Security Relationships

Last Modified on 07/12/2020 4:44 pm AEDT

Overview

The ReadiNow platform lets you define access control policy that takes advantage of the relationships between individual records. Ordinarily this is achieved by using metadata based security to define a report that follows some path of relationships between individual records and the users that are trying to access them.

However, it is sometimes convenient to define, for a particular type of relationship that any permission users have on one record should automatically apply to its related record as well.

For example, a "Recovery Plan" object may relate to a "Plan Step" object such that each Recovery Plan record contains several Plan Step records. There may be various access rules that describe who can view or modify a "Recovery Plan", but a desired policy might be that if a user can access a recovery plan, for whatever reason, they should also be able to access the plan step records for that plan.

A relationship configured in this way is called a *security relationship*. In this case, Recovery Plan is acting as the *granting object*, and Plan Step is acting as the *secured object*.

This option is a property of the relationship, and it cannot be configured on a per-role basis.

Types of Security Relationships

A relationship can be configured so that a User who has access to Recovery Plan (but not necessarily the Plan Steps):

- does not gain any access to Plan Steps → Off
- gains ability to see the ***Name*** of any related Plan Steps → Revealing Name Only
- gain ***Read Access*** to view any related Plan Steps → Propagating View Permission
- gains the ***Same Access*** for Plan Steps that they have for Recovery Plan → Propagating All Permission

Note: it may be the case that a User has access to the secured record via some other access rule(s).

Caution: Poorly configured 'propagate view' and 'propagate all' relationships can significantly increase the number of records that need to be checked when calculating access permissions, which can *significant* impact response times across the application. Refer to best practices below.

Revealing Name Only

If relationship is configured as a 'name only' security relationship, and the user has permission to view some *granting* record, then the user will also automatically have permission to view the name of the related record.

For example, consider a "Recovery Plan last reviewed by Employee" lookup relationship between the Recovery Plan and Employee objects. This can be configured as: "If I have access to Recovery Plan then: I can see the **name** of any related Employee."

Tip: Enabling 'name only' security relationships can boost response times in cases where reports only require the name, or a count, of related records, and where there are other access control rules being applied to the secured object.

Propagate View Permission

If a relationship is configured to propagate view permission only, and if the user has permission to view the granting record, then the user will receive view permissions to see the secured record. Permission to modify and delete are not automatically propagated.

For example, consider a relationship "Recovery Plan was used for Recovery Events" in which each recovery plan has related records that represent historical recovery events. It may be required that a user should always be able to see these historical recovery events, but that they should not necessarily be able to modify them. This can be configured as: "If I have access to Recovery Plan then: I have **read access** to any related Recovery Event."

Propagate All Permission

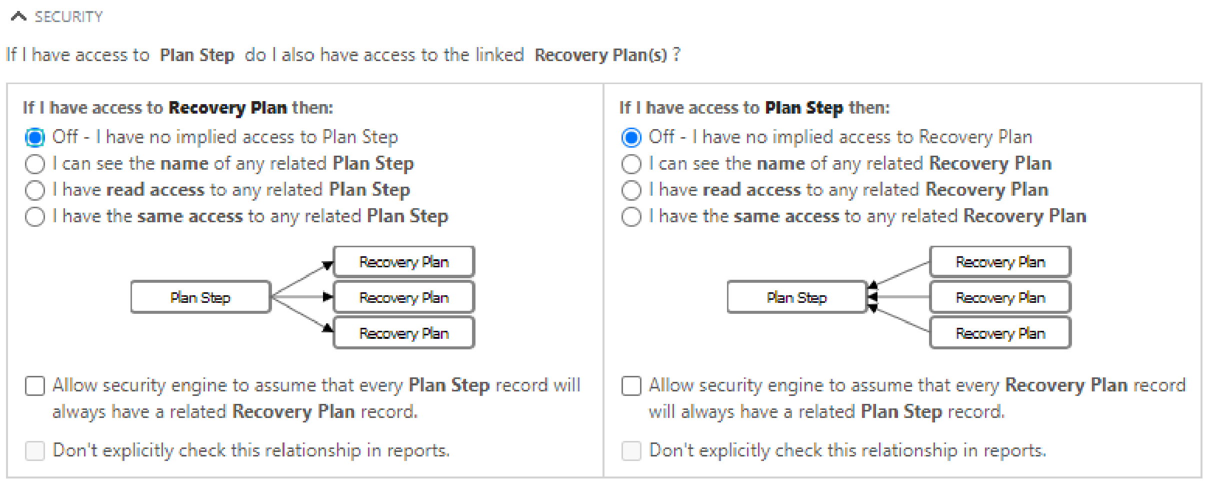
If a relationship is configured to propagate all permissions, then whatever permissions the current user has to view, modify, and delete granting record will also be received by the secured record.

For example, consider a relationship "Recovery Plan has Plan Steps". If the user can view a plan, then they may also view any related Plan Step records. If they can modify the plan, then they may also modify any related Plan Step records. This can be configured as: "If I have access to Recovery Plan then: I have the **same access** to any related Plan Step."

Configuring Security Relationships

To configure a relationship to be a security relationship:

1. Open the Relationship Properties for the relationship or Lookup.
    1. Open Form Builder mode, see Opening and Using Form Builder
    2. Navigate to the field you want.
    3. Hover on relationship or lookup. The control icons appear.
    4. Select Configure icon. The Relationship Properties dialog displays.
2. Expand the Security tab
3. Options such as the following are presented:


SECURITY

If I have access to **Plan Step** do I also have access to the linked **Recovery Plan(s)** ?

If I have access to Recovery Plan then:

- [✓ Off - I have no implied access to Plan Step]
- I can see the **name** of any related **Plan Step**
- I have **read access** to any related **Plan Step**
- I have the **same access** to any related **Plan Step**

☐ Allow security engine to assume that every **Plan Step** record will always have a related **Recovery Plan** record.

☐ Don't explicitly check this relationship in reports.

If I have access to Plan Step then:

- [✓ Off - I have no implied access to Recovery Plan]
- I can see the **name** of any related **Recovery Plan**
- I have **read access** to any related **Recovery Plan**
- I have the **same access** to any related **Recovery Plan**

☐ Allow security engine to assume that every **Recovery Plan** record will always have a related **Plan Step** record.

☐ Don't explicitly check this relationship in reports.

4. Select changes as required.
5. Select **OK** to close the window.
6. Select **Save** to save the form, including the new relationship settings.

Cascading Access

If a securing relationship is configured to use the propagate view or all permissions then these permissions will continue to cascade across other adjoining securing relationships.

For example, in "Recovery Plan has Plan Steps" example, there could be an additional "Plan Steps have Step Revision History" relationship. If both relationships are configured to propagate view or all permissions, and if a user has permission to read a recovery plan record, then they will also receive read permission for the plan step records, and the Step Revision History records related to those plan step records.

Recursive Relationships

A relationship that relates from some object back to the same object is a recursive relationship and forms a hierarchy of records.

If a security relationship is recursive, then access control will cascade along the recursive relationship.

Recursive security relationships are permitted but should only be used to propagate permissions from a parent record to a child record. For example, it's OK to configure a recursive "Region contains Region" relationship such that someone with permission to view an 'Australia' record will automatically be able to see the 'Sydney' record.

Tip: At present, the security engine will generally process relationships recursively in access rule reports more efficiently than recursive security relationships.

Caution: Access control configurations that involve checking a large number of descendent records in a hierarchy just in case one of them might cause an ancestor node to become visible can significantly impact response times.

Caution: Take extra care with recursive relationships to ensure that security is being granted in the intended direction, as it can be easy to accidentally select the wrong direction. Verify that records you intend to be visible are visible, and that records you intend to be hidden are hidden.

Bidirectional Security Relationships

It is possible to configure a relationship to be a security relationship in both directions. However, if this is being done then it is recommended that at least one of the directions be set to 'name only'.

Configuring a relationship to propagate view (or all) permissions in both directions is not recommended and can lead to unexpected outcomes. For example, consider a relationship between "Employee is in Department" that has been incorrectly configured to propagate view in both directions. A user who is able to view an employee record will then receive view access to their department, and then this will cascade to other employees in the same department. Platform support for this may be discontinued in future versions.

This effect can be compounded considerably if the relationship is a many-to-many relationship. Consider an "Employees invited to Meetings" relationship: the access will cascade from an employee record, to all of their meetings, to other employees in those meetings, to their meetings, and so on.

Caution: A relationship that is configured to propagate view or all permissions in both directions can result in permission being granted to more records than expected. In the case of many-to-many relationship, the effect can be considerable.

'Assume Always Related' optimiser option

The relationship properties page has a checkbox option:

"Allow security engine to assume that every Object1 record will always have a related Object2 record."

The ReadiNow platform allows for lookups to be marked as mandatory on forms but does not presently support enforcing this at an object level. Enable this option if, as an app developer or tenant administrator, you know that this relationship is effectively mandatory and will be set on every record.

The ReadiNow report security engine is able to take advantage of this knowledge to perform various optimisations.

For example "Allow security engine to assume that every Employee record will always have a related Department record."

Don't Explicitly Check Relationship In Reports

This option offers a performance optimisation that applies when propagating 'Read Only' and 'All' permissions. The optimisation works by restricting which records are returned to a Report.

Consider 2 Security Access Rules that apply to a User:

- full access to Recovery Plan Records whose Name starts with 'X1'
- full access to Recovery Step Records whose Name ends with '00'

Note: *not all Recovery Plan Names start with 'X1' and not all Plan Steps Names end with '00'*

In this example there is a securing relationship which grants access as follows:

- Recovery Plans starting with 'X1' grant access to any related Plan Steps
- Plan Steps ending in '00' grant access to any related Recovery Plans

If this option is unchecked (default) the ReadiNow platform will fully exhaust all opportunities to return Records that a User is implicitly granted access to. This means a User will likely see Recovery Plans that start with something other than 'X1' because a Plan Step ending '00' was found, and permissions were propagated from 'Plan Step' to 'Recovery Plan'.

If this option is checked the ReadiNow platform will only evaluate securing relationships where the User has explicit access to the Recovery Plan Records. In other words securing relationships that can only ever return Recovery Plan Records via an access grant will not be evaluated. This typically results in fewer Records being returned and can have a significant performance boost.

When to use this option

An ideal situation for using this option is, for example, a general purpose Report of Tasks which can rely on an access control rule such as 'Tasks Assigned to Me'.

In this example consider a system where objects: Risk, Incident, and Remediation all relate to Task. For each of these Objects, if you can access the Record, then you want to have to access to the related tasks. Without this option the Report would need to evaluate the access rules for every security relationship.

Instead, for Reports where it is not necessary for every document to be shown, using this option will significantly improve performance.

Additional Details for Name Only relationships

Name-only security relationships have been introduced to:

- make access control configuration easier by reducing the number of access rules that need to be created for trivial 'name' scenarios.
- improve report performance - the ReadiNow engines are able to take significant advantage of this setting.

Their use is encouraged; and they may be enabled by default for all lookups in a future version of the ReadiNow platform.

Advanced users may wish to be aware of the following behaviors of name-only relationships. They are all consequent of the above two objectives.

Name-Only Lookups on Forms

If a Lookup control for this relationship was placed on a form, and the current user happened to have access to the related employee record because of some other access rule, then the Lookup will show the name as a link and the user can click through to see the related employee record.

However, if the user did not otherwise have permission to view the related employee record, then the form will simply show the name of the employee without it being linked.

Similarly, if a Lookup control has been used to show a to-many relationship as an inline comma separated list, then it is possible that the user may have name-only access to some records, and view access to other records. In this scenario, the records that can be viewed will be shown as links, and the records that cannot be viewed will be shown as text. These may appear intermixed in the same list in accordance with the security configuration.

Name-Only Relationships on Forms

If name-only is enabled for a to-many relationship, and if there are no other access rules that grant permission to view the related records, then the relationship report on a form will only show values for the name field. Other fields will appear as blank in report columns, and as null in calculations.

Name-Only relationships and record-existence

Ordinarily, the ReadiNow access control system causes all features and calculations to treat non-visible records as though they don't exist at all. This is done to ensure that users can't use features such as 'summarize' or 'is defined' to infer information about records that they cannot directly view.

A name-only security relationship causes a user to become aware of the *existence* of related records that they don't otherwise have permission to see.

This means that the following features now act as though they can detect that the related record exists:

- The 'is defined' and 'is not defined' report analyser conditions
- Report summarize as 'count'. Other summarize operations such as max and min that rely on field values will not be able to see those fields.
- Report 'Show Totals' options that relate to count.
- The count calculation function. For example: count([Name of the security relationship])
- The **is null** and **is not null** calculation keywords

Name-Only allows the record's object to be accessed

In addition to a record's name, a name-only securing relationship also allows a report or calculation to access a record's type without requiring or checking for view access to the record. All other fields, relationships, and lookups, behave as though they are not set.

Name-Only only shows record names when the security relationship is being followed

The name-only security relationships only make record names visible in reports, forms and calculations where the security relationships is actually being followed.

That is to say, name-only security relationships are not consulted in other cases where a record is being accessed. For example, consider a "Recovery Plan has Reviewer" name-only securing lookup that allows the review's name to be seen if the user has view permission on a plan. The user does not automatically get to see the review's name in other places such as Person picker reports solely on the basis of this name-only relationship. The user will require normal view access to the Person record to see the Person in a picker or Person report.

Using Name-Only for Performance benefits

The name-only security relationship will give a performance benefit when a report or calculation:

- follows the relationship or lookup in the direction towards the secured (name only) record
- and there are multiple or complex access control rules that would otherwise apply to the secured record
- and the only interactions with the record involve:
  - accessing the record's name
  - or its object type
  - or checking for its existence

The improvement in response times can be considerable for some security configurations.

If the report or calculation accesses other fields or relationships on the record, then a full access control check would typically still be performed on the record.

Best Practices for Security Relationships

Security relationships are a double-edged sword. Best practices can be summarised as follows:

Security relationships can significantly simplify configuration and improve response times.

- DO: use name-only security relationships to ensure related lookup names are visible.
- DO: use propagating (view or all) security relationships to make 'sub-component' records visible with their parent records.
- DO: use the Report Diagnostics tool (run as a 'typical' user) to identify potential problems.

Poorly configured security relationships can significantly impact response times and complicate access control configuration.

- DON'T: configure security such that a record access check may need to consult a potentially large number of related records.
- DON'T: make networks of propagating security relationships that connect back to the same object.
- DON'T: propagate security on a relationship unless it makes sense for every user/role.

The same, in more detail means:

1. Name-only security relationships:
   - Can generally always be used with negligible risk of accidental over exposure or response times.
   - Are usually the right choice for lookups to objects that have any access control policy.
2. Securing relationships that propagate 'view' and propagate 'all' permissions are fairly similar:
   - They provide identical access for view/read operations, which is to say most operations.

  - They have identical performance considerations
  - They only differ when performing security checks for modifications.
3. Propagating security relationships are a good choice when:
   - The relationship represents some sort of parent-record to subcomponent relationship.
   - So long as the permissions are being propagated from the parent to the child.
   - They can aid performance, even more than name-only relationships, when used in this manner.
   - For similar reasons, relationships that are Full Ownership or Partial Ownership may be good candidates.
4. Propagating security relationships will generally cascade efficiently along lookups
   - For example, consider two lookups: "Work Item is for Project" and "Project belongs to Department"
   - These relationships can be configured as "If I have access to Department: then I have read access to any related Project" and "If I have access to Project: then I have read access to any related Work Item" respectively. This configuration will run fairly efficiently as the platform only has to step from a Work Item to its single Project record, to its single Department record, and then evaluate any access rules on the department.
5. Avoid scenarios where multiple related records need to be checked in order to determine if some record is visible
   - For example, consider a relationship such as "Project has Tasks"
   - If this relationship were configured using "If I have access to Tasks: then I have read access to any related Project" then multiple access rules will need to be calculated for a potentially very large number of tasks for each and every record in a report, which will impact response times.
6. Consider how many indirectly related records might need to be checked while calculating access control
   - If a large hierarchy of related records need to be checked, this may impact response times.
   - If a securing relationship propagate in both directions, this may impact response times.
   - If a securing relationship is recursive, and many related child records need to be checked, this will impact response times.
7. Avoid combinations of securing relationships that circle back to the same object to form a cycle
   - When the platform detects a cycle in security relationships, it enters a special processing mode that can handle more complex scenarios but is slower to process access control.
   - For example: Object1 records provide access to Object2 record according to one relationship type; and Object2 (or an object that inherits from it) records provide access to Object3 records; and Object3 records (or an object that inherits from it) provide access to Object1 records.
   - There is also a fixed internal limit for the maximum number of records that may be checked in this way per report run.
8. Avoid combinations of securing relationships where a secured object might receive access along more than one direct securing relationship
   - For example, a Plan Step is visible because its Recovery Plan is visible *or* because its Author is visible.
   - The platform is designed to handle this scenario, however it can lead to inconsistent response times in complex tenants.

9. Don't use propagating security relationship relationships unless it is always appropriate for every possible role.
    - The propagation always applies to every role and cannot be conditionally applied or revoked for only certain roles.
    - For example: consider a "Recovery Plan has Plan Steps" relationship. Some users may have permission to no plans, some may have access to some plans, and some may have access to all plans. It's OK to propagate permission from the Recovery Plan to the Plan Steps so long as that's appropriate for every user for whichever plans they can see. However, if there exists even a single role that should have access to the Recovery Plans without having access to the Plan steps, then security relationships should not be used. Define individual access rules to the Plan Steps instead.
10. Use the Report Diagnostics and Tenant Health Check tools to help identify problematic configurations.
    - Always run the report diagnostics as a user that is representative of the type of users who are experiencing problems. This ensures that the diagnostics tool considers security configuration that is applicable to that user.
    - Refer to the Active Security Relationships section of the report diagnostics tool.
    - Any warnings in this section should, in the first instance, be treated as likely or possible causes for the slow reports.