

Control de acceso basado en roles para aplicaciones .NET

¿Es la mejor forma para la autenticación y los permisos?

1	OBJETIVO DE ESTE DOCUMENTO.....	3
2	CONCEPTOS PRINCIPALES.....	3
2.1	AUTENTIFICACIÓN.....	3
2.2	AUTORIZACIÓN.....	3
2.3	AUDITORÍA.....	3
3	COMPONENTES CLAVES.....	4
3.1	UN REPOSITORIO ASEGURADO PARA ALMACENAR LOS DATOS DE RBAC.....	4
3.2	UN COMPONENTE INTEGRADO EN LA APLICACIÓN.....	4
3.3	UNA CONSOLA DE ADMINISTRACIÓN.....	4
3.4	DOCUMENTACIÓN PARA LOS DESARROLLADORES Y LOS ADMINISTRADORES.....	4
4	¿POR QUÉ LOS PERMISOS DEBEN SER INDEPENDIENTES DEL CÓDIGO?.....	5
4.1	¿QUÉ SIGNIFICA ESTO?.....	5
4.2	PERMISOS Y APLICACIÓN: UN CICLO DE VIDA INCOMPATIBLE.....	5
4.3	INDEPENDENCIA DE LOS ADMINISTRADORES.....	5
4.4	LIBERA AL EQUIPO DE DESARROLLO.....	5
5	PREGUNTAS CLAVE AL DISEÑAR SU SISTEMA RBAC:.....	6
5.1	NECESITA USTED ... ¿ASEGURAR DIVERSAS APLICACIONES EN UN REPOSITORIO CENTRALIZADO?.....	6
5.2	...¿GESTIONAR ROLES COMPARTIDOS?.....	6
5.3	...¿SOPORTAR EL SINGLE SIGN-ON (AUTENTIFICACIÓN ÚNICA)?.....	6
5.4	...¿SOPORTAR DIVERSAS TECNOLOGÍAS (WINFORMS, WEBFORMS, WEBSERVICES)?.....	6
5.5	...¿GENERAR REPORTE SOBRE LAS CUENTAS DE USUARIO Y SUS PERMISOS?.....	6
5.6	...¿CUMPLIR CON REQUERIMIENTOS DE AUDITORÍA (SOX...)?.....	6
5.7	...¿SOPORTAR UN EQUIPO DE DESARROLLADORES GRANDE O INTERNACIONAL?.....	6
5.8	...¿TIPO DE PERMISO: QUÉ TANTO NECESITA ASEGURAR LA APLICACIÓN?.....	6
6	MANTENIMIENTO: LOS COSTOS SUBESTIMADOS.....	7
6.1	APOYAR A LOS DESARROLLADORES Y A LOS ADMINISTRADORES.....	7
6.2	MANTENER LOS PERMISOS COHERENTES CON EL CÓDIGO.....	7
6.3	DESPLEGAR NUEVAS VERSIONES DE LA APLICACIÓN.....	7
6.4	REALIZAR EL CONTROL DE VERSIONES DE LOS DATOS DE SEGURIDAD.....	7
7	DESARROLLO INTERNO, LA SOLUCIÓN CLÁSICA PARA LA SEGURIDAD DE UNA APLICACIÓN.....	8
7.1	EL DESARROLLO Y EL MANTENIMIENTO DEPENDEN DE LOS RECURSOS INTERNOS.....	8
7.2	CUENTAS DE USUARIO ESPECÍFICAS A LA APLICACIÓN.....	8
7.3	ACCESO Y PERMISO: BAJA GRANULARIDAD.....	8
7.4	CÓDIGO ESPECÍFICO EN LA APLICACIÓN.....	8
7.5	UNA SOLUCIÓN SEPARADA PARA CADA APLICACIÓN: PROBLEMAS DE MANTENIMIENTO.....	8
8	VISUAL GUARD, UNA SOLUCIÓN CORPORATIVA PARA LA SEGURIDAD DE SUS APLICACIONES.....	9
8.1	UNA SOLA SOLUCIÓN PARA ASEGURAR Y CENTRALIZAR TODAS SUS APLICACIONES .NET.....	9
8.2	PERMISOS INDEPENDIENTES DEL CÓDIGO.....	9
8.3	HERRAMIENTAS DE ADMINISTRACIÓN DISEÑADAS PARA PERSONAS NO TÉCNICAS.....	9
8.4	DESARROLLAR HERRAMIENTAS PARA GESTIONAR PERMISOS, VERSIONES Y DESPLIEGUES.....	9
8.5	HERRAMIENTAS DE AUDITORÍA PARA GENERAR REPORTE DE LOS PERMISOS Y REVISAR LOS LOGS (REGISTROS) DE LAS APLICACIONES.....	9
8.6	PRODUCTO ESTÁNDAR CON UN SOPORTE PROFESIONAL Y ACTUALIZACIONES FRECUENTES.....	9

1 Objetivo de este documento

El objetivo de este documento es proveer al lector una información útil sobre el diseño y la creación de un sistema de Control de Acceso Basado en Roles (RBAC).

2 Conceptos principales

Un sistema RBAC proporciona tres tipos de características: autenticación, autorización y auditoría:

2.1 Autenticación

Confirma la identidad del usuario: la autenticación consiste en comprobar la identidad del usuario que entra a su aplicación. Este proceso se lleva a cabo en dos pasos: primero, la identificación, donde el usuario declara quién es. El segundo paso consiste en comprobar dicha identificación. Generalmente, este proceso se realiza por medio de cuentas de usuario y contraseñas. Esta etapa es el primer nivel de seguridad.

2.2 Autorización

Las autorizaciones definen lo que un usuario puede hacer en una aplicación: básicamente, usted define lo que el usuario podrá ver, hacer y modificar en la aplicación.

Existen dos métodos para definir las autorizaciones:

- * La forma más segura es prohibir todo desde un principio, para después otorgar los permisos y abrir posibilidades. Sin embargo, utilizando este método, usted corre el riesgo de olvidar definir algún permiso, imposibilitando así el trabajo de un usuario final u otorgando permisos a usuarios no indicados.

- * La forma más rápida es autorizar toda las acciones, para después asignar restricciones y así prohibir algunas de ellas. Esta forma es más rápida que la anterior puesto que generalmente existen menos restricciones que permisos.

La etapa de autorización es el segundo nivel de seguridad y es, en efecto, la parte más pesada del diseño de un sistema RBAC, ya que usted tiene que codificar cada permiso/restricción.

2.3 Auditoría

Conserve un historial y un control de las transacciones sensibles en su aplicación: usted podría necesitar esta información para cumplir ciertas reglas de gestión de su empresa, con requerimientos legales como SOX o para cumplir con procesos de certificación tipo ISO.

La auditoría le permitirá saber quién hizo qué en su aplicación, cuándo lo hizo y quién concedió qué permiso a quién.

3 Componentes claves

El sistema RBAC para aplicaciones corporativas se compone de los siguientes elementos:

3.1 Un repositorio asegurado para almacenar los datos de RBAC

Usted necesita un lugar seguro donde almacenar los datos y las contraseñas de los usuarios, sus roles y sus permisos.

3.2 Un componente integrado en la aplicación

Este componente se comunicará con el repositorio RBAC de modo que la aplicación se ajustará a las autorizaciones de los usuarios.

3.3 Una consola de administración

Esta aplicación está diseñada para el personal no técnico con el objetivo de que éstos puedan gestionar el uso de las cuentas de usuario y conceder permisos. Esta consola se compone de una interfaz amigable que permite el manejo de esta información sin complicación alguna, liberando así al grupo de desarrolladores de esta tarea.

3.4 Documentación para los desarrolladores y los administradores

En cualquier momento, usted puede necesitar documentación para todo el personal que trabaje en el proceso de seguridad de sus aplicaciones. Por ejemplo una guía de integración, un manual del usuario, una FAQ (Preguntas y Respuestas Frecuentes), etc.

4 ¿Por qué los permisos deben ser independientes del código?

4.1 ¿Qué significa esto?

Para que los permisos sean independientes del código de la aplicación, éstos no deben ser definidos dentro del mismo. Por lo tanto, usted necesitará insertar una llamada en el código de su aplicación para activar el sistema RBAC. Cabe resaltar que esta llamada no es un permiso por sí mismo.

Si el código que define los permisos es mezclado con el código de la aplicación, usted puede tener, en el largo plazo, problemas con el mantenimiento de sus aplicaciones.

4.2 Permisos y aplicación: un ciclo de vida incompatible

La administración del control de acceso es requerida frecuentemente (debido a la creación de nuevas cuentas, cambios en los permisos, etc.). Los cambios en una aplicación son, al contrario, no tan frecuentes. De hecho, la versión final de una aplicación puede ser válida por meses. Por lo general, la vida útil de un permiso tiende a ser mucho más corta que la vida útil de la aplicación. Si los permisos son mezclados con el código de la aplicación, los usuarios finales tienen que esperar una nueva versión de la misma para beneficiarse de los nuevos permisos.

Por lo tanto, la definición o la concesión de permisos no deberían apoyarse en las nuevas versiones de la aplicación. Usted debería ser capaz de añadir o conceder un permiso sin cambiar el código de la aplicación (lo cual requiere un ciclo de desarrollo completo: codificación, pruebas, eliminación de bugs, despliegue ...). Si los nuevos permisos son tomados en cuenta de manera dinámica (sin tocar el código pesado en la aplicación), usted será capaz de definir/conceder los permisos mientras su aplicación se encuentra aún en producción. ¡De ser así, ellos podrían ser utilizados inmediatamente!

4.3 Independencia de los administradores

Los administradores locales conocen a cada usuario final y las autorizaciones del mismo (sobre todo en el caso de sitios remotos). Ellos necesitan un instrumento de administración de fácil uso para manejar las cuentas y los permisos de los usuarios sin necesidad de apoyarse en el equipo de desarrollo. Esto sólo es posible si la creación de permisos no implica un cambio en el código de la aplicación.

4.4 Libera al equipo de desarrollo

Si los usuarios finales y los administradores se encarganran de la gestión diaria de las cuentas y permisos de los usuarios, el equipo de desarrollo podría dedicarse sólomente al desarrollo de las aplicaciones.

5 Preguntas clave al diseñar su sistema RBAC:

5.1 Necesita usted ... ¿Asegurar diversas aplicaciones en un repositorio centralizado?

¿Necesita usted centralizar todas sus aplicaciones en un mismo repositorio centralizado? ¿Sus administradores deberían tener una vista general de todas las aplicaciones y de las cuentas usuario de éstas?

5.2 ...¿Gestionar roles compartidos?

Un rol compartido puede ser usado para varias aplicaciones, mientras que un rol específico sólo es usado para una sola. Los roles compartidos son, en particular, útiles si sus empleados realizan varias actividades: así, usted no tiene que gestionar un rol para cada aplicación a la cual un usuario puede tener acceso.

5.3 ...¿Soportar el Single Sign-On (autenticación única)?

Si usted utiliza el Directorio Activo para gestionar sus cuentas de usuario, se acordará que generalmente es necesario proporcionar un proceso de identificación única, de modo que la seguridad pueda ser aplicada tan fácil como posible desde el punto de vista del usuario final: una vez que el usuario comienza una sesión, cualquier otra aplicación podrá ser abierta sin necesidad de volver a identificarse.

5.4 ...¿Soportar diversas tecnologías (Winforms, Webforms, Webservices)?

En un futuro, ¿Necesitará usted que su sistema RBAC soporte otras tecnologías? .Net proporciona una amplia gama de tecnologías de desarrollo: Winform, Webforms, Webservices... Si su sistema toma en cuenta solamente sus necesidades a corto plazo (aplicaciones Winform por ejemplo) usted podría verse en la necesidad de desarrollar otras soluciones para soportar otras tecnologías, y terminar así con varios sistemas RBAC y con costos mayores en desarrollo, mantenimiento y capacitación.

5.5 ...¿Generar reportes sobre las cuentas de usuario y sus permisos?

Usted podría verse en la necesidad tener que proporcionar descripciones detalladas de las cuentas de usuario existentes, así como de los permisos concedidos a cada usuario.

5.6 ...¿Cumplir con requerimientos de auditoría (SOX...)?

Para cumplir con requerimientos legales o de una certificación (Sarbanes-Oxley el Acto, ISO, CMMI, ITIL ...) usted podría necesitar un seguimiento de todas las acciones que los usuarios finales realizaron con la aplicación (¿Quién entró en la aplicación?, ¿Quién abrió un formulario? ¿Quién modificó un registro?). En un futuro, usted podría necesitar este registro para monitorear sus transacciones sensibles (finanzas).

5.7 ...¿Soportar un equipo de desarrolladores grande o internacional?

¿Cuenta con un equipo de desarrollo grande? ¿Su equipo de desarrollo está disperso en diferentes ubicaciones?. Si éste es su caso, coordinar su equipo le requiere un esfuerzo importante. La rotación de personal y el manejo del conocimiento pueden convertirse en un verdadero problema.

5.8 ...¿Tipo de permiso: qué tanto necesita asegurar la aplicación?

Al definir el control de acceso, usted empieza generalmente deshabilitando ciertas opciones del menú. En esta situación usted podría necesitar una gama más amplia de acciones en sus aplicaciones, desde la opción de deshabilitar ciertas opciones del menú hasta filtrar informaciones de acuerdo a los permisos otorgados a los usuarios.

¿Necesita usted hacer alguna de las actividades mencionadas anteriormente? ¿Esconder o deshabilitar campos u opciones del menú, tabs o controles? ¿Filtrar una lista de información diferente dependiendo el usuario final que utiliza la aplicación? ¿Modificar las reglas de negocio?

6 Mantenimiento: los costos subestimados

Cuando escogemos una solución, por lo general sólo nos fijamos en el costo inicial. ¿Cuánto me costaría desarrollar o comprar la solución adecuada?

¡Muy pocas son las personas que se dan cuenta que el costo inicial sólo representa del 10 a 20 % del costo total!

La mayor parte del costo aparece en el largo plazo: es decir, al usar y al mantener la solución.

Forrester Research estima que el 78 % del presupuesto de informática es utilizado en los costos de mantenimiento. ¡Esta afirmación no es solamente verdadera para aplicaciones de gestión, sino también para el sistema RBAC!

Cuando diseñe su sistema, usted deberá prever los gastos mayores que tendrá que pagar en los próximos 10 a 20 años.

6.1 Apoyar a los desarrolladores y a los administradores

Los desarrolladores deberán ser asistidos cuando integren un sistema de seguridad a su aplicación: definir permisos, implementar procesos de autenticación, usar las nuevas versiones del framework.Net... Los administradores también necesitarán ayuda para gestionar las cuentas de usuario, los roles y los permisos.

Por lo tanto, usted necesitará una solución para apoyar a sus desarrolladores y administradores en el largo plazo. En el caso de soluciones internas, usted tendrá que mantener un equipo de soporte y gestionar, al mismo tiempo, la rotación de personal y la transferencia de conocimientos dentro de éste a lo largo de los años.

6.2 Mantener los permisos coherentes con el código

Cuando una aplicación evoluciona, es necesario asegurarse de que los permisos permanezcan compatibles con el código de la aplicación. Por ejemplo, si un control es modificado, usted tiene que comprobar que todas las restricciones relacionadas con este control continúen trabajando de manera correcta. Cada nueva versión de la aplicación necesita una verificación completa de todos sus permisos. Algunos sistemas RBAC proporcionan un proceso de verificación de permisos automatizado, el cual podrá ser utilizado una vez que usted valide su aplicación.

6.3 Desplegar nuevas versiones de la aplicación

Cada nueva versión de una aplicación viene con un nuevo conjunto de permisos. Estos nuevos permisos deberían ser desplegados con la nueva versión de la aplicación. Lamentablemente, usted no puede solamente sustituir el viejo repositorio por el nuevo, puesto que de esta forma todos los datos capturados por el administrador cuando la aplicación estaba en producción (cuentas de usuario, roles y permisos concedidos a estas cuentas) se perderían.

Por ello, usted tendrá que capturar con cuidado los nuevos permisos en el repositorio existente. Esto también implica una importación/ exportación manual, a no ser que su sistema RBAC le proporcione un instrumento de despliegue automatizado.

6.4 Realizar el control de versiones de los datos de seguridad

Mientras una nueva versión de una aplicación está siendo desplegada, algunos usuarios pueden usar la nueva versión mientras que otros pueden estar trabajando todavía en la versión antigua. Tanto el nuevo repositorio de permisos como el ya existente, deben estar disponibles durante el proceso de migración.

Usted deberá, por lo tanto, asegurarse de que las dos versiones permanezcan disponibles y de que cada usuario tenga acceso solamente al repositorio de la versión apropiada.

7 Desarrollo interno, la solución clásica para la seguridad de una aplicación

Las soluciones de seguridad desarrolladas internamente proporcionan algunas características de seguridad sin embargo estas soluciones presentan ciertos problemas, como:

7.1 El desarrollo y el mantenimiento dependen de los recursos internos

Si usted tiene una solución interna, usted tendrá que ser particularmente cuidadoso con su manejo. De hecho, usted no puede permitirse perder el conocimiento de la misma mediante la rotación de personal y así quedarse sin ningún miembro en su equipo capaz de gestionar la seguridad de su aplicación. De hecho, lo ideal es que usted mantenga un equipo que pueda manejar cualquier pregunta, cambio o crisis que llegue a perturbar su sistema de seguridad.

7.2 Cuentas de usuario específicas a la aplicación

Implementar un sistema de autenticación única en una aplicación .NET es complicado. Esta acción implica conectarse con el Directorio Activo y así utilizar las cuentas de Windows para identificar a los usuarios. Por consiguiente, la mayor parte de las aplicaciones utilizan cuentas de usuario creadas específicamente para dicha aplicación y son almacenadas en la base de datos de la misma. Esta solución es probablemente la más fácil de desarrollar, sin embargo, ella es también la más difícil de mantener: las cuentas de usuario necesitan ser gestionadas por el equipo de desarrollo mientras que las cuentas del Directorio Activo/Windows ya existen y son gestionadas por administradores de sistema.

7.3 Acceso y permiso: baja granularidad

Adaptar una solución desarrollada internamente a todas las exigencias de un negocio es, a menudo, una actividad complicada. Los gerentes tienden a tomar en cuenta solamente la funcionalidad de una solución, dejando a un lado las implicaciones técnicas de la misma. ¡Hacer que las opciones de un menú estén disponibles es una cosa, lograr que ellas trabajen es otra totalmente diferente!

7.4 Código específico en la aplicación

Generalmente, las soluciones hechas en casa codifican los permisos dentro de la aplicación de manera directa. Por lo tanto, los desarrolladores terminan con un sistema de seguridad complejo, costoso de mantener y difícil de actualizar. Cuando los datos de seguridad necesitan ser capturados o actualizados, el desarrollador debe referirse al código, encontrar la información anterior, cambiar el código, para después volver a probarlo y desplegar la aplicación nuevamente. Todo este proceso carece de reactividad y de flexibilidad.

Además, de manera general la gente se da cuenta de sus necesidades de seguridad mucho después de la fase de diseño. A menudo, es muy difícil o incluso imposible cumplir con esta exigencia de seguridad durante la fase de desarrollo. Generalmente, los permisos complejos son identificados cuando la aplicación está en producción, requiriendo una modificación inmediata dentro de la aplicación.

7.5 Una solución separada para cada aplicación: problemas de mantenimiento

En teoría, usted debe ser capaz de tener una visión clara de lo que una empresa llegará a ser dentro de los próximos 5 a 10 años, sin embargo, en la vida real esto es más que imposible. Es probable que en un futuro usted necesite desarrollar una nueva aplicación basada en una tecnología no soportada por su sistema de seguridad antiguo. Por lo tanto, probablemente usted acabaría creando un nuevo sistema de seguridad para esta nueva aplicación. Eventualmente, usted tendría un sistema de seguridad para cada aplicación o, dicho de otra manera, tendría enormes costos de mantenimiento.

8 Visual Guard, una solución corporativa para la seguridad de sus aplicaciones

Novalys ha estado desarrollando soluciones de autenticación y de permisos para aplicaciones corporativas durante más de 15 años. Basados en esta experiencia, nosotros hemos diseñado Visual Guard, una solución dirigida a asegurar las aplicaciones .Net, no importando lo complejo del entorno.

8.1 Una sola solución para asegurar y centralizar todas sus aplicaciones .Net

Visual Guard le permite asegurar cualquier tipo de aplicación Winform, Webform y Webservices dentro de una misma herramienta. Usted puede centralizar la gestión de todas sus aplicaciones con sus respectivas cuentas de usuario en un único repositorio asegurado. Visual Guard lo provee de un sistema de autenticación único y listo para usar, el cual se apoya en las cuentas de Windows.

Para simplificar aún más la gestión de los permisos, usted puede crear roles compartidos y autorizaciones para los usuarios finales que se apliquen a varias aplicaciones.

8.2 Permisos independientes del código

Con Visual Guard, usted no necesita escribir código dentro de su aplicación para definir los permisos. Esto significa que usted puede definir e implementar los permisos sin tener que pasar por todo el ciclo de codificación, pruebas, despliegue y retroalimentación. Usted puede definir permisos en cualquier momento, incluso cuando la aplicación está aún en producción. ¡Los permisos toman efecto inmediatamente!

8.3 Herramientas de administración diseñadas para personas no técnicas

La consola de Visual Guard facilita la gestión cotidiana de la seguridad para el personal no técnico. La gestión de las cuentas de usuario, los roles y los permisos de los mismos no requiere, en efecto, de una habilidad técnica en específico. Por lo tanto, cuando usted elija la persona que estará a cargo de la seguridad, usted no estará suspendido a sus habilidades técnicas, optimizando así su proceso de negocio y haciéndolo más acertivo: sus administradores, sus jefes de sistemas así como sus gerentes en sitios remotos, pueden encargarse de la seguridad. Para una flexibilidad aún mayor, usted puede definir varios niveles de privilegios para los administradores. Usted puede crear también su propio formato de administrador - e integrarlo en su aplicación – y así llamar al API de Visual Guard para administrar cuentas de usuario y roles.

8.4 Desarrollar herramientas para gestionar permisos, versiones y despliegues

Visual Guard ofrece una amplia gama de herramientas y asistentes para desarrolladores: usted puede crear permisos en unos cuantos clics, verificar la coherencia entre las aplicaciones y los permisos de manera automática, desplegar nuevos permisos en repositorios ya existentes sin alterar la seguridad de los datos en la producción, gestionar varias versiones de un repositorio cuando la aplicación se despliegue, etc

8.5 Herramientas de auditoría para generar reportes de los permisos y revisar los logs (registros) de las aplicaciones

Los auditores tienen un acceso de sólo lectura en sus aplicaciones, para explorar los permisos existentes, los roles y las cuentas de usuario y pueden también generar informes detallados acerca de los mismos. Ellos pueden revisar los registros de aplicación para supervisar operaciones específicas, así como también, verificar los registros de la consola de administración para inspeccionar quién dió qué autorizaciones a quién.

8.6 Producto estándar con un soporte profesional y actualizaciones frecuentes

¿Por qué re-inventar la rueda? Visual Guard le proporciona funcionalidades listas para usar en la seguridad de sus aplicaciones. Las mismas han sido desarrolladas durante los últimos 15 años para así llegar a ser una solución estándar para las empresas. Implementar Visual Guard es fácil y rápido, de la misma manera que lo es su curva de aprendizaje. Novalys proporciona un apoyo internacional: usted podrá enfocarse a su actividad principal y dejarnos solucionar la seguridad de sus aplicaciones. Novalys está también comprometido en lograr un mejor uso de todas las continuas innovaciones de Microsoft en el ambiente .Net, Microsoft OS, etc.