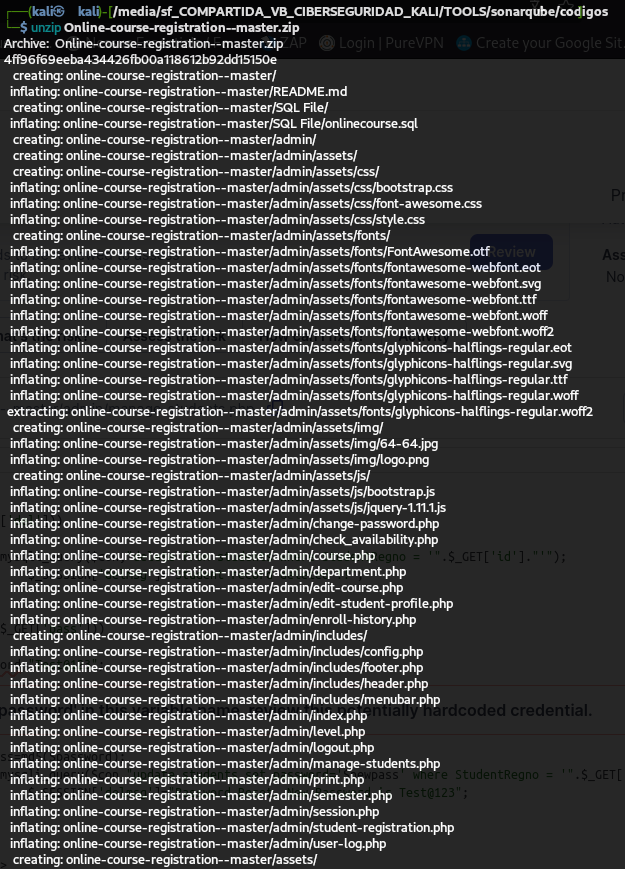
**Online-course-registration--master**

1.- INTRODUCCIÓN

SonarQube es una plataforma de código abierto, compatible con múltiples lenguajes de programación, usada para la revisión automática de errores y vulnerabilidades en el código fuente de las aplicaciones, integrándose con diversas herramientas, entre la que se encuentra SonarScanner, la cual, se utiliza para escanear el código fuente, enviando los resultados a un servidor de SonarQube para su análisis y visualización.

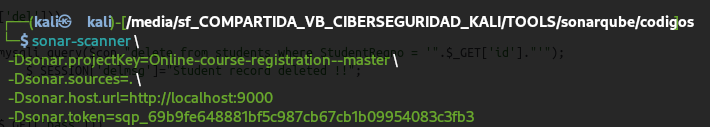
2.- PROCESO DE ESCANEO DEL CODIGO

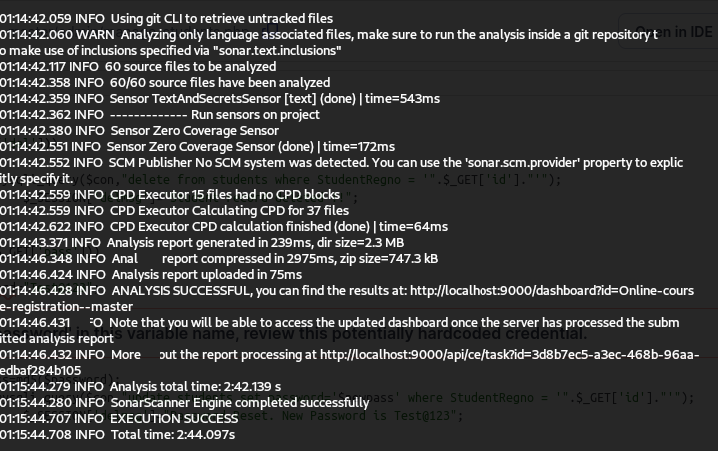
En primer lugar, vamos a proceder a la descomprimir el archivo facilitado “Onlinecourse-registration—master”:



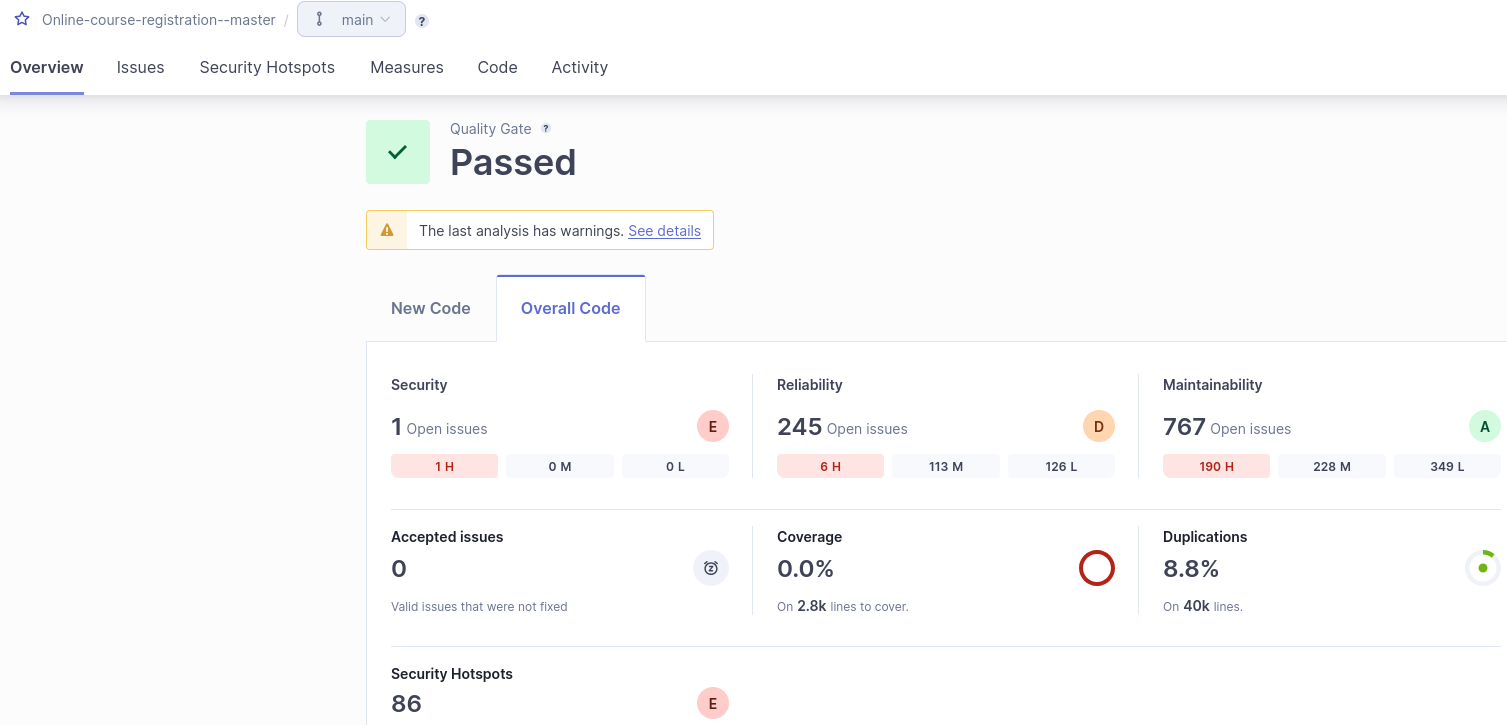
Esto es una muestra de la información, siendo el código PHP.

Después, se procede al análisis mediante la herramienta sonnar-scanner, dando como resultado un informe que envía al servidor de la aplicación SonarQube:

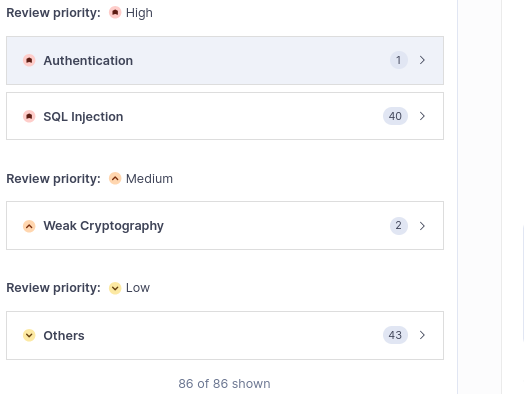
****

****

3.- PROCESO DE ANALISIS DEL RESULTADO

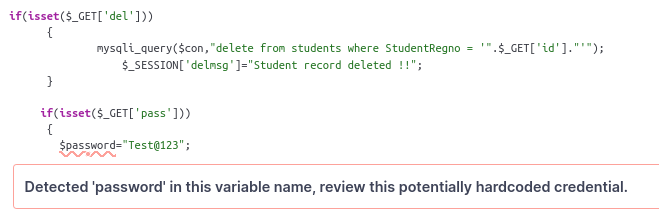


Dentro del campo de la ciberseguridad, concretamente en la pestaña de “Security Hotspots” podemos observar que el código tiene presuntas vulnerabilidades altas, medias y bajas, que deben ser verificadas:

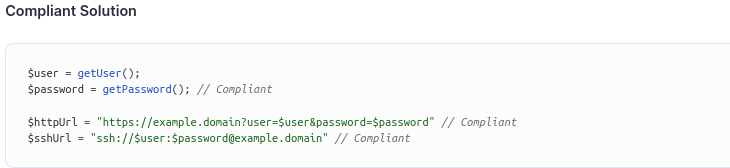


3.1.- RIESGO ALTO

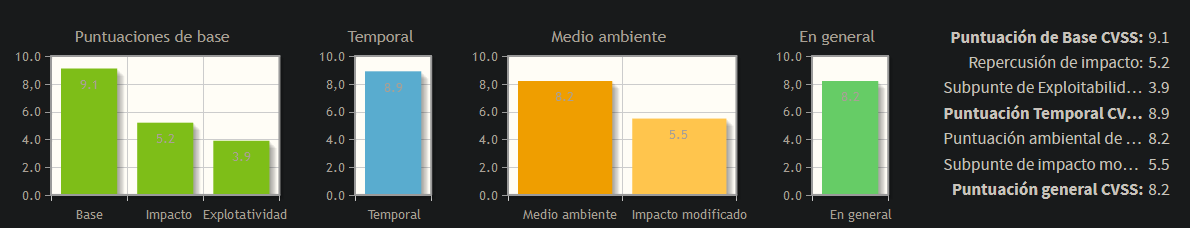
- 3.1.1.- La vulnerabilidad de Autenticación detectada de prioridad alta, se debe a que dentro del código aparece una contraseña legible:



Esta vulnerabilidad podría permitir a atacantes malintencionados el acceso a datos sensibles y confidenciales (BBDD, APIs, servicios, etc), recomendándose su verificación y corrección, en su caso, por otras prácticas de codificación más seguras, como, el almacenamiento las credenciales en un archivo de configuración fuera del repositorio y usar los servicios iCloud para almacenar datos sensibles, encontrándose en el OWASP Top 10 2021, en la categoría número 7 llamada “Identificación y autenticación”.



Esta vulnerabilidad podría tener una puntuación CVSS alta, adjuntando grafico del resultado:

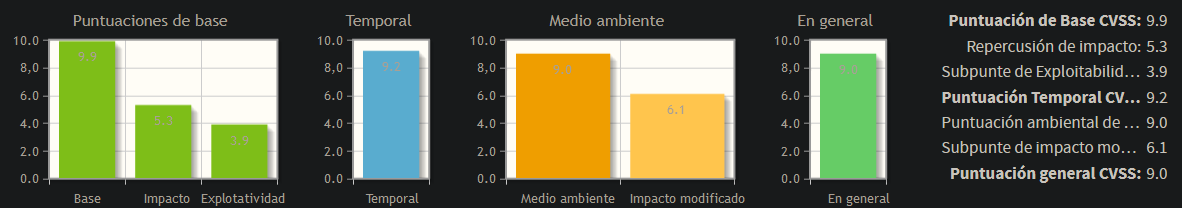


-3.1.2.- Se han detectado 40 posibles fallos en la codificación que podrían permitir a los atacantes realizar inyecciones SQL maliciosas al concatenar valores que no son de confianza en la misma consulta, recomendando su verificación y corrección, en este caso, usando consultas parametrizadas y/o declaraciones preparadas, como en este caso:



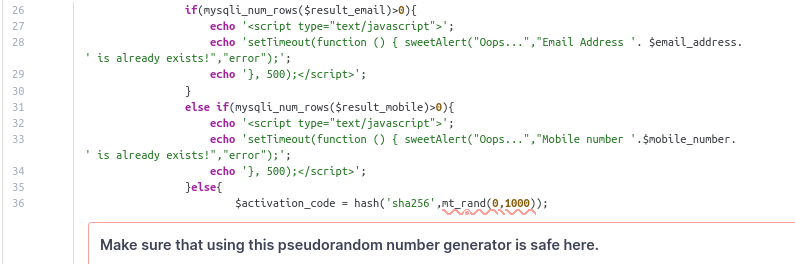
Estas vulnerabilidades se encuentran en el OWASP – Top 10 2021 en la tercera categoría denominada “Inyección”.

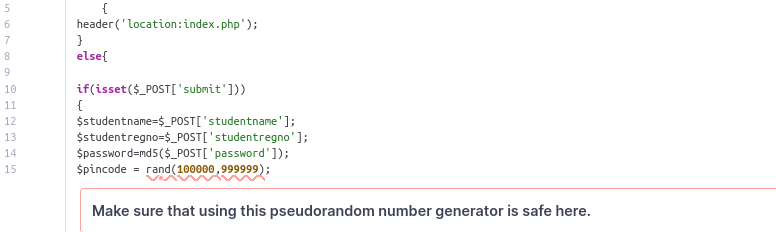
Esta vulnerabilidad podría tener una puntuación CVSS alta, adjuntando grafico del resultado:



3.2.- RIESGO MEDIO

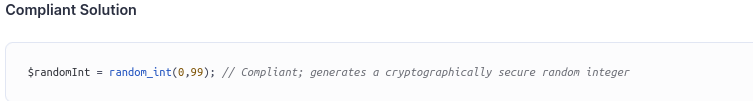
Se han encontrado 2 presuntas debilidades criptográficas, concretamente en 2 partes del código:





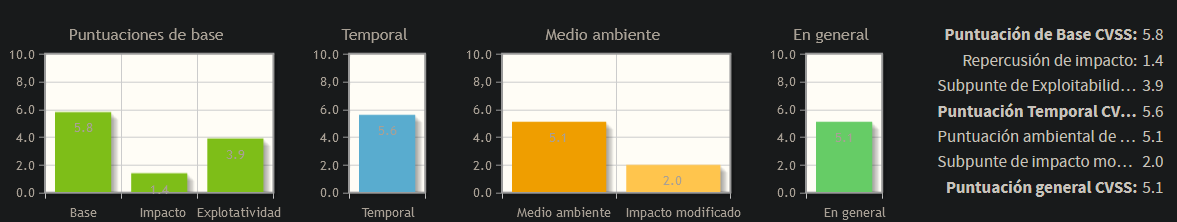
Estas vulnerabilidades podrían ser usada por atacantes para adivinar valores pseudoaleatorios producidos por funciones PHP (RAND) mediante ataques de fuerza bruta, debiendo ser verificado, y corregido, en caso afirmativo, recomendando el uso de funciones que se basen en un generador de valores aleatorios fuertes y seguros (*Random*\_int(), *Random\_bytes(),* etc). En caso que desee continuar usando la función *Random\_pseudo\_bytes ()* verifique el parámetro *crypto\_strong*, el cual permite verificar si los bytes pseudoaleatorios generados pueden ser utilizados de manera segura en operaciones criptográficas (generación de claves o tokens de autenticación), indicando:

* true: si los datos aleatorios generados son criptográficamente seguros.
* false: si la fuente aleatoria no es suficientemente fuerte para fines criptográficos, aunque si lo sean para otros propósitos menos críticos.



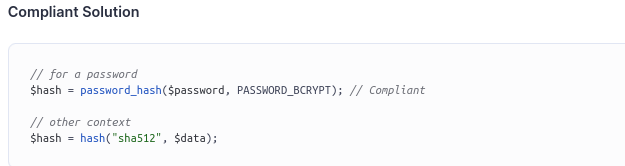
Estas debilidades pertenecen a la OWASP – Top 10 2021, dentro de la categoría 2 llamada “Fallos Criptográficos”.

Estas vulnerabilidades podrían tener una puntuación CVSS medio, adjuntando grafico del resultado:



3.3.- RIESGO BAJO

Se han encontrado 43 supuestas vulnerabilidades de riesgo bajo que podrían afectar a los algoritmos hash criptográficos (MD2, MD4, MD5, MD6, HAVAL-128, HMAC-MD5, etc), siendo éstos, usados para el almacenamiento de datos sensibles (contraseñas) , generadores de token de seguridad o para el calculo de la integridad de los mensajes, por lo que se recomienda su verificación, y el uso de alternativas más seguras (SHA-256, SHA-512, SHA-3, bcrypt, scrypt, etc), siendo algoritmos más fuertes ante ataques de fuerza bruta.



Estas debilidades pertenecen a la OWASP – Top 10 2021, dentro de la categoría 2 llamada “Fallos Criptográficos”.