

# APPS SEGURAS

## Caso práctico 2.

---

Ramiro de Santos



## **INDICE:**

- 1. Pasos para la ejecución del análisis de nuestro proyecto-----pag3**
- 2. Análisis de vulnerabilidad-----pag6**
- 3. Vulnerabilidades detectadas en el fallo SSL/TSL-----pag9**
- 4. Solución-----pag10**

## 1. Pasos para la ejecución del análisis de nuestro proyecto

Empezamos por instalar la versión 17 de jdk y la descargar la versión 10 de sonarcube.

Seguidamente, ejecutamos sonarcube:

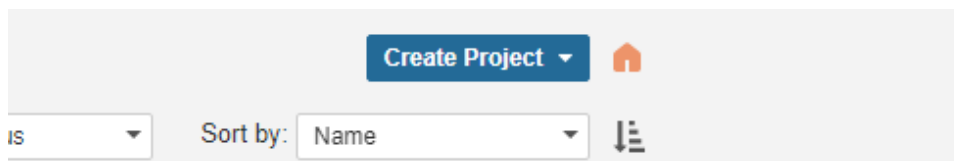
```
C:\Users\34669\Desktop\MASTER FULL STACK\APPS SEGURAS\sonarcube-10.0.0.68432\sonarcube-10.0.0.68432\bin\windows-x86-64>StartSonar.bat
Starting SonarQube...
```

Si todo funciona correctamente, veremos el siguiente mensaje:

```
2023.04.22 18:04:44 INFO app[][o.s.a.SchedulerImpl] Process[ce] is up
2023.04.22 18:04:44 INFO app[][o.s.a.SchedulerImpl] SonarQube is operational
```

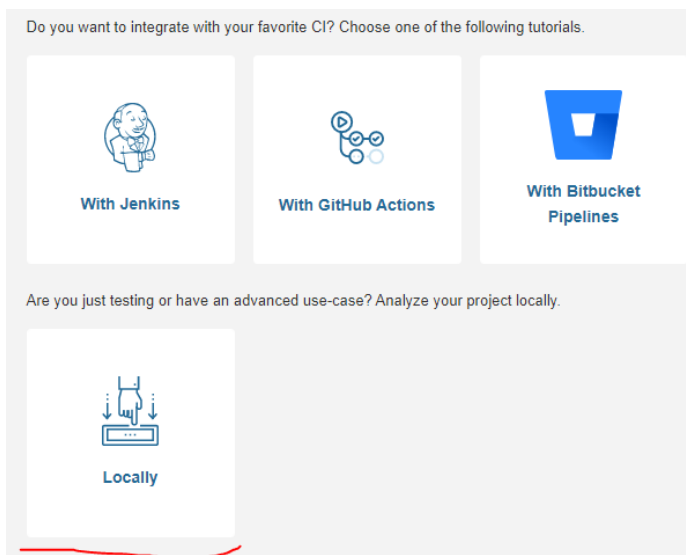
Con sonarcube ejecutado, abrimos el navegador y vamos al localhost:9000, que es el puerto por el que escucha sonarcube. Una vez dentro, necesitamos conectar sonarcube con el código fuente del proyecto que queremos auditar.

En localhost:9000, nos vamos a create project



y le damos un nombre identificativo y también es recomendable incluir en dicho nombre la fecha sobre la que haremos el análisis. En mi caso “openemr-master22042023”.

Clickamos el botón de set up y en la siguiente pantalla seleccionamos en este caso locally, porque tenemos descargado el código fuente.



Seleccionamos una caducidad (en mi caso, que no expira) y sonarcube nos proporciona un token que será la referencia a nuestro proyecto.

Ahora seleccionamos la opción que mejor se ajusta al código que queremos valorar

What option best describes your build?

Maven

Gradle

.NET

Other (for JS, TS, Go, Python, PHP, ...)

E indicamos nuestro sistema operativo

What is your OS?

Linux

Windows

macOS

Sonarcube nos proporciona un comando para ejecutar el análisis

Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner.bat -D"sonar.projectKey=aa" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.token=sqp_524cf482783401edcfdb5394022a058639fec46c"
```

Copy

En este punto, debemos descargar sonar escanner, en mi caso la versión para windows.

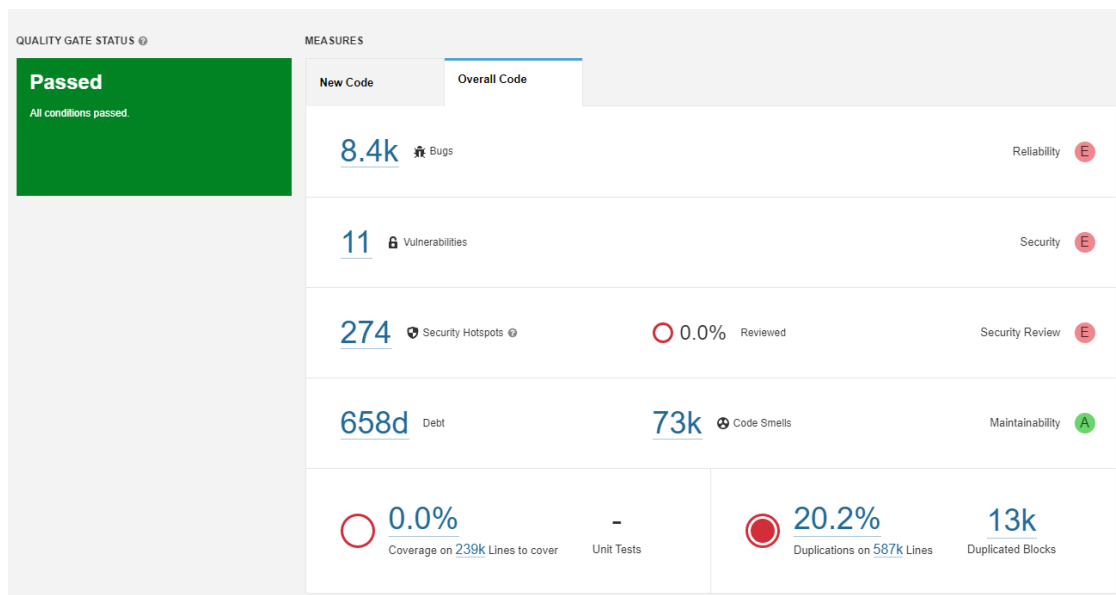
Una vez descargado sonar escanner, abrimos otro CMD, vamos a la ruta del proyecto que queremos analizar y pegamos la ruta en la que se encuentra nuestro sonar escanner seguido del comando que nos ha proporcionado sonarcube.

```
C:\Users\34669>cd C:\Users\34669\Desktop\MASTER_FULL_STACK\APPS_SEGURAS\EJERCICIO2\openemr-master  
C:\Users\34669\Desktop\MASTER_FULL_STACK\APPS_SEGURAS\EJERCICIO2\openemr-master>C:\Users\34669\Desktop\sonar-scanner-4.8.0.2856-windows\bin\sonar-scanner.bat -D"sonar.projectKey=openemr-master22042023" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.token=sqp_27108d467dcde0a2b7a2393630e0fdb4c6fd2a9"
```

Ejecutamos el comando en la ruta del proyecto y en este momento comienza el análisis de nuestro código. Una vez termina el análisis, ya podremos comenzar la auditoría.

```
INFO: Analysis report uploaded in 336ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=openemr-master22042023
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AYepwdtSvbepszyYgib
INFO: Analysis total time: 9:44.218 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 9:46.896s
INFO: Final Memory: 53M/197M
INFO: -----
```

Volvemos a nuestro localhost:9000 y debemos obtener un resultado como el siguiente:



Ahora, debemos analizar una de las 11 vulnerabilidades que hemos obtenido.

ccdaservice/ccda_gateway.php	<div> <input type="checkbox"/> </div> <div> <b>Enable server certificate validation on this SSL/TLS connection.</b> </div> <div> 52 minutes ago ▾ L106 🔗 ⚙️ </div> <div> 🔒 Vulnerability ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment </div> <div> 🔗 cwe, owasp-a3, owasp-a6, owasp-m3, ... ▾ </div>
	<div> <input type="checkbox"/> </div> <div> <b>Enable server hostname verification on this SSL/TLS connection.</b> </div> <div> 52 minutes ago ▾ L107 🔗 ⚙️ </div> <div> 🔒 Vulnerability ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment </div> <div> 🔗 cwe, owasp-a3, owasp-a6, owasp-m3, ... ▾ </div>
contrib/utill/deidentification/deidentification.php	<div> <input type="checkbox"/> </div> <div> <b>Add password protection to this database.</b> </div> <div> 52 minutes ago ▾ L66 🔗 ⚙️ </div> <div> 🔒 Vulnerability ▾ 🔴 Blocker ▾ 🔵 Open ▾ Not assigned ▾ 45min effort Comment </div> <div> 🔗 cwe, owasp-a2, owasp-a3 ▾ </div>
interface/eRxXMLBuilder.php	<div> <input type="checkbox"/> </div> <div> <b>Enable server certificate validation on this SSL/TLS connection.</b> </div> <div> 52 minutes ago ▾ L104 🔗 ⚙️ </div> <div> 🔒 Vulnerability ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment </div> <div> 🔗 cwe, owasp-a3, owasp-a6, owasp-m3, ... ▾ </div>
interface/eRx_xml.php	<div> <input type="checkbox"/> </div> <div> <b>Enable server certificate validation on this SSL/TLS connection.</b> </div> <div> 52 minutes ago ▾ L949 🔗 ⚙️ </div> <div> 🔒 Vulnerability ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment </div> <div> 🔗 cwe, owasp-a3, owasp-a6, owasp-m3, ... ▾ </div>
library/maviq_phone_api.php	<div> <input type="checkbox"/> </div> <div> <b>Enable server certificate validation on this SSL/TLS connection.</b> </div> <div> 52 minutes ago ▾ L54 🔗 ⚙️ </div> <div> 🔒 Vulnerability ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment </div> <div> 🔗 cwe, owasp-a3, owasp-a6, owasp-m3, ... ▾ </div>
	<div> <input type="checkbox"/> </div> <div> <b>Enable server hostname verification on this SSL/TLS connection.</b> </div> <div> 52 minutes ago ▾ L55 🔗 ⚙️ </div> <div> 🔒 Vulnerability ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment </div> <div> 🔗 cwe, owasp-a3, owasp-a6, owasp-m3, ... ▾ </div>
modules/sms_email_reminder/sms_clickatell.php	<div> <input type="checkbox"/> </div> <div> <b>Enable server certificate validation on this SSL/TLS connection.</b> </div> <div> 52 minutes ago ▾ L339 🔗 ⚙️ </div> <div> 🔒 Vulnerability ▾ 🔴 Critical ▾ 🔵 Open ▾ Not assigned ▾ 5min effort Comment </div> <div> 🔗 cwe, owasp-a3, owasp-a6, owasp-m3, ... ▾ </div>

## 2. Análisis de vulnerabilidad

Vamos a analizar la siguiente vulnerabilidad:

interfaz/eRxXMLBuilder.php	<div> <input checked="" type="checkbox"/> </div> <div> <b>Habilite la validación del certificado del servidor en esta conexión SSL/TLS.</b> </div> <div> Hace 1 día ▾ L 104 🔗 ⚙️ </div> <div> 🔒 Vulnerabilidad ▾ 🔴 Crítico ▾ 🔵 Abierto ▾ No asignado ▾ 5 minutos de esfuerzo Comentario </div> <div> 🔗 cwe, owasp-a3, owasp-a6, owasp-m3, ... ▾ </div>
----------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Empecemos por definir qué es una conexión SSL/TSL

El protocolo SSL, "Secure Socket Layer" (en español, capa de puertos seguros), es el predecesor del protocolo TLS "Transport Layer Security" (Seguridad de la Capa de Transporte, en español). Se trata de protocolos criptográficos que proporcionan privacidad e integridad en la comunicación entre dos puntos en una red de comunicación. Esto garantiza que la información transmitida por dicha red no pueda ser interceptada ni modificada por elementos no autorizados, garantizando de esta forma que sólo los emisores y los receptores legítimos sean los que tengan acceso a la comunicación de manera íntegra.

## Sonaracube nos indica que el problema está en que no está habilitada la validación del certificado de servidor en la conexión SSL/TSL

¿Dónde está el problema?

¿Por qué es esto un problema?


openemr-master22042023

interfaz/ eRxXMLBuilder.php

Ver todos los problemas en este archivo

```
99 ... $datos = matriz ( 'RxInput' => $xml );
100
101 curl_setopt( $curlHandler , CURLOPT_URL, $esto -> getGlobals ()->getPath());
102 curl_setopt( $curlHandler , CURLOPT_POST, 1 );
103 curl_setopt( $curlHandler , CURLOPT_POSTFIELDS, 'RxInput=' . $xml );
104 curl_setopt ( $ curlHandler , CURLOPT_SSL_VERIFYPEER,0);

105 curl_setopt( $curlHandler , CURLOPT_FOLLOWLOCATION, 1 );
106 curl_setopt( $curlHandler , CURLOPT_COOKIESESSION, verdadero);
107 curl_setopt( $curlHandler , CURLOPT_COOKIEFILE, $sitePath . '/newcrop-cookiefile' );
108 curl_setopt( $curlHandler , CURLOPT_COOKIEJAR, $sitePath . '/newcrop-cookiefile' );
109 curl_setopt( $curlHandler , CURLOPT_COOKIE, session_name() . '=' . session_id());
110 curl_setopt( $curlHandler , CURLOPT_USERAGENT, 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)' );
111 curl_setopt( $curlHandler , CURLOPT_RETURNTRANSFER, verdadero);
112
113 $resultado = curl_exec( $curlHandler ) o die (curl_error( $curlHandler ));
```

 **Habilite la validación del certificado del servidor en esta conexión SSL/TLS.**

## Sonarcube nos indica por qué esto puede ser un problema:

¿Dónde está el problema?

¿Por qué es esto un problema?

La validación de los certificados X.509 es esencial para crear sesiones SSL/TLS seguras que no sean vulnerables a los ataques de intermediarios.

La validación de la cadena de certificados incluye estos pasos:

- El certificado lo emite su autoridad de certificación principal o la CA raíz en la que confía el sistema.
- Cada CA puede emitir certificados.
- Cada certificado de la cadena no está caducado.

No se recomienda reinventar la rueda implementando una validación de cadena de certificados personalizada.

Las bibliotecas TLS proporcionan funciones de validación de certificados integradas que deben usarse.

Ejemplo de código no conforme

```
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSO); // No conforme
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0); // No conforme
```

**Los certificados X.509 son un tipo de certificado digital utilizado en la seguridad informática para establecer conexiones seguras a través de Internet entre dos**

entidades. Un certificado X.509 permite que sitios web, usuarios, empresas y otras organizaciones prueben sus identidades en Internet.

Las autoridades de certificación (CA) son las entidades responsables de emitir certificados digitales X.509. Hacen esto para asegurarse de que cada certificado que emiten se adhiere a estándares de autenticación específicos y cumple con requisitos de validación específicos.

Para que una parte pueda confiar en una clave pública de otra entidad, por ejemplo para autenticar la identidad de esa entidad, la clave pública se obtendrá de una fuente confiable. Dicha fuente, denominada autoridad de certificación (CA), certifica una clave pública mediante la emisión de un certificado de clave pública que vincula la clave pública a la entidad que posee la clave privada correspondiente.

### En nuestro código

```
curl_setopt ( $ curlHandler , CURLOPT_SSL_VERIFYPEER,0);
```

**CURLOPT\_SSL\_VERIFYPEER** es una opción de configuración de cURL (una herramienta de transferencia de datos en línea) que permite activar o desactivar la validación del certificado del servidor durante una conexión SSL/TLS. Si esta opción está activada, cURL verificará que el certificado del servidor sea válido y haya sido emitido por una autoridad de certificación confiable.

Por defecto **CURLOPT\_SSL\_VERIFYPEER** está en **true** (o 1). Si se establece en **false** (o 0), la validación del certificado se desactivará y la conexión se establecerá aunque el certificado del servidor sea inválido o no confiable. Esto nunca debe usarse en producción ya que compromete la seguridad de la conexión.



### 3. Vulnerabilidades detectadas en el fallo SSL/TSL

Como indica Sonarcube, este fallo está relacionado con las siguientes vulnerabilidades:

- OWASP Top 10 2021 Category A2 - Cryptographic Failures:

X.509 es un formato estándar para certificados de clave pública, documentos digitales que asocian de forma segura pares de claves criptográficas con identidades como sitios web, individuos u organizaciones.

- OWASP Top 10 2021 Category A5 - Security Misconfiguration:

Las sesiones SSL/TLS seguras están relacionadas la categoría OWASP A5, ya que una configuración incorrecta o insegura de los servidores web y de las sesiones SSL/TLS puede permitir que los ataques, como por ejemplo, el “man-in-the-middle” que consiste en un ataque en el que se adquiere la capacidad de leer, insertar y modificar a voluntad. El atacante debe ser capaz de observar e interceptar mensajes entre las dos víctimas y procurar que ninguna de las víctimas conozca que el enlace entre ellos ha sido violado. El ataque MitM es particularmente significativo en el protocolo original de intercambio de claves de Diffie-Hellman, cuando este se emplea sin autenticación. Hay ciertas situaciones donde es bastante simple, por ejemplo, un atacante dentro del alcance de un punto de acceso wifi sin cifrar, donde este se puede insertar como intermediario.

- OWASP Top 10 2021 Category A7 - Identification and Authentication Failures

Las sesiones SSL/TLS seguras están relacionados con esta vulnerabilidad ya que ambas se centran en proteger la autenticación y la autorización en aplicaciones web.

- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure

Por ejemplo, si un atacante puede inyectar código malicioso en una consulta SQL o en un comando de sistema, puede obtener acceso no autorizado a información

confidencial o incluso tomar el control del servidor. Si esta información se envía a través de una sesión SSL/TLS no segura, un atacante puede interceptarla y acceder a ella.

Además, las vulnerabilidades de inyección pueden explotarse para alterar la configuración de la sesión SSL/TLS y permitir que un atacante suplante al servidor y se haga pasar por él, engañando al cliente para que envíe información confidencial.

- OWASP Top 10 2017 Category A6 - Security Misconfiguration

Una configuración insegura de la sesión SSL/TLS puede permitir a los atacantes explotar vulnerabilidades para comprometer la seguridad de la aplicación.

#### 4. Solución

Como solución se puede usar el valor 1 o true en el valor de `CURLOPT_SSL_VERIFYPEER`, lo que provoca que se validen los certificados y tengamos por tanto una conexión segura.

```
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, TRUE)
```

Where is the issue?

Why is this an issue?

```
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE); // Noncompliant
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0); // Noncompliant
```

#### Compliant Solution

```
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, TRUE); // Compliant; default value is TRUE
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 1); // Compliant
```

#### See

- [OWASP Top 10 2021 Category A2](#) - Cryptographic Failures
- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [OWASP Top 10 2021 Category A7](#) - Identification and Authentication Failures
- [OWASP Top 10 2017 Category A3](#) - Sensitive Data Exposure
- [OWASP Top 10 2017 Category A6](#) - Security Misconfiguration
- [Mobile AppSec Verification Standard](#) - Network Communication Requirements
- [OWASP Mobile Top 10 2016 Category M3](#) - Insecure Communication
- [MITRE, CWE-295](#) - Improper Certificate Validation