White Paper

# Data Segregation

## Getting Started with Remedyforce Series

Hugo R. Gracia

25 March 2015

bmc

## Welcome to the "Getting Started with BMC Remedyforce" Series

Today's IT departments must drive business growth and innovation, while coping with less resources and increasing complexity. To do this, they require an IT Service Management solution that provides best practices while minimizing costs. BMC Remedyforce is built on Salesforce—the world's most widely used cloud platform—to deliver complete IT service management functionality with the secure social, mobile, and collaborative capabilities users expect.

With the "Getting Started with Remedyforce" white paper series, our aim is to help you leverage BMC Remedyforce to improve the effectiveness and efficiency of your ITSM operations. Each paper addresses a specific area of interest and provides you with conceptual, functional and technical best practices to make configuration decisions and take action to gain value from your BMC Remedyforce investment.

# Data Segregation

The Salesforce platform provides a sharing model that is an essential element to your organization's ability to provide secure application data access. It is crucial to determine what level of data exclusion your organization will need in order to fully develop a solution that meets today's needs but will also scale with tomorrow's needs.

This white paper is intended for system administrators with a working knowledge of the Salesforce sharing and security model and will discuss the Salesforce sharing model as it pertains to the common scenarios in Remedyforce. Although the sharing model extends beyond the topics covered in this document, you will still be able to fully utilize this documentation for the purposes of Remedyforce.

The topics not covered in this paper are: Data Accessibility Architecture: Accounts, Folder Access, Content Access, Chatter Access, Knowledge Base Access, Ideas, Questions/Answers Access, Salesforce2Salesforce, and mobile data accessibility. You can get additional information on those topics at wiki.remedyforce.com and in our BMC Remedyforce Community.

## Securing Data Using Data Segregation

Record-level security lets you give users access to some object records, but not others. As with most applications, data access begins with a user. The application needs to know who you are before it provides access. For Remedyforce, there are different types of users and, in some cases, the level of access is different by type. The most common types of users are Clients (End Users), Staff, and Administrators. Instead of reviewing every attribute of every license type, we're going to focus on those interesting attributes that have significant impact on data access. ***Record ownership*** and ***full access*** are synonymous and interchangeable concepts, and provide the user with the highest level of access to a record. Also, for the remainder of this document, we are assuming a Salesforce user type utilizing a full sharing model.

# Components

## Profiles and Permission Sets

Profiles and permission sets provide object-level security by determining what types of data users see, and whether they can edit, create, or delete records. For each object, the "View All" and "Modify All" permissions ignore sharing rules and settings, allowing administrators to quickly grant access to records associated with a given object across the organization. These permissions are often preferable alternatives to the "View All Data" and "Modify All Data" administrative permissions. Determining the level of access across your end users and staff is a critical step to properly design the security architecture of your Remedyforce instance. Profiles and permission sets also control field-level security, which determines the fields within every object that users can access. For example, an object may have 20 fields, but field-level security can be set up to prevent the users from seeing five of the 20 fields. A practical example usually comes up when Human Resource sensitive information may appear in the tickets; these fields can be set so that only personnel with the corresponding profile can see this information whereas everyone else will not know the field is present.

## Record Ownership and Queues

Every record must be owned by a single user or a queue. The owner has full access to the record. Users higher in a hierarchy inherit the same data access as their subordinates for standard objects. Managers gain the same level of access as their subordinates. If the subordinate has read-only access, so will the manager. This access applies to records owned by users, as well as records shared with them. Queues help your teams manage tickets such as incidents, tasks, change requests, problems, etc. Once records are placed in a queue manually or through a workflow rule, records remain there until they're assigned to a user or taken by one of the queue members. Any queue member (or users above them in the role hierarchy) can take ownership of records in a queue.

If a single user owns more than 10,000 records, as a best practice:

- The user record of the owner should not hold a role in the role hierarchy.
- If the owner's user record must hold a role, the role should be at the top of the hierarchy in its own branch of the role hierarchy.

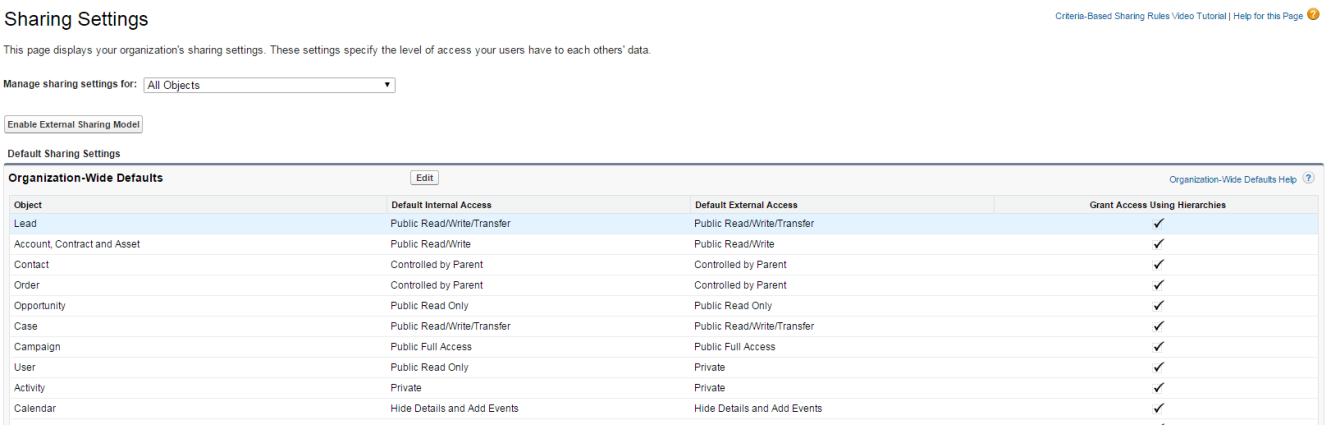
## Organization-Wide Defaults

Organization-wide sharing settings specify the default level of access users have to each other's records. You use organization-wide sharing settings to lock down your data to the most restrictive level, and then use the other record-level security and sharing tools to selectively give access to other users. For example, let's say users have object-level permissions to read and edit opportunities, and the organization-wide sharing setting is Read-Only. By default, those users can read all opportunity records, but can't edit any unless they own the record or are granted additional permissions.

Organization-wide defaults are the only way to restrict user access to a record. Organization-wide default settings can be changed from one setting to another (private, to controlled by parent, then back to private); however, these changes require sharing recalculation and depending on volume could result in very long processing times. For custom objects only, use the Grant Access Using Hierarchies setting, which if unchecked (default is checked), prevents managers from inheriting access. This setting is found in the organization-wide default settings.

Organization-Wide Defaults for objects can be viewed by following these steps:

- Go to Setup > Administer > Security Controls > Sharing Settings



Sharing Settings

Criteria-Based Sharing Rules Video Tutorial | Help for this Page

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data.

Manage sharing settings for: All Objects

Enable External Sharing Model

Default Sharing Settings

Organization-Wide Defaults — Edit — Organization-Wide Defaults Help

| Object | Default Internal Access | Default External Access | Grant Access Using Hierarchies |
|---|---|---|---|
| Lead | Public Read/Write/Transfer | Public Read/Write/Transfer | ✓ |
| Account, Contract and Asset | Public Read/Write | Public Read/Write | ✓ |
| Contact | Controlled by Parent | Controlled by Parent | ✓ |
| Order | Controlled by Parent | Controlled by Parent | ✓ |
| Opportunity | Public Read Only | Public Read Only | ✓ |
| Case | Public Read/Write/Transfer | Public Read/Write/Transfer | ✓ |
| Campaign | Public Full Access | Public Full Access | ✓ |
| User | Public Read Only | Private | ✓ |
| Activity | Private | Private | ✓ |
| Calendar | Hide Details and Add Events | Hide Details and Add Events | ✓ |

Figure 1 - Sharing Settings - All Objects

Accessing individual object settings is as simple as selecting the object from the drop down list. Figure 2 shows the settings for the incident object:

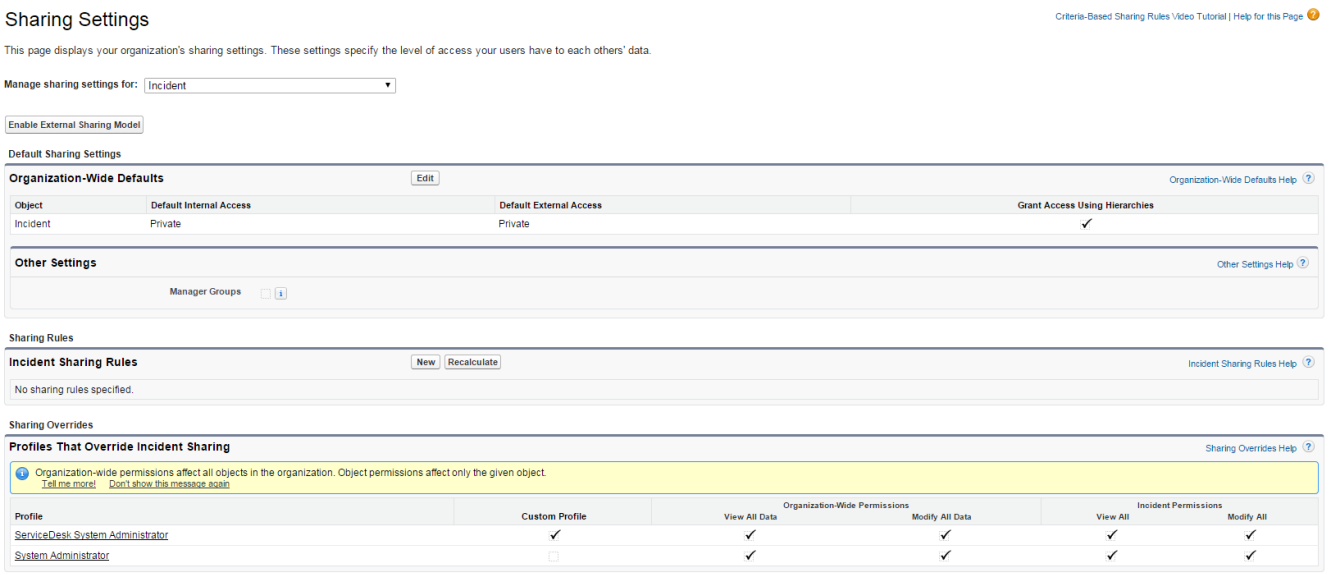



Figure 2 - Sharing Settings - Incident Object

## Role Hierarchy

A role hierarchy represents a level of data access that a user or group of users needs. The role hierarchy ensures that managers always have access to the same data as their employees, regardless of the organization-wide default settings. Role hierarchies don't have to match your organization chart exactly. Instead, each role in the hierarchy should represent a level of data access that a user or group of users needs. An organization is allowed 500 roles; however, this number can be increased by Salesforce.

As a best practice:

- Keep the number of non-portal roles to 25,000.
- Keep the number of portal roles to 100,000.
- Keep the role hierarchy to no more than 10 levels of branches in the hierarchy.

There are a few other things to take into consideration when working with roles:

- When a user's role changes, any relevant sharing rules are evaluated to correct access as necessary.
- Peers within the same role don't guarantee them access to each other's data.
- Modeling the role hierarchy begins with understanding how the organization is structured.

Modeling the role hierarchy usually starts from understanding a manager's scope, starting from the top. The CEO oversees the entire company. The CEO usually has direct reports that can then be segmented by Business Unit (Sales or Support) or geographical region (EMEA, APAC). That person then has direct reports that could be further segmented, and so on. Although this sounds very much like an HR organizational chart, and we have said they might be very much alike, keep in mind, when modeling data access, you need to focus on data accessibility with a consideration to HR reporting.

Overlays are always the tricky part of the hierarchy. If they're in their own branch, they'll require either sharing rules, teams, or territory management to gain needed access. If they are folded into the hierarchy, there might be reporting implications. It's important to spend the time setting up the role hierarchy because it's the foundation for the entire sharing.

*NOTE: Whenever Remedyforce is being brought into an existing Salesforce organization, it is imperative that a thorough review of the existing role hierarchy take place to identify any potential issues or gaps with the data sharing model.*

Creating and editing roles can be done by accessing them through these steps:

1. Go to Remedyforce Administration > Manage Users > Roles+


Figure 3 - Role Hierarchy Sample

## Public Groups

A public group (not Chatter group) is a collection of individual users, roles, territories, and so on, that all have a function in common.

Public groups can consist of:

- Users
- Roles
- Roles and Subordinates
- Other public groups (nesting)

Groups can be nested (Group A nested into Group B), however don't nest more than five levels. Nesting has an impact on group maintenance and performance due to group membership calculation. As a best practice, keep the total number of public groups for an organization to 100,000.

Public groups can be access via these steps:

1. Go to Setup > Administer > Manage Users > Public Groups

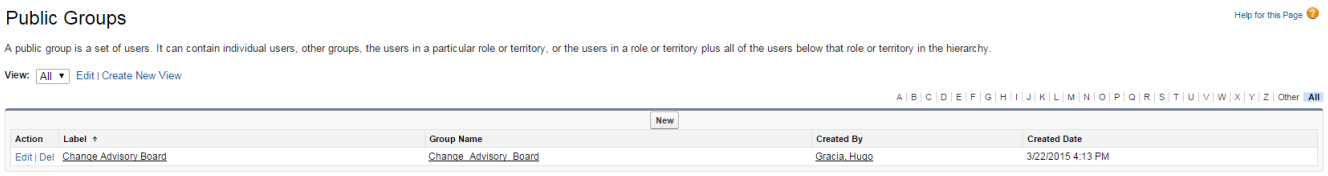



Figure 4 - Public Groups

## Ownership-based Sharing Rules

Ownership-based sharing rules allow for exceptions to organization-wide default settings and the role hierarchy that give additional users access to records they don't own. Ownership-based sharing rules are based on the record owner only. Contact ownership-based sharing rules don't apply to private contacts. As a best practice, keep the number of ownership-based sharing rules per object to 1,000.

Below is an example of an ownership based sharing rule:

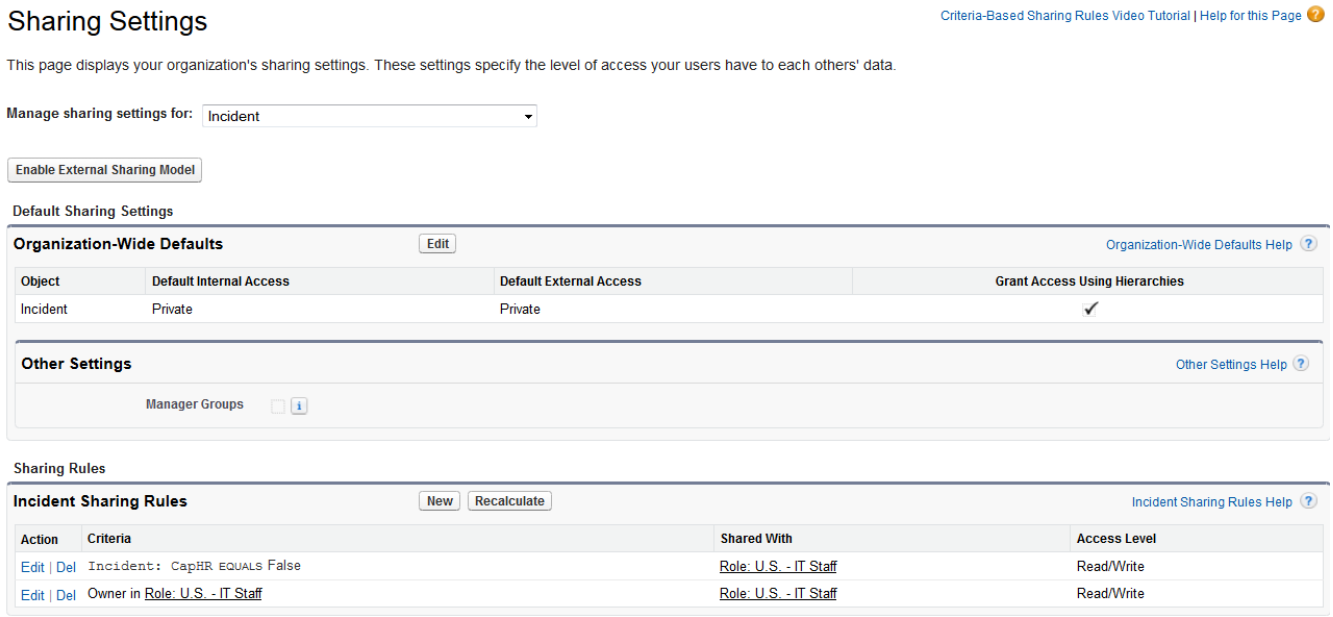



Figure 5 -Sharing Rule 1 of 2

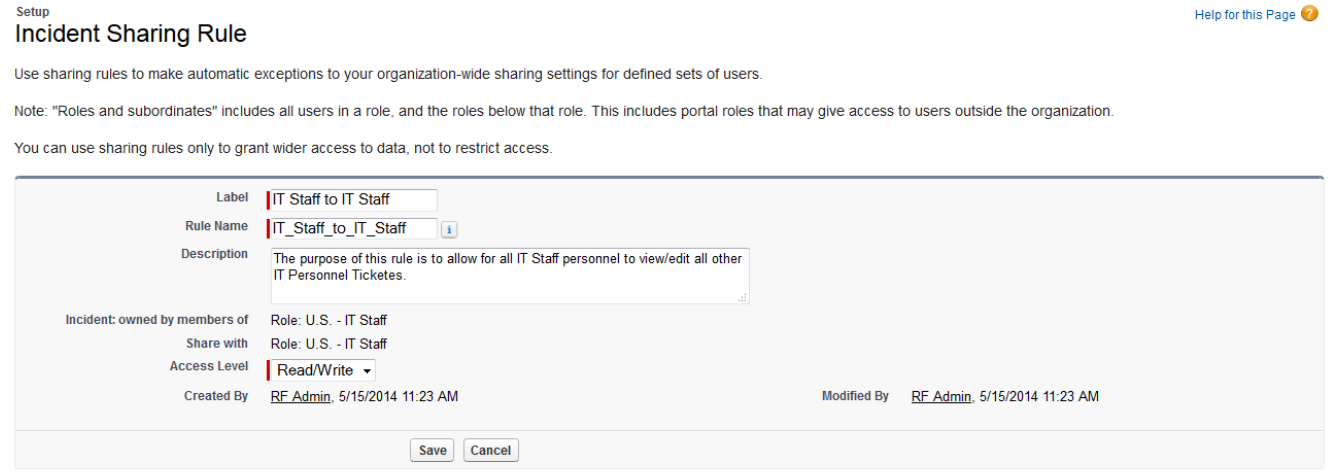



Figure 6 - Sharing Rule 2 of 2

## Criteria-based Sharing Rules

Criteria-based sharing rules provide access to records based on the record's field values (criteria). If the criteria are met (one or many field values), then a share record is created for the rule. Record ownership is not a consideration. As a best practice, keep the number of criteria-sharing rules per object to 50; however, this can be increased by Salesforce.

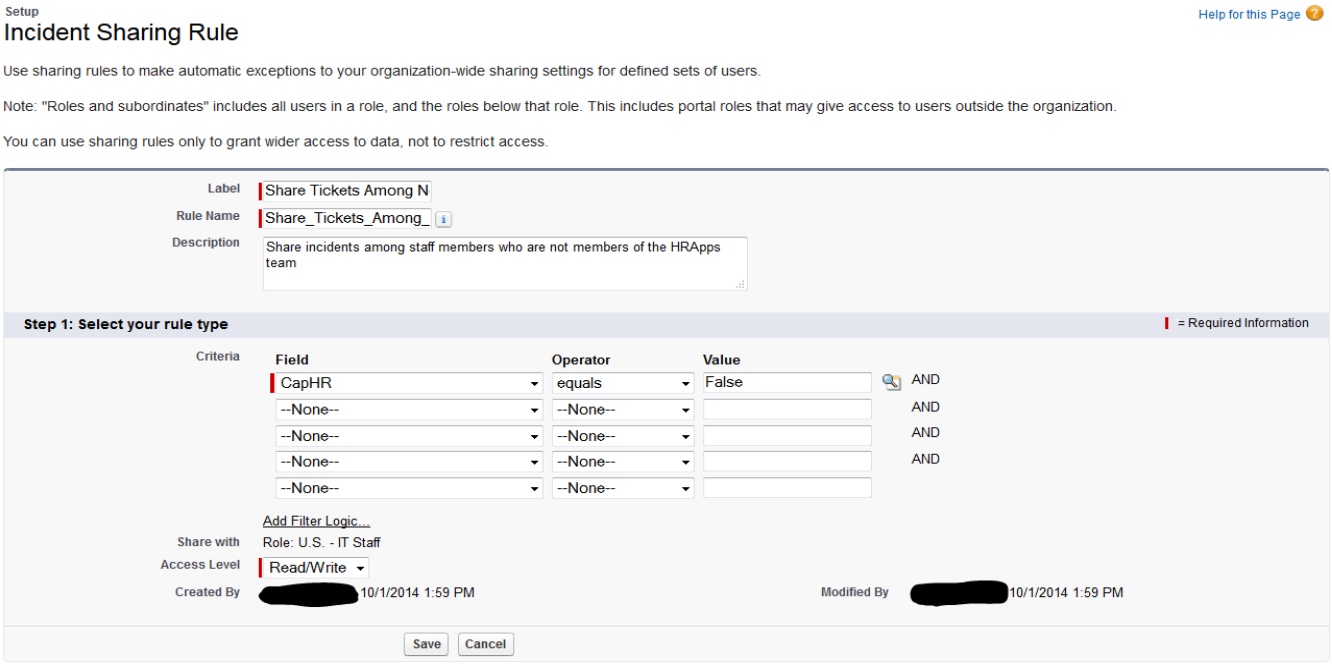



Figure 7 - Criteria Based Sharing Rule

## Manual Sharing

Sometimes it's impossible to define a consistent group of users who need access to a particular set of records. In those situations, record owners can use manual sharing to give read and edit permissions to users who would not have access to the record any other way. Although manual sharing isn't automated like organization-wide sharing settings, role hierarchies, or sharing rules, it gives record owners the flexibility to share particular records with users that need to see them.

Manual sharing is removed when the record owner changes or when the sharing access granted doesn't grant additional access beyond the object's organization-wide sharing default access level. This also applies to manual shares created programmatically. Only manual share records can be created on standard objects. Manual share records are defined as share records with the row cause set to manual share. All share records (standard and custom objects) with a row cause set to *manual share* can be edited and deleted by the Share button on the object's page layout, even if the share record was created programmatically.

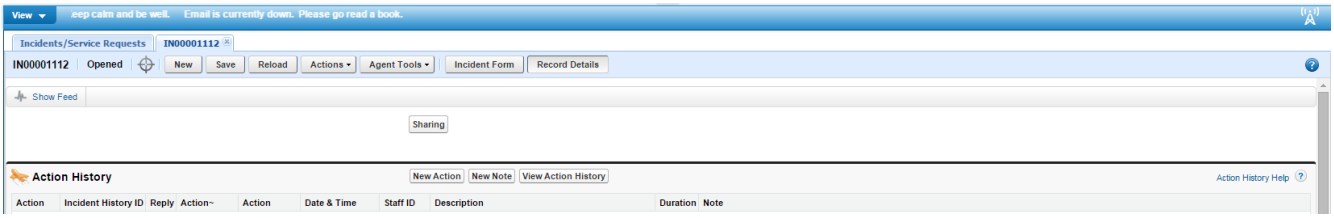



Figure 8 - Page Layout with Sharing Button

By clicking on the Sharing button, you will receive a screen that shows you who currently has access to it. (Figure 9) Once you click on the Add button, you will be able to add additional sharing permissions for this record only. (Figure 10)

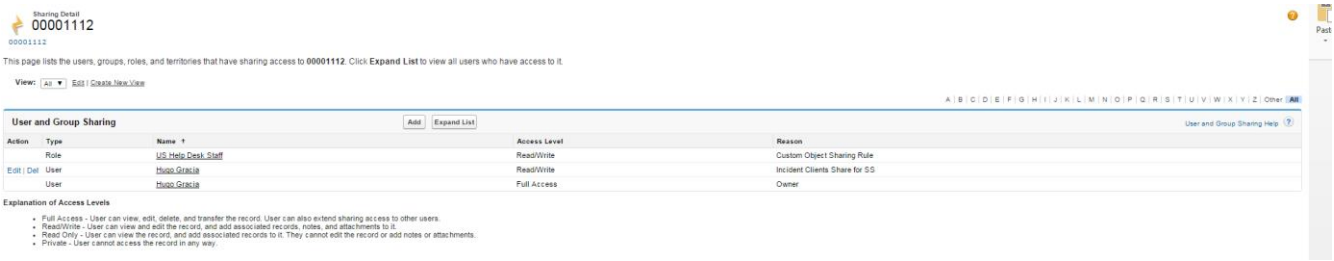



Figure 9 - Current Sharing Visibility

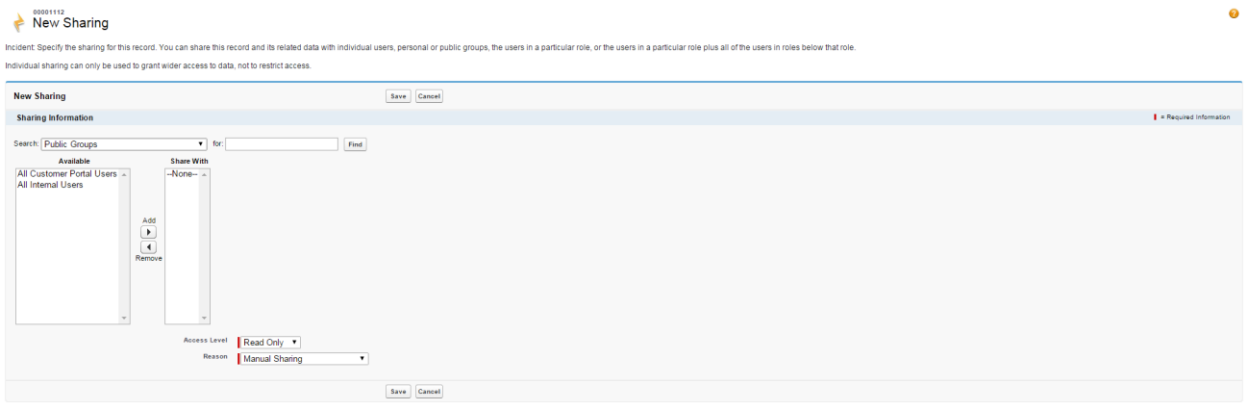



Figure 10 - Additional Sharing Permissions

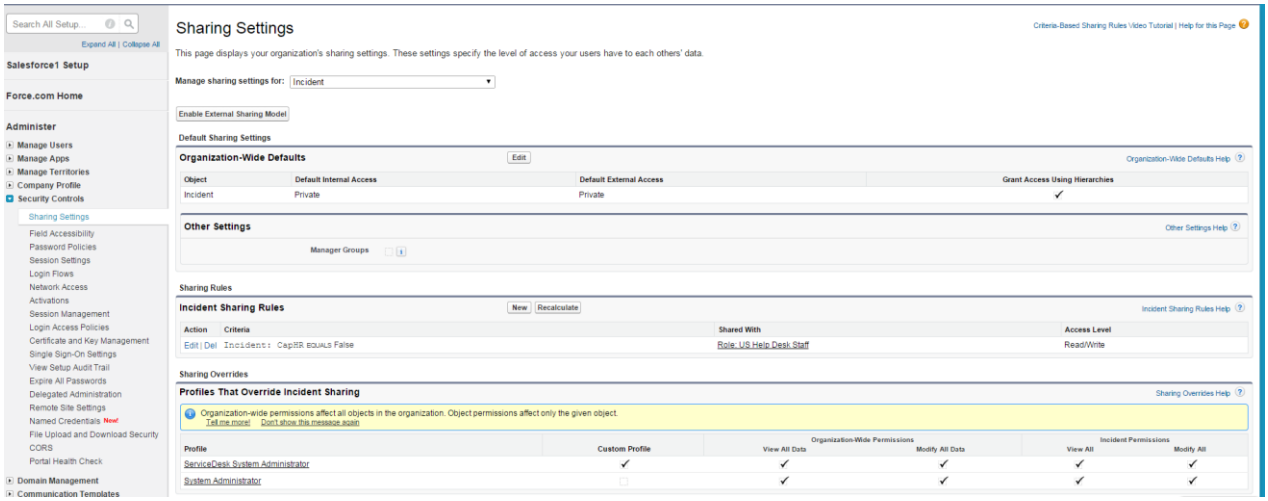



Figure 11 – Sharing Settings

# Troubleshooting

Once you have completed your sharing model architecture, you will likely be challenged with why a user can or can't see a record. Typically, you won't hear when someone can see something they shouldn't, but should that arise, there is a way to see every user who has access to a record and why. You can access these settings by navigating to Setup > Administer > Security Controls > Sharing Settings. Once in there, select the object from the drop down list. Refer to figure 11 below for the screen details. The more difficult challenge, and probably more common, is why a user can't see a record. The security layers you have architected will determine where you start. If you know the sharing model well, then you will probably know what component should have provided the access and should start there. But if you are less familiar with the sharing model, start with the role hierarchy and peel back each layer to determine which one should provide the access and make an update. Here is a sample troubleshooting flow.

1) Verify that the user has permissions to access to the object.
2) Identify the user's role who can't see the record and note it.
3) Identify the owner's role of the record and note it.
4) Review the role hierarchy and verify these two roles are in two different branches (they should be).
5) Review the sharing rules for the object and make sure there is no rule that will grant the user access.
    a. This can also cause you to look in public groups as well. Maybe the user just got left out of a group where there is a sharing rule, or does it make sense to create a new sharing rule to grant the user access? This depends on the architecture you are trying to maintain, and applies to both ownership-based sharing rules and criteria-based sharing rules.
    b. If you are using teams, should this user be on the team for that record? How are teams maintained and how did the miss occur?
    c. If manual sharing is used, the user may have lost access because the record owner changed. Manual shares are dropped when ownership changes. The manual share could also have been removed using the Share button.
6) If you are using territory management, determine if the user missing from one of the territories. Where is the membership of territories maintained and how did the miss occur? Or, maybe the record did not get stamped with the territory where the user is a member.
7) If you are creating programmatic shares and there are criteria for creating the share in code, review the code to understand why this user was omitted.

# Summary

Data segregation can be, and usually is, a complex undertaking. The 80/20 rule applies here in the sense that 80% is planning and 20% is configuration. Although this document focuses on a non-multi-tenancy organization, the principles apply just the same, with the one exception being that, in a multi-tenancy organization, you will also be dealing with ***accounts*** and account hierarchy. Nonetheless, a well-planned out sharing architecture can serve the organization for a long time to come, providing secure record access.

Fortunately, the BMC Remedyforce Services team is here to assist you in focusing on developing a data sharing strategy that enables you to custom fit the approach for your organization. You can learn more about BMC Remedyforce Services at http://www.bmc.com/it-services/remedyforce-services.html.

BMC Remedyforce has an extremely active user community where you can get answers to additional questions on this topic. We encourage you to take a look at bmc.com/communities.

BMC delivers software solutions that help IT transform digital enterprises for the ultimate competitive business advantage. We have worked with thousands of leading companies to create and deliver powerful IT management services. From mainframe to cloud to mobile, we pair high-speed digital innovation with robust IT industrialization—allowing our customers to provide amazing user experiences with optimized IT performance, cost, compliance, and productivity. We believe that technology is the heart of every business, and that IT drives business to the digital age.

**BMC – Bring IT to Life.**