Trabajo Practico Seguridad Web

Alumnos: Nicolas Russo, Maximiliano Arrieta, Federico Wasylciw, Rodrigo Vazquez

Aplicación de documentación online

La aplicación insegura a ser vulnerada se trata de una app web cuya función es facilitar la comunicación entre el personal de gestión y los empleados de una empresa. Para ello, usuarios con permiso de administrador suben documentos para que los mismos sean consultados por usuarios con permiso de lector. Algunas características sobre el sitio:

- La página de inicio cuenta con un login y una barra de búsqueda donde se pueden buscar documentos.
- Un usuario no autenticado puede utilizar la barra de búsqueda pero no ve ningún documento, ya que el sitio no permite ver documentos a este tipo de usuario
- Un administrador puede descargar documentos ya subidos
- La persistencia es gestionada a través de SQLite

Ataque de vulnerabilidades concatenadas

1. SQL Injection:

   El atacante realiza un ataque de SQL Injection sobre la barra de búsqueda, tipeando:" ' UNION SELECT id, username FROM User –" obteniendo como resultado todos los nombres de usuario de la tabla users

2. JWT Forgery:

   El atacante nota que el sistema implementa autenticación a través de OAuth.

   Creando una aplicación propia en Oauth, es posible registrar un usuario con el nombre de alguno de los usuarios obtenidos en el paso previo. A partir de esto, el atacante puede obtener un jwt de identificación que tenga codificado el nombre de usuario de un usuario válido en el sistema que quiere vulnerar.

   Debido a una mala implementación de oauth en el backend, el sistema no valida firma ni audiencia.

   El atacante inserta su jwt en el header de Authorization (utilizando, por ejemplo, Postman) y se autentica como un usuario válido.

3. IDOR:

   El atacante observa que se puede acceder a una página donde se listan los documentos compartidos bajo la URL 'https://sitio.com/documents', y al hacer clic en uno accede a la URL 'documents/{id}'. Prueba escribir a mano la URL 'https://example.com/documents/{id}/download', lo que lo lleva a una vista con el botón de descarga del documento. Este recurso sólo debería ser accesible por un administrador, pero el permiso para accederlo no es correctamente validado por la aplicación

4. Directory traversal

   El atacante observa que el botón de descarga posee una redirección a la URL 'https://example.com/download?file=report.txt'.

accede a la dirección ‘https://example.com/download?file=..%2F..%2F..%2Fappsettings.json’ y descarga el archivo appsettings.json de la aplicación.

Leyendo el archivo json descargado, el atacante encuentra el path desde root hasta el archivo .sqlite donde se encuentra toda la persistencia del sistema. Explotando la vulnerabilidad Directory traversal de nuevo, el atacante logra descargar dicho archivo.

5. Ejecución Remota de Código (RCE)

   La base de datos se encuentra encriptada, pero como cuenta con el archivo .sqlite se puede acceder por un ataque de fuerza bruta.

   El atacante puede leer el archivo .sqlite utilizando la consola o alguna herramienta como sqlbeaver y encuentra la tabla ‘System’ donde se encuentra entre otros datos el nombre de usuario del administrador.

   Repitiendo el proceso utilizado para entrar como un usuario, el atacante logra ingresar como administrador al sistema.

   Una vez logueado como administrador, la aplicación permite al atacante la subida de archivos sin aplicar restricciones adecuadas, por lo que puede subir un archivo con extensión .php o bash con código malicioso. Cualquier usuario que acceda a “ver” el archivo subido por el atacante estará ejecutando su código sin su consentimiento