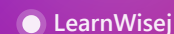


El modelo de seguridad de Wisej.NET

Guía de la lección · Guía del laboratorio / examen



Guía de la lección

Lo que el navegador puede y no puede ver en una interfaz de usuario del lado del servidor, el límite de sesión, el modelo de amenazas y dónde se ubica cada control. Qué significa esto para un desarrollador que protege una aplicación Wisej.NET en producción, y cómo leer este módulo.



Enfoque en producción

Lo que el navegador puede y no puede ver en una interfaz de usuario del lado del servidor, el límite de sesión, el modelo de amenazas y dónde se ubica cada control.

1. El problema en producción

AccessOps es un portal interno de operaciones: departamentos, empleados, documentos confidenciales, solicitudes de aprobación y una consola de administración. Todo el que puede llegar a él ya trabaja para la empresa, y precisamente por eso el trabajo de seguridad se aplaza, y por eso un único portal acaba conteniendo bandas salariales, cartas disciplinarias y el historial de aprobaciones de cada excepción que la empresa ha concedido. Antes de que exista una sola línea de código de autenticación necesitas tres respuestas: qué tiene el navegador, por dónde pasan los límites y cuáles de las cosas que podrían salir mal merecen protegerse primero.

2. Conceptos clave

Concepto	Por qué importa
modelo de interfaz del lado del servidor	Los controles son objetos C# en el proceso del servidor; el navegador tiene un reflejo renderizado y un canal de eventos, así que tu código y tus consultas nunca se descargan.
carga útil del estado del widget	Los valores de propiedades que Wisej.NET serializa para que el navegador pueda dibujar un control. Todo lo que asignas viaja por la red, también en los controles que el usuario no ve.
límite de confianza	Una línea donde los datos pasan entre partes con distinto nivel de confianza. Todo lo que la cruza lo vuelve a comprobar el lado más fiable, porque el otro lado puede mentir.
límite de sesión	La frontera entre un navegador y su propio estado en el servidor, en <code>Application.Session</code> . Las decisiones salen del estado que está en el lado del servidor, nunca de un valor del navegador.
punto de entrada	Cualquier vía de acceso que un atacante pueda alcanzar: la página de inicio de sesión, los controladores de eventos, las subidas, las descargas, los handlers HTTP personalizados, las llamadas de interoperabilidad y el prompt del asistente de IA.
STRIDE	Seis categorías para nombrar amenazas —suplantación, manipulación, repudio, divulgación de información, denegación de servicio y elevación de privilegios— para que una revisión sea sistemática.
ubicación de controles	Elegir el límite en el que se puede hacer cumplir una regla. Una regla que solo existe en la interfaz es una sugerencia; la misma regla dentro de un servicio es un control.

3. Desarrollo de la lección

Lo que el navegador recibe realmente

Una aplicación Wisej.NET es un programa .NET corriente. Los controles que colocas en una ventana son objetos C# del espacio de nombres `Wisej.Web` que viven en el proceso del servidor, y el navegador tiene un reflejo renderizado de ellos más un canal para los eventos. El usuario hace clic; el navegador notifica el clic; el servidor lanza el controlador en la sesión propietaria del widget; tu código se ejecuta junto a la base de datos; las propiedades modificadas se serializan de vuelta. Ningún ensamblado, consulta LINQ, regla de validación ni cadena de conexión se envía nunca al cliente.

Lo que sí cruza la red es cada valor de propiedad que asignas, porque el navegador no puede dibujar lo que no se le ha dicho: el texto de una etiqueta, los elementos de un `ComboBox`, las celdas de cada fila que cargaste en un `DataGridView`, un tooltip, un `Tag` aparcado en un nodo de árbol por comodidad. Poner `Visible` a false en una columna cambia lo que se dibuja, no lo que se envió. Lo mismo vale para lo que sacas deliberadamente del proceso: los archivos transmitidos con `Application.Download`, los archivos aceptados por un control `Upload`, los handlers HTTP personalizados que registras y cualquier JavaScript que pidas ejecutar al cliente, con sus argumentos.

De ahí sale la primera regla de AccessOps: lo que sale del servidor lo decide una consulta, no una propiedad del diseñador. La cuadrícula del directorio recibe una proyección de las columnas que cualquier usuario puede ver; la banda salarial es una segunda llamada, comprobada.

```
// ✗ the whole record crosses the wire
gridEmployees.DataSource = _repo.GetAllEmployees();
gridEmployees.Columns[5].Visible = false;    // Salary

// ✓ the server decides what leaves the process
gridEmployees.DataSource =
    _employees.GetDirectory(_session.CurrentUserId);
```

Cuatro límites, una aplicación

El primer límite está entre el navegador y la sesión de Wisej.NET. Todo lo que llega a través de él es una afirmación: qué widget se pulsó, qué fila de la cuadrícula es la actual, qué se escribió, qué id de documento menciona un enlace. Una afirmación no es un hecho, y la sesión es el primer lugar donde puede existir un hecho.

El segundo separa una sesión del proceso que las aloja a todas. Las sesiones comparten memoria, así que un campo estático, un servicio singleton o una caché sin clave por usuario es visible para todos los que tienen la sesión iniciada en ese servidor. Por eso la identidad verificada pertenece a `Application.Session` y nunca a un «usuario actual» estático. El trabajo en segundo plano iniciado con `Application.StartTask` sobrevive a la solicitud que lo lanzó, así que debe llevar consigo la identidad que recibió en lugar de ir a buscar la sesión que sea la actual cuando termine.

El tercero y el cuarto miran hacia fuera, a la base de datos y a los servicios externos: el proveedor de identidades que añadirás en el Módulo 4, el almacén de documentos, el servidor de correo. Da al login de la base de datos solo los derechos que AccessOps necesita y parametriza todos los comandos, para que un nombre de departamento escrito en un cuadro de búsqueda no pueda convertirse en SQL. Asigna un responsable a cada límite, porque un control que no está en ninguno de ellos no tiene nada que detener.

Un modelo de amenazas para AccessOps

Un modelo de amenazas son tres listas y una tabla. Primero los activos: las fichas de empleados, el almacén de documentos, las decisiones de aprobación, el registro de auditoría que las demuestra y las credenciales que lo protegen todo. Después los puntos de entrada: la página de inicio de sesión, cada controlador de eventos, el control `Upload`, cada descarga, cualquier handler HTTP personalizado y el asistente de IA, cuyo prompt es una entrada no fiable como cualquier otra. Por último las amenazas, nombradas por categoría y no por temor.

- Suplantación: alguien inicia sesión como el responsable de finanzas; autenticación e identidad de sesión en el Módulo 2, inicio de sesión federado en el Módulo 4.
- Manipulación y elevación: alguien que no es responsable aprueba su propia solicitud; autorización dentro del servicio, Módulo 3, no un botón oculto.
- Divulgación de información: un documento llega a alguien ajeno a su departamento, o un término de búsqueda inyectado devuelve filas prohibidas; Módulo 5.

- Repudio: nadie puede demostrar quién aprobó qué, ni cuándo; el registro de auditoría y los ID de correlación, Módulo 6.
- Denegación de servicio: la página de inicio de sesión se convierte en un punto para adivinar contraseñas, o una subida sobredimensionada llena el disco; Módulos 2 y 7.

Valora cada fila dos veces, por probabilidad y por impacto. Un público interno cambia esas cifras sin eliminarlas: un compañero curioso es más probable y menos grave que alguien de fuera, y un empleado que se va llevándose el almacén de documentos es a la vez plausible y grave. La columna más útil es la última, la que nombra el módulo que implementa la mitigación: una fila sin ninguno es o bien un riesgo aceptado, que debería decirlo por escrito, o bien una carencia que has encontrado gratis.

Poner cada control donde se puede hacer cumplir

Un control solo cuenta en un límite que pueda vigilar. Quitar la pestaña Admin, desactivar el botón Approve, filtrar una cuadrícula al departamento actual: todo ello ocurre en el lado no fiable del límite de sesión, así que todo puede eludirse pidiendo un evento que la interfaz ya no ofrece. Merece la pena hacerlo, porque una interfaz que ofrece trabajo que el usuario no puede realizar es una mala interfaz, pero es experiencia de usuario, no seguridad.

Por eso AccessOps coloca cada decisión importante en un servicio del lado del servidor de ese límite: `DocumentService` decide si este llamante puede leer este documento, `ApprovalService` si este llamante puede aprobar esta solicitud, y ambos leen la identidad de la sesión, no de un parámetro que haya proporcionado la pantalla. La interfaz refleja esas decisiones y nunca las toma. Además, así las reglas se pueden probar: un servicio se puede llamar con argumentos hostiles en una prueba unitaria.

El mapa de ubicación de controles del laboratorio tiene una fila por control: la regla, el límite en el que se sitúa, la clase que la hace cumplir y cómo demostrarías que funciona. Convierte el diagrama en un plan que los seis módulos restantes van cumpliendo línea a línea.



Regla de seguridad

Nunca se confía en el navegador. Toda decisión importante —quién es el usuario, qué puede hacer, qué registro puede ver— se toma en el servidor, dentro de un servicio, y la interfaz solo la refleja. Si una regla solo existe como un botón oculto, no existe.



Objetivos de aprendizaje

Al terminar este módulo serás capaz de:

- ✓ Describir lo que recibe realmente un cliente Wisej.NET —widgets renderizados y eventos, no tu código ni tus datos— y lo que aun así cruza la red.
- ✓ Identificar los límites de confianza de una aplicación Wisej.NET: navegador, sesión, proceso del servidor, base de datos y servicios externos.
- ✓ Construir un modelo de amenazas para una aplicación de gestión: identidad, autorización, exposición de datos, inyección, abuso de sesión y denegación de servicio.
- ✓ Colocar cada control de seguridad en el límite en el que se puede hacer cumplir.

Guía del laboratorio / examen

Qué construirás en el laboratorio práctico, el enfoque sugerido y los entregables requeridos.



Laboratorio práctico

Crea la solución AccessOps y redacta su modelo de amenazas: dibuja los límites de confianza, enumera los activos (fichas de empleados, documentos, aprobaciones), identifica los puntos de entrada (inicio de sesión, subidas, descargas, endpoints personalizados, el asistente de IA), valora cada amenaza y asigna cada mitigación al módulo de este curso que la implementa. Construye el esqueleto de la aplicación con una página de inicio de sesión provisional.

Punto de partida

Partes únicamente de un proyecto Wisej.NET Web Application vacío llamado AccessOps, su ventana de inicio generada y los archivos de configuración predeterminados. En este módulo creas el esqueleto —una página de inicio de sesión provisional y la navegación hacia las áreas de departamentos, empleados, documentos, aprobaciones y administración que el resto del curso irá completando— y el modelo de amenazas escrito con el que se medirá cada módulo posterior.

Enfoque sugerido

- ✓ Crea la solución AccessOps y una página de inicio de sesión provisional con un cuadro de nombre de usuario, un cuadro de contraseña y un botón Sign in cuyo controlador, por ahora, solo actualiza una etiqueta de estado, además de pantallas vacías para departamentos, empleados, documentos, aprobaciones y la consola de administración.
- ✓ Dibuja el diagrama de límites de confianza con cuatro regiones —navegador, sesión de Wisej.NET, proceso del servidor, y la base de datos y los servicios externos— y marca en él exactamente lo que cruza cada línea, incluidas las subidas, las descargas, los handlers personalizados y la interoperabilidad con JavaScript.
- ✓ Redacta el inventario de activos y puntos de entrada: en una lista, las fichas de empleados, los documentos, las decisiones de aprobación, el registro de auditoría y las credenciales; en la otra, la página de inicio de sesión, los controladores de eventos de cada pantalla, el control Upload, las descargas, los handlers HTTP personalizados y el asistente de IA.
- ✓ Construye la tabla de amenazas recorriendo cada punto de entrada por la suplantación, la manipulación, el repudio, la divulgación de información, la denegación de servicio y la elevación de privilegios; después valora cada fila por probabilidad e impacto y nombra la mitigación y el módulo que la aporta.
- ✓ Elabora el mapa de ubicación de controles: una fila por control con la regla, el límite en el que se sitúa, la clase que la hará cumplir y la evidencia que mostrarías a un revisor de que funciona.

Entregables requeridos

- ✓ Solución AccessOps con una página de inicio de sesión provisional
- ✓ Diagrama de límites de confianza

- ✓ Inventario de activos y puntos de entrada
- ✓ Tabla de amenazas con probabilidad, impacto y mitigación
- ✓ Mapa de ubicación de los controles de seguridad

Preguntas de repaso

- ✓ Tu cuadrícula de empleados está enlazada a las fichas completas y la columna del salario está oculta en el diseñador. Describe lo que tiene el navegador y nombra el cambio que de verdad saca el salario de la red.
- ✓ Dos personas tienen la sesión iniciada en AccessOps en el mismo servidor. Explica qué partes del estado comparten por defecto y qué tendría que ocurrir para que una de ellas viera el departamento actual de la otra.
- ✓ Elige la amenaza de tu tabla con mayor impacto y defiende la postura contraria: ¿qué tendría que ser cierto en AccessOps para que esa mitigación fuera una pérdida de tiempo para el equipo?



Regla de seguridad

Nunca se confía en el navegador. Toda decisión importante —quién es el usuario, qué puede hacer, qué registro puede ver— se toma en el servidor, dentro de un servicio, y la interfaz solo la refleja. Si una regla solo existe como un botón oculto, no existe.