Curso de Desarrollo seguro

Las fases por las que se debe pasar son las siguientes:

- Requisitos

  En esta fase debemos solventar posibles vulnerabilidades conocidas, hacer un análisis de los posibles riesgos, utilizar frameworks extendidos y respaldados y tecnologías con mantenimiento y actualizaciones. Esta parte es muy importante ya que nos ayuda a reducir costes a futuro.

- Arquitectura y diseño

En esta fase debemos identificar vulnerabilidades en las tecnologías que hemos escogidos y intentar protegerlas, asegurar los recursos de datos, emplear protocolos seguros y estudiar estándares y arquitecturas típicas. Este es el primer paso para la creación de una aplicación segura.

- Implementación

Debemos evitar el uso de funciones deprecated, realizar el control de errores así como controlar los flujos de datos. Además, debemos tender a realizar los procesos lo más simples posibles.

- Testeo

Debemos testear las vulnerabilidades que eran conocidas y asegurarnos de que las hemos protegido correctamente, realizar fuzz testing que es una forma de testeo automatizada y pentesting que se basa en intentar atacar nuestro software para descubrir posibles debilidades.

- Despliegue

Deberemos configurar de manera segura tanto cada servicio como sistema, además debemos gestionar de manera correcta los permisos y operaciones para que cada usuario solo pueda acceder a las funciones a las que está destinado

- Mantenimiento

Se deben tener todas las librerías y framworks con versiones actualizadas, así como las tecnologías que usemos y los procedimientos que hayamos definido anteriormente ya que las amenazas pueden cambiar con el tiempo. Debemos revisar los logs con frecuencia para detectar posibles problemas

Para garantizar un desarrollo seguro debemos validar todas las entradas que nos envía el usuario para comprobar que sean válidas, ya que si no hacemos esto podremos ser vulnerables a ataques del tipo sql injection.

Debemos codificar siempre las salidas para evitar la insercción de código, además, debemos cifrar los datos siempre antes de guardarlos para que solo se puedan leer si se tiene la clave de cifrado.

También debemos controlar el buffer overflow por el cual nos pueden subir archivos demasiado grandes con el fin de bloquear la aplicación, para solucionar esto debemos poner restricciones como el tipo de archivo, el tamaño...

La seguridad en los procesos y procedimientos la garantizamos mediante una buena autenticación y manejo de contraseñas, las cuales deben ser guardadas habiendo sido cifradas con anterioridad y obligando a los usuarios a usar contraseñas complejas para que resulte más complicado realizar un ataque de fuerza, en caso de aplicaciones web usar el método POST para que no aparezcan los parámetros en la url.

Debemos realizar un buen manejo de sesiones, es decir, los identificadores de sesión deben ser únicos y generados de manera aleatoria, cuando realizamos un logout debemos eliminar por completa, no hacer que el usuario no pueda acceder y que la sesión siga activa, cambiar los id de sesión de manera periódica, terminar las sesiones de manera periódica, configurar el atributo secure para las cookies.

No debemos mostrar información sobre los errores que se produzcan, por ejemplo si estamos en un login no debemos mostrar el error con la consulta que se ha realizado, ya que dejaríamos a la vista que es vulnerable a sql injection.

Seguridad en el entorno

Debemos controlar los accesos, para ello debemos comprobar todo en el servidor, no a nivel cliente, y en caso de fallo no dar información que de una idea de cómo se gestionan las sesione, forzar la autenticación para cualquier petición para garantizar de que el usuario es quien dice ser, limitar el número de transacciones para evitar ataques de fuerza bruta, dar privilegios mínimos a cada usuario, es decir no dar privilegios que no necesite un usuario.

Para protyeger los datos debemos hacer que solo puedan acceder a ellos personas autorizadas mediante los privilegios de cada usuario, encriptar los datos, eliminar el acceso a información innecesaria que pueda dar información innecesaria sobre nuestra aplicación

Debemos manejar los ficheros con consciencia, es decir poniendo restricciones como el tipo de fichero, restringiendo el tamaño...

Debemos proteger las comunicaciones entre el servidor y el cliente cifrando la información mientras está viajando, debemos utilizar un certificado TLS válido, es decir que no esté expirado.

Asegurar la configuración del sistema controlando los métodos HTTP que están permitidos, eliminando cualquier tipo de información que pueda resultar útil para un atacante y no mostrar nuestra estructura de ficheros en robots.txt ya que este fichero es accesible por cualquier persona

Para proteger los datos de una base de datos debemos usar queries fuertemente parametrizadas, validar las entradas y codificar las salidas, usar contraseñas seguras, restringir los permisos a los usuarios que tengan acceso a la base de datos, la configuración de la misma debe estar cifrada y debemos cerrar las conexiones ASAP