Document number	Revision
DOCU12214	1

Application security

Architectural considerations

The goal of the architecture behind the security is the following:

Goals	Description
Simplicity	Security must be reasonable simple and easy to understand to be used the correct way and thus secure.
Performance	Performance is an issue in most complex systems. The goal is to provide a system with very high performance.
Agility	The system must be easy to adapt to business requirements.
Consistency	The same security must apply in both search and view

Authentication

Microsoft Windows server and IIS is managing authentication of requests. Authentication must be enabled on website for security and traceability, however in some circumstances it may be required that anonymous users are able to read certain information, in this case the IIS account used for anonymous authentication must be granted access to relevant web pages, basetypes

and workspaces.

Authorization

Once the user is authenticated as described above the authorization is performed as described below.

Web page

According to IIS authentication method and NTFS access rules on web site application directory/files.

Imported Ldap Roles

A subset of groups and users from Ldap is replicated to TurboStage. The parameter LdapGroups (semicolon separated list) specifies which groups to replicate. All the groups specified in this parameter will be copied to TurboStage as well as all contained sub groups and users contained in all groups and sub groups.

BaseType

Access to Types are controlled by BaseType type security columns TrustRead, TrustChange and TrustCreate which sets permission to type through Ldap roles. Use this link to monitor an set BaseType security.

Only users with TrustRead for a **BaseType** will be able to see menues for the **BaseType**.

The security columns are cached by default to maximize performance. Cache refresh time is set by parameter CacheTime which has the default value 600 seconds (10 minutes).

Workspace

According to Manager, TeamMembers and Trustees in entity referred by object data's entity column

VectorRoles

Several VectorRoles are provided. The following subset defines user levels:

Level	Description
User	Basic user relieved from all disturbance like advanced menus, sections, buttons, text etc. The default level for new users.
AdvancedUser	Provides access to advanced menus, sections, buttons, text etc.
SuperUser	Full access to support of business processes. Superuser does not have any increase in workspace transparency or restricted data. AdminRead has the same file-system permissions as User. Only users with *IsSuperUser* status is set to '1' will be able to switch to SuperUser user level. By default only an Admin will be able to set this status.
AdminRead	AdminRead will be able to read all web page data. AdminRead however has the same file-system permissions as User. Only users listed in parameter *AdminReadMembers* will be able to switch to AdminRead user level.
AdminWrite	AdminWrite is a superset of AdminRead permissions. AdminWrite will be able to read all web page data. AdminWrite however has the same file-system permissions as User. The parameter AdminWriteAuthentication specifies if user is required to provide logon credentials to switch to AdminRead/AdminWrite user levels. Only users listed in parameter AdminWriteMembers will be able to switch to AdminWrite user level.

The actual user level is chosen by the user and is non-volatile i.e.. does not change at reset or server reboot.

Refer to **VectorRoles** schema element for more information on how to create customized vector roles.

Some VectorRoles are highly dependent on actual data contained in an object and is set on per object basis. The VectorRoles in turn gives access to read and write columns. This is described in Detailed object level authorization.

Publishing objects on web pages and public distribution lists

In some cases it may be desirable to only publish certain objects on web pages for certain users. In other cases it may be desirable to notify by email on changes to objects and maybe send information including document files to these recipients.

Example on this is employee information and document distribution lists. The type attribute xxxxxxxxx defines a column that contains list of Ldap roles to be used for publishing and distribution.

External distribution lists

In some cases the distribution list may contain non users like external business partners that are not defined as system users. Also user groups (non security groups) that may or may not contain system users that may not be validated.

Detailed object level authorization

If all of the above is resolved positive the user is able to see objects (data) of specific basetype. However depending of basetype/subtype setup the actual data itself may grant (or deny) the user detailed read and write permission on columns.

The object level authorization is based on actual object data. Depending on which user that access what object the user may be granted permission to certain columns. For complex types containing a process the authorization may depend on the process step of the object. If the object is displayed as a reference then the authorization to reference columns will depend on the parent object. The data contained in the actual reference like BOM data (Qty, Pos etc.) may also depend on the process step of the parent object and the resources (roles) assigned to the parent object.

Column authorization

Column permissions are set according to read and write attributes of type and column elements in conjunction with resolves user vector roles.

If no read/write attributes have been specified at column the read/write permissions will be inherited from the type and if a process has been defined for the type the step read/write attributes for the step. The type attribute will be be logically OR'ed with the possible step attributes. However if the column defines the attributes then this vector will overrule all, the type and process attributes will be ignored.

Resource column

The type attribute *resourcecolumn* defines one or more data columns that contain LDAP roles.

If a user is a member of these roles then the role vector bit Resource will be set. Columns that include the Resource vector role in read/write attribute may be accessed by that user. Also columns that do not explicitly define write attribute may inherit from type or process step the permission for read/write.

Process based authorization

A process defines a number of process steps. Each step may define the **resourcecolumn** attribute. If the user belongs to any of the LDAP roles in the resourcecolumn then the corresponding resource vector bit is set, if the process currently is in this step then the corresponding activeresource vector bit will be set.

VectorRoles listed

../../diagnostics/vectorroles.aspx

How is SQL injection attacks prevented?

SQL server attacks are prevented by using SQL parameters for updating database and by surrounding all input from users with ‘ characters.

Highstage