Propuestas de Solución para cada Vulnerabilidad

1. **A05: SECURITY MISCONFIGURATION - Carga de Archivos**

   Nuestro sistema de carga de archivos se relaciona mucho cabal con la vulnerabilidad de A05: Security Misconfiguration y A08: Software and Data Integrity Failures, pues como no tenemos una configuración segura para la carga de archivos, se compromete la integridad del sistema con algún archivo malicioso. Una buena solución es validar por tipo de archivo según corresponda y rechazar cualquier archivo que no cumpla con la política de extensión y peso establecida. También almacenar los archivos subidos fuera del directorio público para que no sean accesibles desde fuera.

2. **A01: BROKEN ACCES CONTROL - Rutas expuestas**

   Como tenemos algunas rutas expuestas fuera del middleware de autenticación, se relaciona mucho la vulnerabilidad a A01: Broken Access Control al permitir acceso a datos restringidos. En este caso, debemos proteger los endpoints expuestos del módulo de /productos envolviendolas en el middleware de auth y forerunner (nuestro guardián de permisos de rol) para evitar que usuarios no autenticados y/o que no tengan el rol autorizado accedan a módulos que puedan comprometer datos importantes.

3. **A02: CRYPTOGRAPHIC FAILURES - Exposición de datos sensibles**

   El tener campos expuestos en texto claro, se relaciona con la vulnerabilidad
   A02: Cryptographic Failures. Para este caso, la solución sera proteger estos datos sensibles, deshabilitando APP_DEBUG en el entorno de producción para no divulgar logs de errores o warnings que puedan comprometer información sensible. Además, utilizar HTTPS para el trafico web, de modo que aseguremos que los datos como credenciales, DPI, etc. viajen cifrados. También implementar el cifrado a nivel de base de datos como lo hacemos con la contraseña, utilizando Crypt que Laravel tiene en caso de que alguna vez la base de datos se vea comprometida. Además, debemos eliminar las credenciales por defecto que tenemos en el entorno de desarrollo para que no se pasen al entorno de producción también, allí deberá tener credenciales mucho más robustas.

4. **A03: INJECTION, CROSS-SITE SCRIPTING - Vulnerabilidad XSS**

   En la parte del menu del layout principal, lo renderizamos con HTML generado desde la base de datos usando sintaxis sin escape, especificamente para la parte del icono, por lo que esto se relaciona con la vulnerabilidad A03: Injection, Cross-Site Scripting. En este caso, podemos escapar el campo de icono y restringir solo a una lista segura de clases CSS como lo pueden ser los nombres de los iconos de FontAwesome que es cabal lo que usamos para los iconos del sistema.

5. **A05: SECURITY MISCONFIGURATION - Configuración de seguridad inadecuada**

   El tener las opciones de debug activadas a la hora de irnos a producción, no tener las cookies endurecidas y seguir usando valores predeterminados igual cuando nos vayamos a producción puede generar problemas de seguridad por mala configuración, por lo mismo es que está relacionada con A05: Security Misconfiguration. Para solucionar esto, debemos establecer la configuración distinta y robusta en producción del .env. Configurar la cookie de sesión asegurando que solo viajen por HTTPS y no sean enviadas en otros contextos. También implementar encabezados de seguridad en las respuestas HTTP con el middleware que trae Laravel para proteger el sistema.