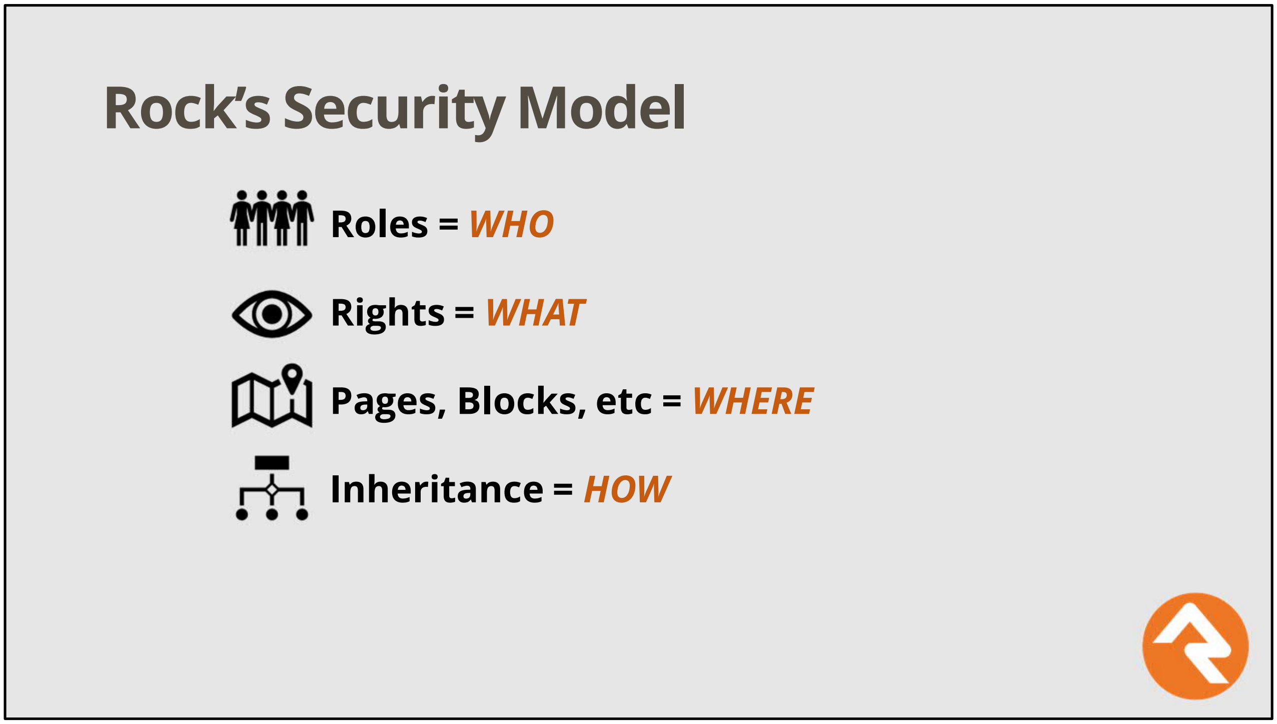
Demystifying Rock Security

Me

Jim Michael
IT Manager / Rock Wrangler™
The Crossing **(St. Louis!)**
@jimmichael

Me and my wife. Just wanted to point out I'm from The Crossing near ST LOUIS, since there are many other "The Crossing"s and it's easy to get us confused.



Rock's Security Model

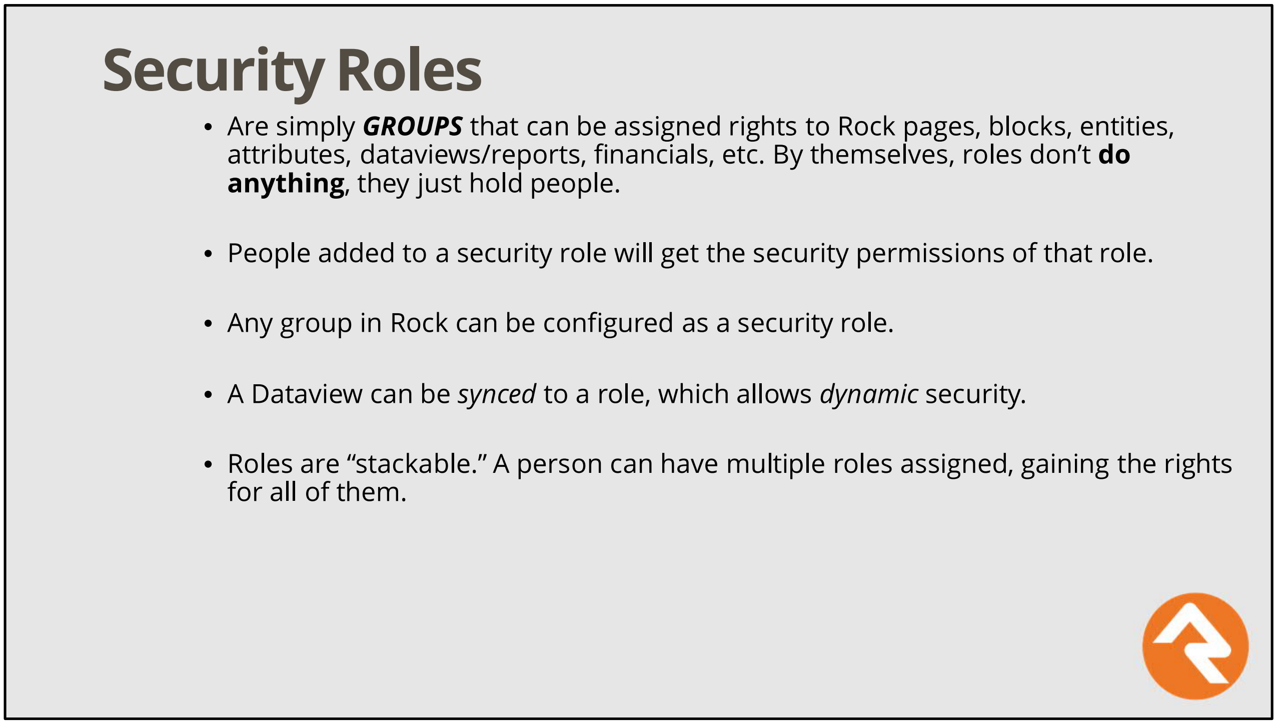
- Roles = *WHO*
- Rights = *WHAT*
- Pages, Blocks, etc = *WHERE*
- Inheritance = *HOW*

Roles are the *who* you're applying security to… most of the time these are people in "Security Roles"

Rights are *what kind* of security you're applying (View, Edit, Administrate, etc)

Pages, Blocks, Attributes, Entities, Reports, Dataviews, etc. are *where* you're applying the security.

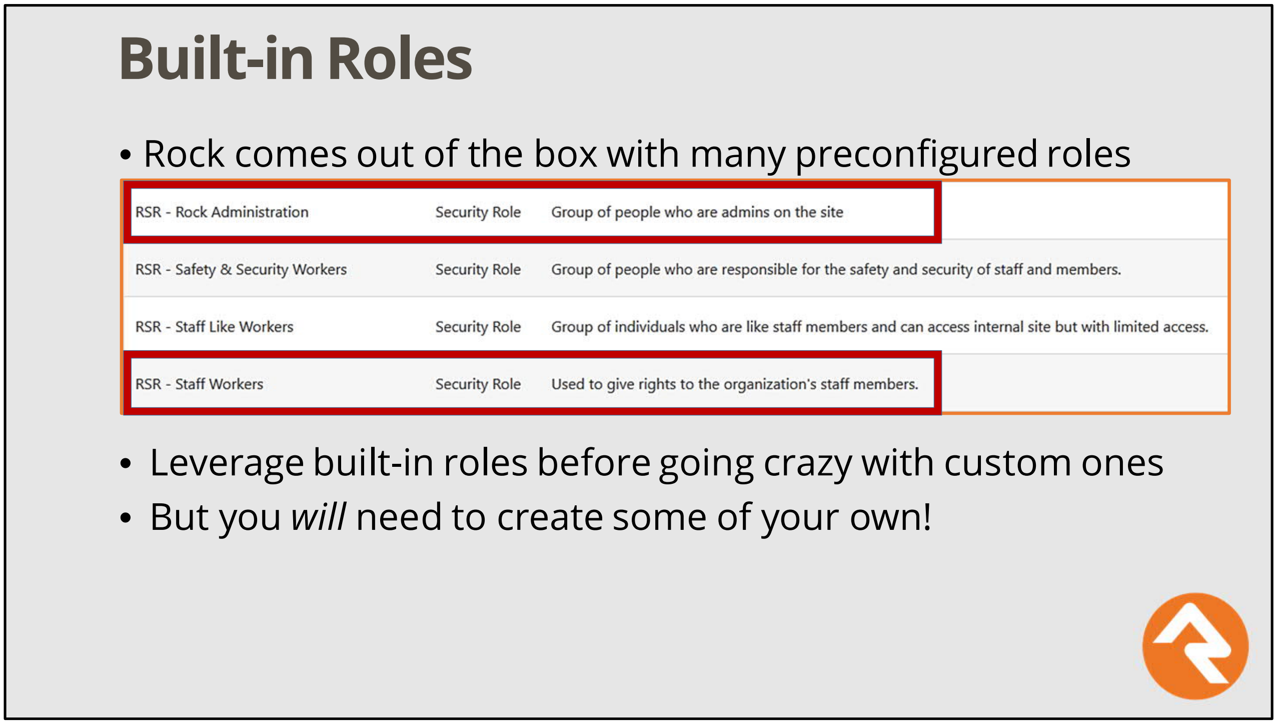
Inheritance (or overriding existing inheritance) is *how* the security is being applied.



Security Roles

- Are simply ***GROUPS*** that can be assigned rights to Rock pages, blocks, entities, attributes, dataviews/reports, financials, etc. By themselves, roles don't **do anything**, they just hold people.
- People added to a security role will get the security permissions of that role.
- Any group in Rock can be configured as a security role.
- A Dataview can be *synced* to a role, which allows *dynamic* security.
- Roles are "stackable." A person can have multiple roles assigned, gaining the rights for all of them.

Roles are the WHO you apply security to, and are nothing more than groups. By themselves, they don't "do anything"… they just hold people.



Built-in Roles

- Rock comes out of the box with many preconfigured roles

RSR - Rock Administration	Security Role	Group of people who are admins on the site
RSR - Safety & Security Workers	Security Role	Group of people who are responsible for the safety and security of staff and members.
RSR - Staff Like Workers	Security Role	Group of individuals who are like staff members and can access internal site but with limited access.
RSR - Staff Workers	Security Role	Used to give rights to the organization's staff members.

- Leverage built-in roles before going crazy with custom ones
- But you *will* need to create some of your own!

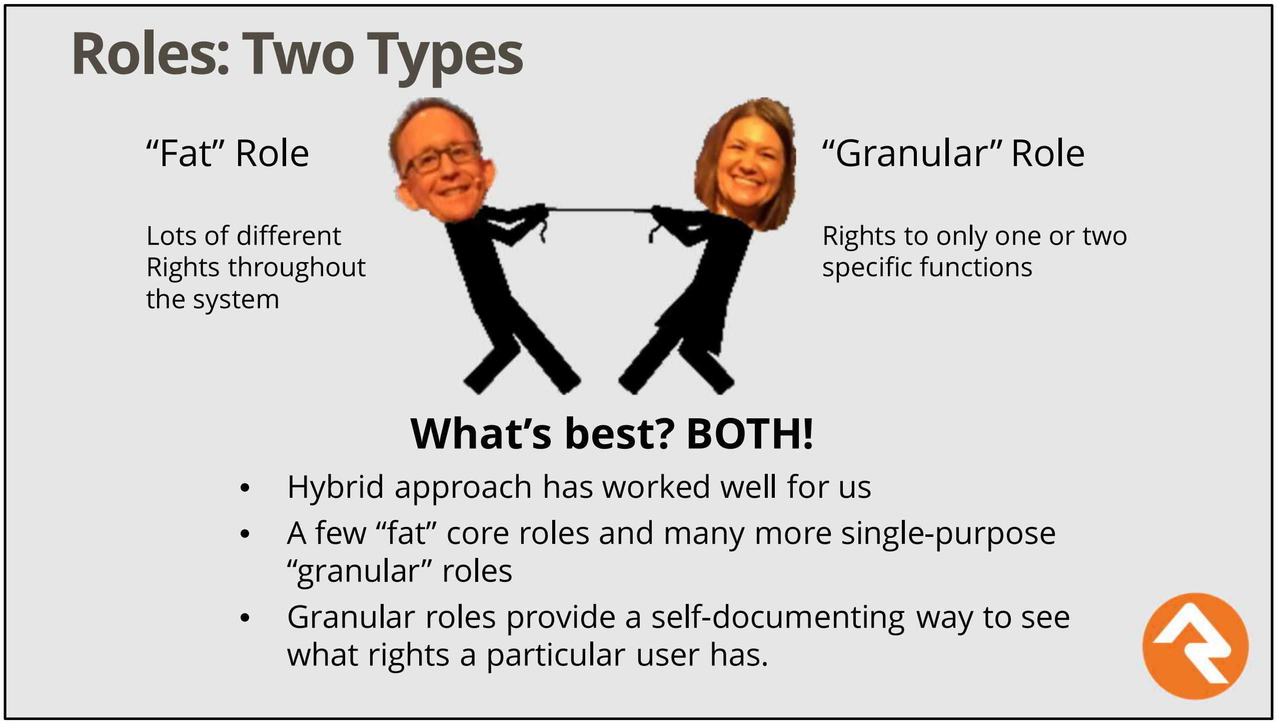
Two most important roles are Admin and Staff Workers.

Consider Staff Workers the “baseline” role for people that need to log into Internal Rock.

Roles like Finance are added IN ADDITION to Staff Workers. Most roles (other than Rock Admin, Staff, Staff-Like) will NOT let you log into Internal Rock by themselves.

Staff-Like workers has exactly the same rights as Staff Workers. It’s simply intended for people who need to “act like” Staff (but are not). Over time you will likely remove rights from Staff-like as it is too powerful for most volunteers, but it’s a good baseline for a high level volunteer role.

While Rock comes with a lot of pre-configurated roles, you WILL have to create your own, especially around reporting, groups, and content channels because those areas/trees aren’t built out when you initially install Rock.



Roles: Two Types

“Fat” Role

Lots of different Rights throughout the system

“Granular” Role

Rights to only one or two specific functions

What’s best? BOTH!

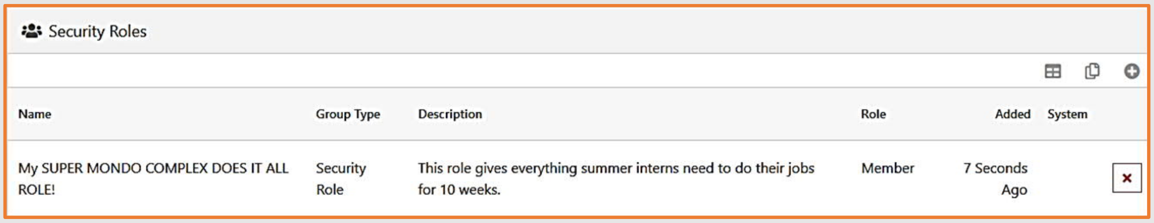
- Hybrid approach has worked well for us
- A few “fat” core roles and many more single-purpose “granular” roles
- Granular roles provide a self-documenting way to see what rights a particular user has.

There is a natural tension between two types of Roles…

Which is better? BOTH

Roles

Would you rather see this…



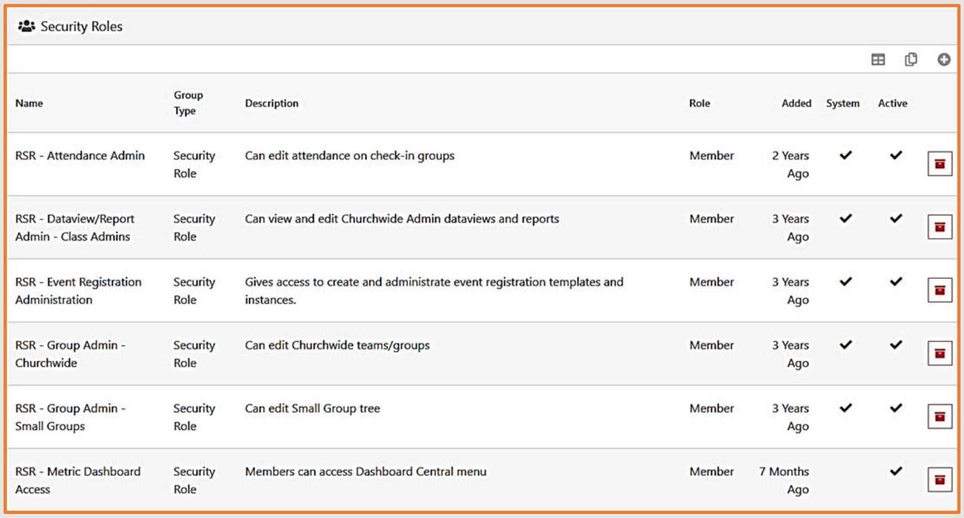
Security Roles

Name	Group Type	Description	Role	Added	System	
My SUPER MONDO COMPLEX DOES IT ALL ROLE!	Security Role	This role gives everything summer interns need to do their jobs for 10 weeks.	Member	7 Seconds Ago		☒

Here's an example of a Fat role which does a ton of stuff… but you have no easy way to know WHERE it has rights throughout the system.

Roles

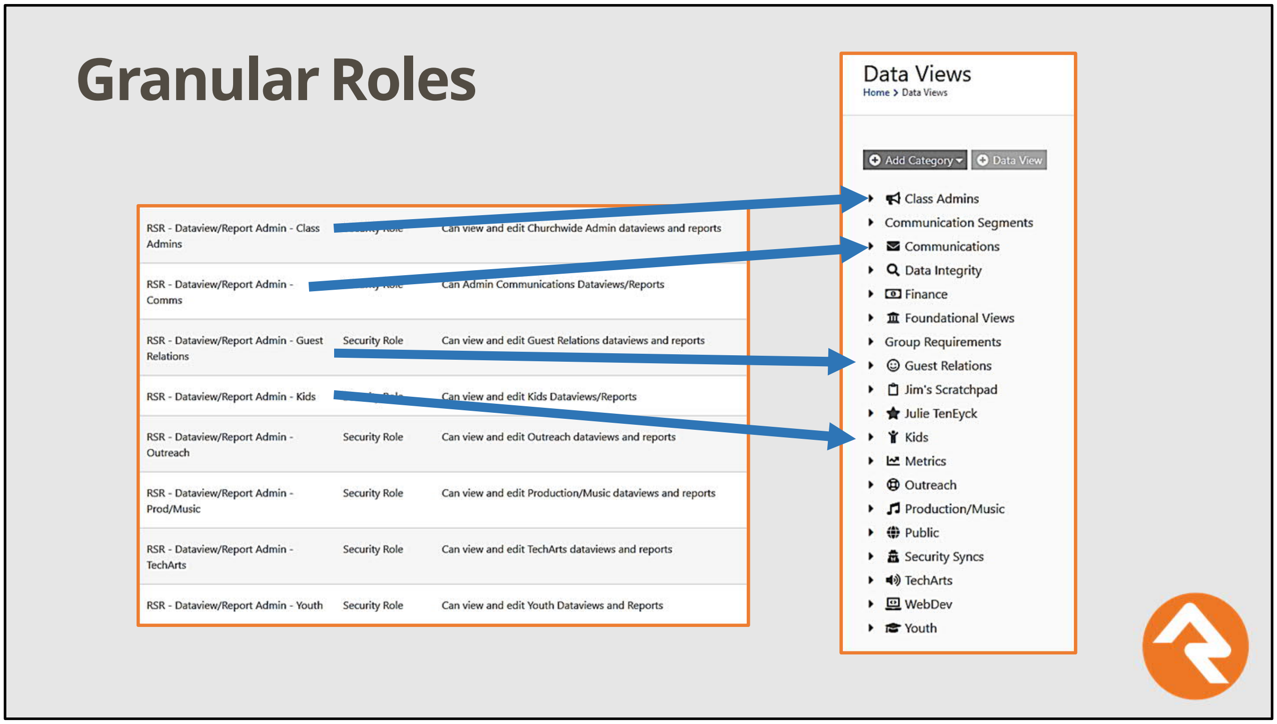
…or this?



Security Roles

Name	Group Type	Description	Role	Added	System	Active	
RSR - Attendance Admin	Security Role	Can edit attendance on check-in groups	Member	2 Years Ago	✓	✓	▣
RSR - Dataview/Report Admin - Class Admins	Security Role	Can view and edit Churchwide Admin dataviews and reports	Member	3 Years Ago	✓	✓	▣
RSR - Event Registration Administration	Security Role	Gives access to create and administrate event registration templates and instances.	Member	3 Years Ago	✓	✓	▣
RSR - Group Admin - Churchwide	Security Role	Can edit Churchwide teams/groups	Member	3 Years Ago	✓	✓	▣
RSR - Group Admin - Small Groups	Security Role	Can edit Small Group tree	Member	3 Years Ago	✓	✓	▣
RSR - Metric Dashboard Access	Security Role	Members can access Dashboard Central menu	Member	7 Months Ago		✓	▣

Here’s another user, but with multiple granular roles applied instead of one “does everything they need” role. A stacked list of granular roles is self-documenting, telling you pretty much exactly what this person has rights to.



Granular Roles

RSR - Dataview/Report Admin - Class Admins	Security Role	Can view and edit Churchwide Admin dataviews and reports
RSR - Dataview/Report Admin - Comms	Security Role	Can Admin Communications Dataviews/Reports
RSR - Dataview/Report Admin - Guest Relations	Security Role	Can view and edit Guest Relations dataviews and reports
RSR - Dataview/Report Admin - Kids	Security Role	Can view and edit Kids Dataviews/Reports
RSR - Dataview/Report Admin - Outreach	Security Role	Can view and edit Outreach dataviews and reports
RSR - Dataview/Report Admin - Prod/Music	Security Role	Can view and edit Production/Music dataviews and reports
RSR - Dataview/Report Admin - TechArts	Security Role	Can view and edit TechArts dataviews and reports
RSR - Dataview/Report Admin - Youth	Security Role	Can view and edit Youth Dataviews and Reports

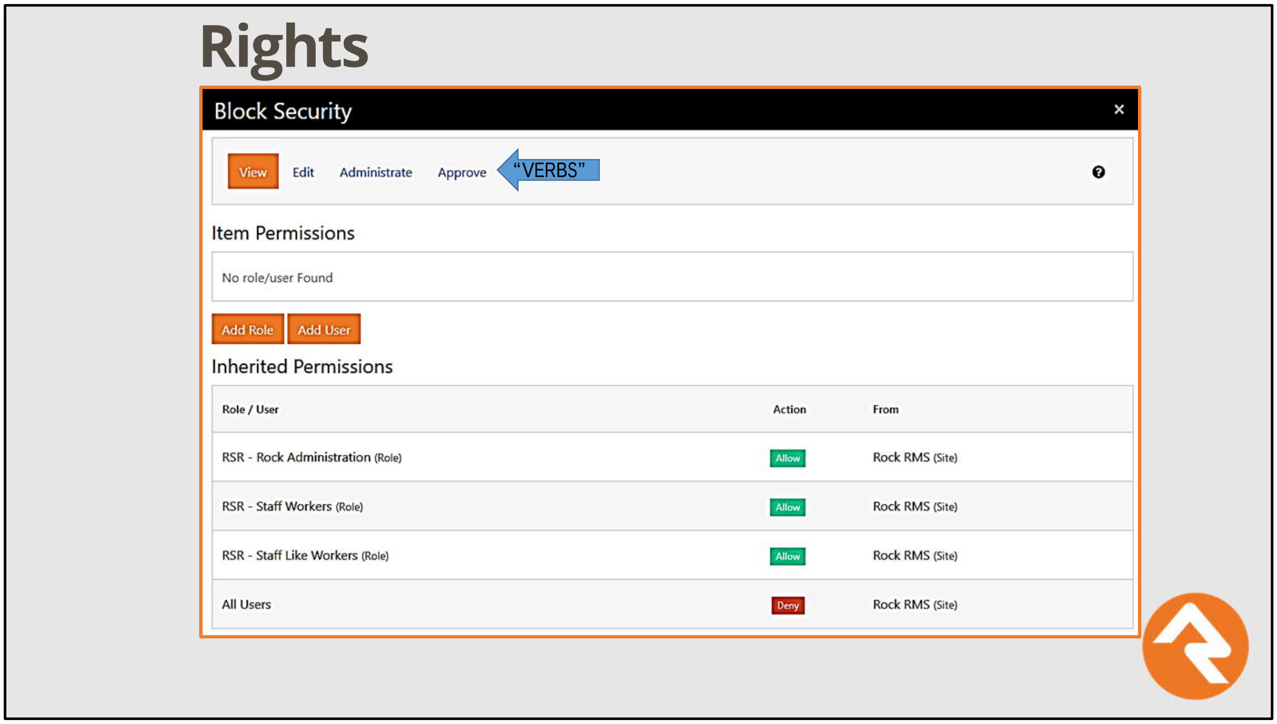
Data Views

Home > Data Views

- Class Admins
- Communication Segments
- Communications
- Data Integrity
- Finance
- Foundational Views
- Group Requirements
- Guest Relations
- Jim's Scratchpad
- Julie TenEyck
- Kids
- Metrics
- Outreach
- Production/Music
- Public
- Security Syncs
- TechArts
- WebDev
- Youth

Here we see how granular roles map to the one-or-two things they have rights to.

There is nothing magical about roles… they are nothing but groups of people. You must ASSIGN the role some RIGHTS in order for it to do anything useful.



Rights

Block Security

Item Permissions

No role/user Found

Inherited Permissions

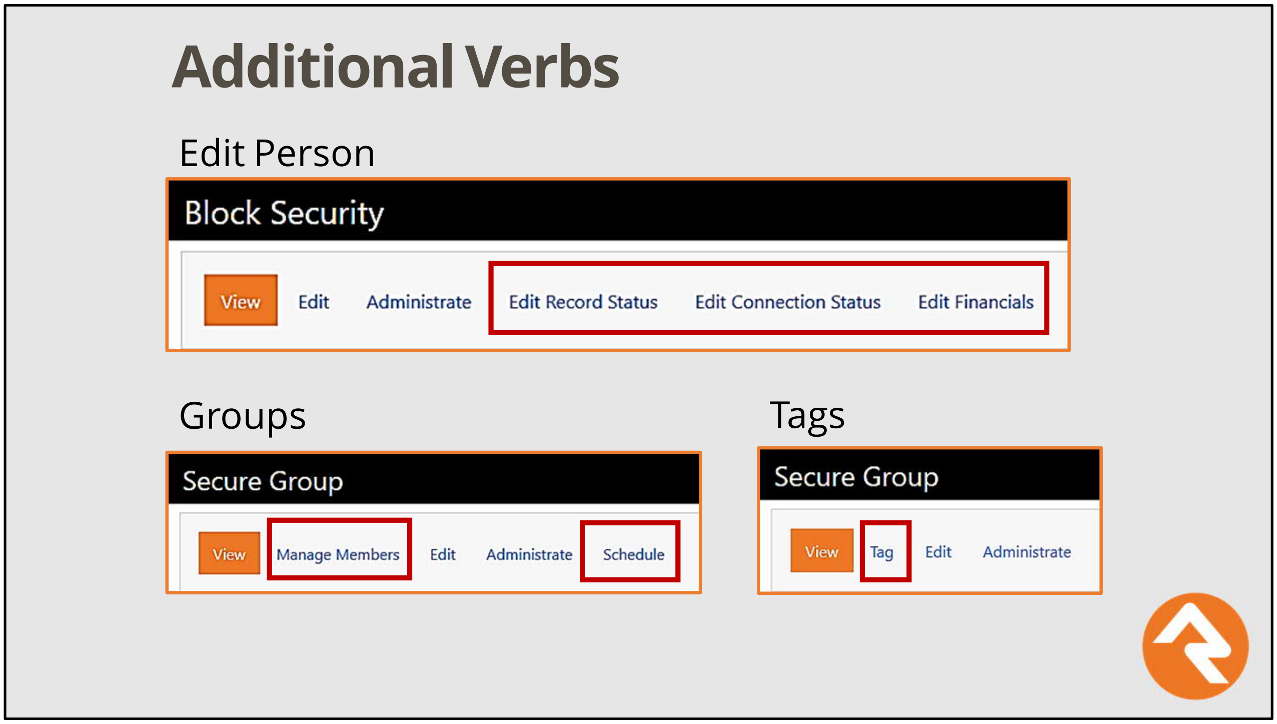
Role / User	Action	From
RSR - Rock Administration (Role)	Allow	Rock RMS (Site)
RSR - Staff Workers (Role)	Allow	Rock RMS (Site)
RSR - Staff Like Workers (Role)	Allow	Rock RMS (Site)
All Users	Deny	Rock RMS (Site)

If Roles are the WHO, then Rights are the WHAT… specifically what KIND of access you’re granting or denying.

View = Person can view this page, block , attribute, dataview, etc.

Edit = Person can edit the page, block, attribute, etc.

Approve = Person can approve content, event, etc.



Additional Verbs

Edit Person

Groups

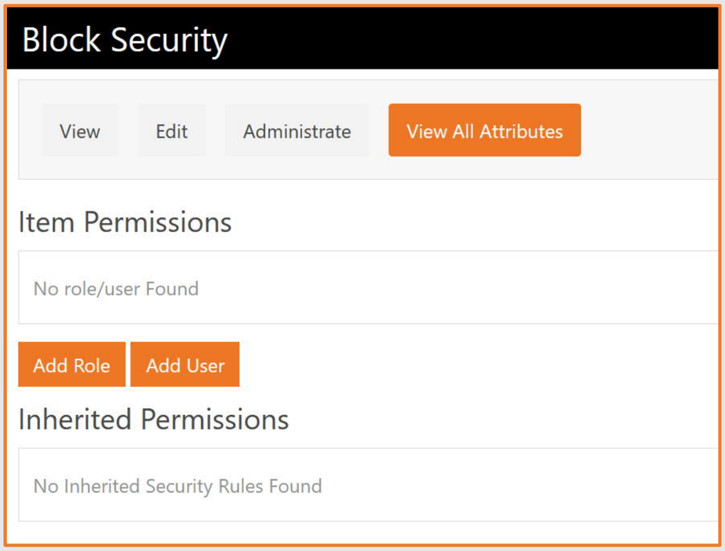
Tags

EDIT PERSON BLOCK: Additional rights to limit who can change certain things vs. EDIT which lets you change everything.

GROUPS: MANAGE MEMBERS lets you add/remove people from Groups without editing the group itself. SCHEDULE lets you use scheduling functionality without needing full Edit rights.

TAGS: TAG verb lets you control who can use a specific org tag. VIEW lets you control who can even SEE a tag.

Person Merge



Block Security

Item Permissions

No role/user Found

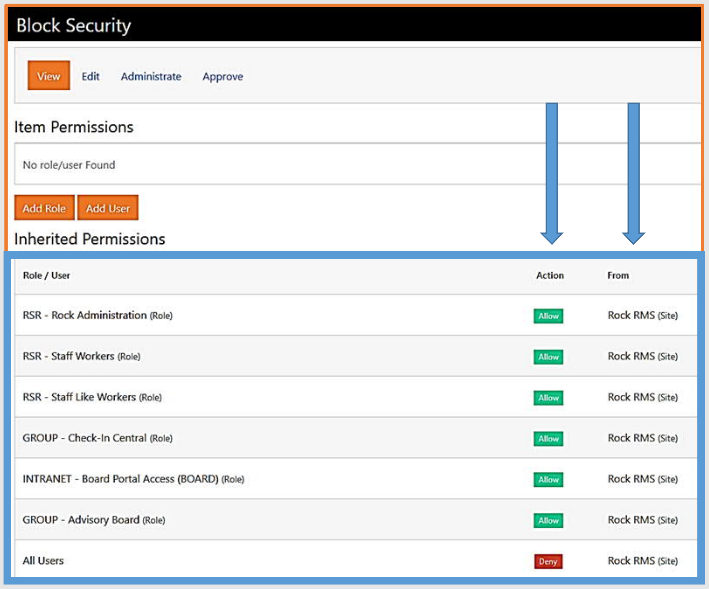
Inherited Permissions

No Inherited Security Rules Found

VIEW ALL ATTRIBUTES on Person Merge block is relatively new. This lets a non-Rock Admin do merges safely, by granting them view rights to Person Attributes they otherwise don't have normal View rights to. This is important because without View Rights to ALL Person Attributes, you could lose data when merging two people with different values in their attributes.

I recommend you add Rock Admin here because it's possible you have restricted View access to a sensitive person attribute from Admin role, which would cause a problem even for THAT role doing merges. Add any other roles here that need to merge, too.

Inheritance



Block Security

Item Permissions

No role/user Found

Inherited Permissions

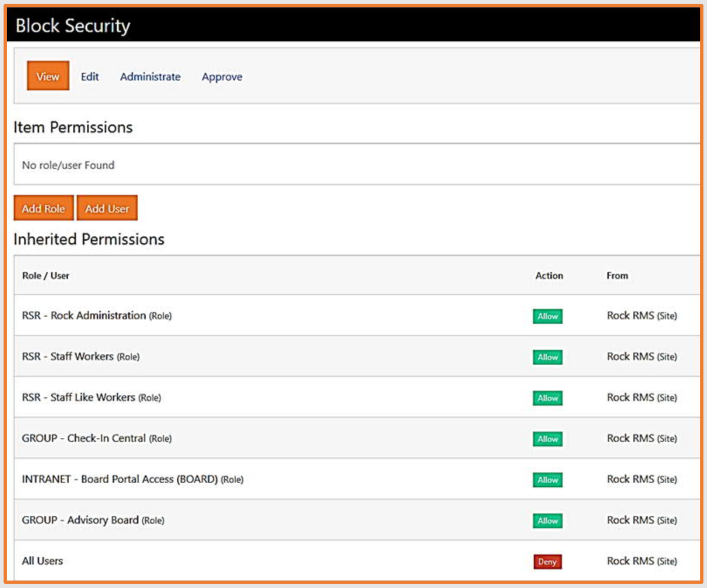
Role / User	Action	From
RSR - Rock Administration (Role)	Allow	Rock RMS (Site)
RSR - Staff Workers (Role)	Allow	Rock RMS (Site)
RSR - Staff Like Workers (Role)	Allow	Rock RMS (Site)
GROUP - Check-In Central (Role)	Allow	Rock RMS (Site)
INTRANET - Board Portal Access (BOARD) (Role)	Allow	Rock RMS (Site)
GROUP - Advisory Board (Role)	Allow	Rock RMS (Site)
All Users	Deny	Rock RMS (Site)

Inherited permissions are the rights the page/block got from somewhere else.

It's important to understand WHERE an inherited right is coming from, and Rock makes that crystal clear.

A common question is HOW DO I DELETE AN INHERITED RIGHT? You can't!

Inheritance



Item Permissions

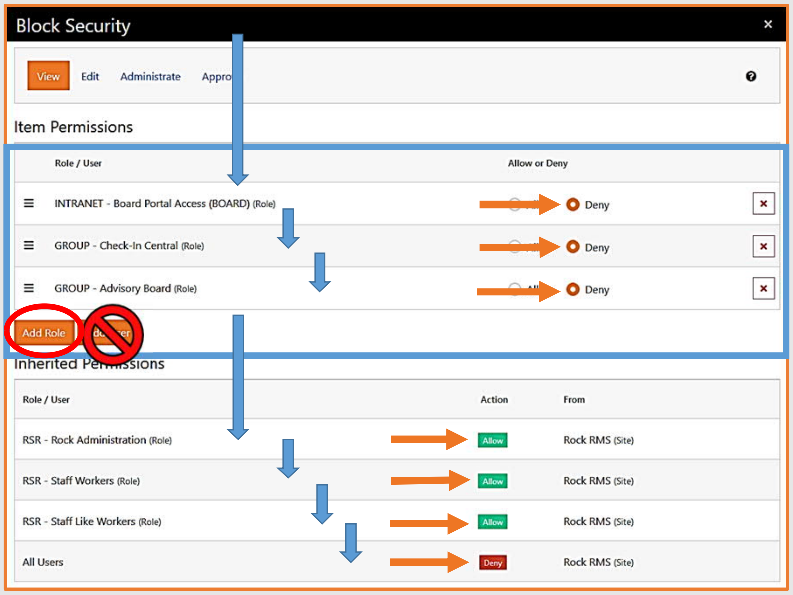
No role/user Found

Inherited Permissions

Role / User	Action	From
RSR - Rock Administration (Role)	Allow	Rock RMS (Site)
RSR - Staff Workers (Role)	Allow	Rock RMS (Site)
RSR - Staff Like Workers (Role)	Allow	Rock RMS (Site)
GROUP - Check-In Central (Role)	Allow	Rock RMS (Site)
INTRANET - Board Portal Access (BOARD) (Role)	Allow	Rock RMS (Site)
GROUP - Advisory Board (Role)	Allow	Rock RMS (Site)
All Users	Deny	Rock RMS (Site)

So if we can’t delete inherited rights and we don’t want a role to have the inherited right, what’s the solution…? OVERRIDING the inherited rights!

Item Permissions / Rights Evaluation



Item Permissions

Role / User	Allow or Deny
INTRANET - Board Portal Access (BOARD) (Role)	Deny
GROUP - Check-In Central (Role)	Deny
GROUP - Advisory Board (Role)	Deny

Inherited Permissions

Role / User	Action	From
RSR - Rock Administration (Role)	Allow	Rock RMS (Site)
RSR - Staff Workers (Role)	Allow	Rock RMS (Site)
RSR - Staff Like Workers (Role)	Allow	Rock RMS (Site)
All Users	Deny	Rock RMS (Site)

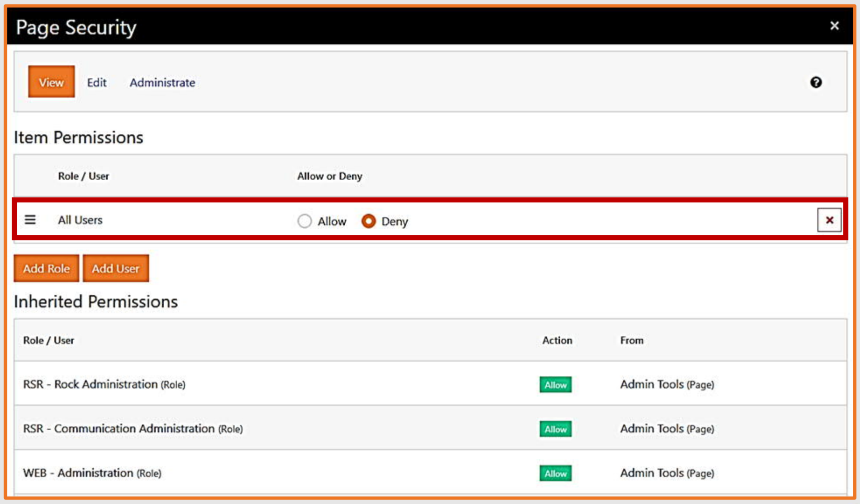
Item Permissions should be used when you want to OVERRIDE the rights that were inherited (for example, a Role that inherits ALLOW VIEW rights can be added as an Item Permission to DENY View, thus “overriding” the allowed inherited right.

Always leverage inherited rights when you can, vs. adding redundant Item Permissions that do the same thing.

Avoid applying rights to a specific person instead of a role. It will come back to bite you in the future if that person leaves, because (most likely) you will have no idea *all of the places* you need to go re-assign the rights to the NEW person taking over those duties, and even worse… you have no idea where to go to REMOVE the rights for that departed person (because they’re still a person in Rock, even if they’re not on staff anymore).

ALWAYS use a Role, even if there’s only one person in it. Your future self will thank you!

Tip: Oops…



Page Security

View | Edit | Administrate

Item Permissions

Role / User	Allow or Deny
All Users	○ Allow ● Deny

Add Role | Add User

Inherited Permissions

Role / User	Action	From
RSR - Rock Administration (Role)	Allow	Admin Tools (Page)
RSR - Communication Administration (Role)	Allow	Admin Tools (Page)
WEB - Administration (Role)	Allow	Admin Tools (Page)

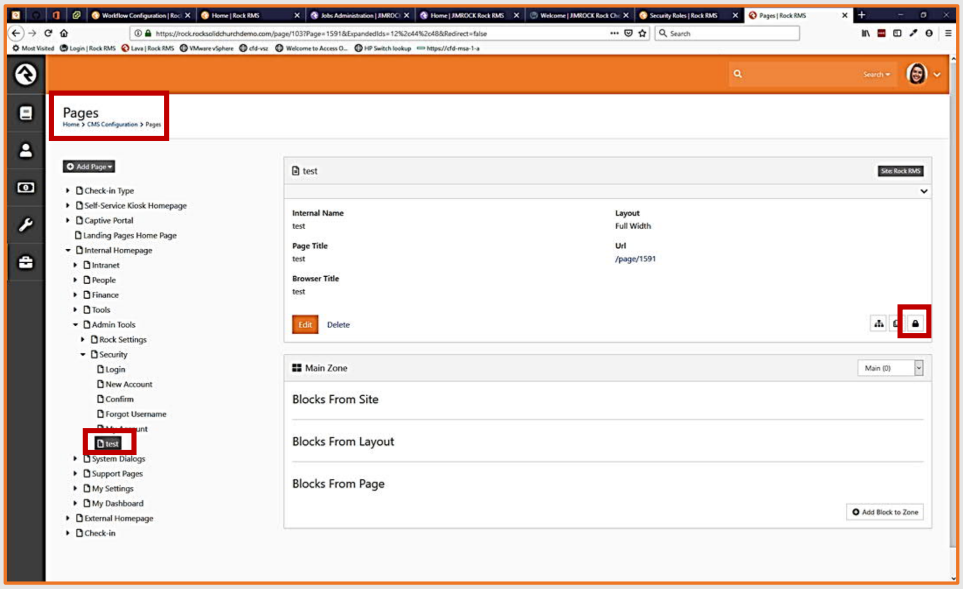
…what could possibly go wrong?

We've all locked ourselves out of a page, right? (If not, it will happen!) You just locked yourself out of that page/block because you (even Rock Admin) are indeed part of the All Users role.

The ORDER in which you add rights is important.

Always add Rock Admin FIRST when you're going to eventually adding a DENY ALL (to limit a page/block to only specific roles)

Fix #1 Admin | CMS Config | Pages



Internal Name
test

Page Title
test

Browser Title
test

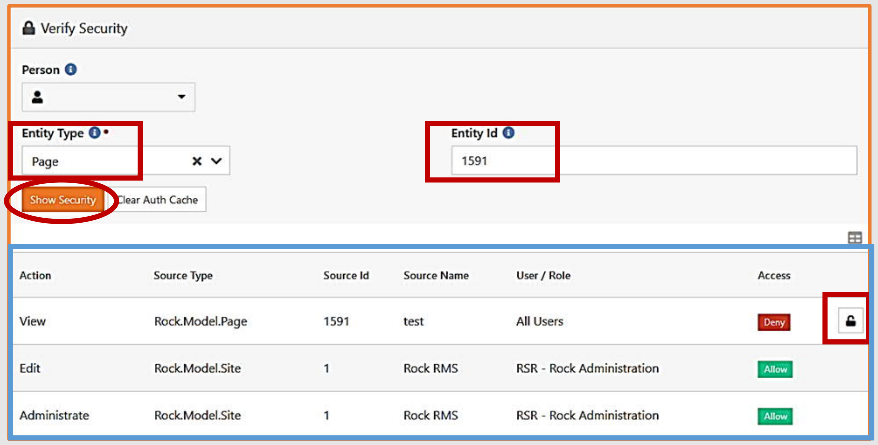
Layout
Full Width

Url
/page/1591

Use Admin | CMS Config | Pages to go find the page/block and simply change the rights back.

Fix #2

Admin | Security | Inspect Security



Person

Entity Type: Page

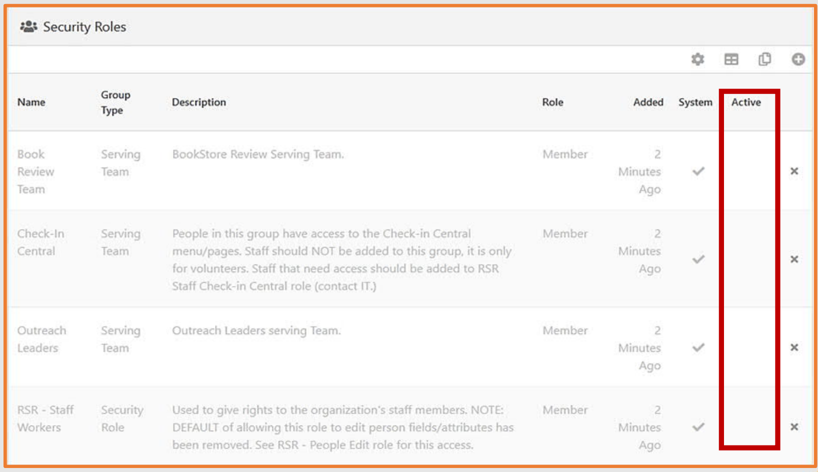
Entity Id: 1591

Action	Source Type	Source Id	Source Name	User / Role	Access	
View	Rock.Model.Page	1591	test	All Users	Deny	🔓
Edit	Rock.Model.Site	1	Rock RMS	RSR - Rock Administration	Allow	
Administrate	Rock.Model.Site	1	Rock RMS	RSR - Rock Administration	Allow	

Use Inspect Security to undo a Deny All on any Entity, as long as you know its Id. In a pinch, if you don't KNOW the Id of the Entity you're locked out of, you can just look it up via SQL.

Tip: Inactivating Users

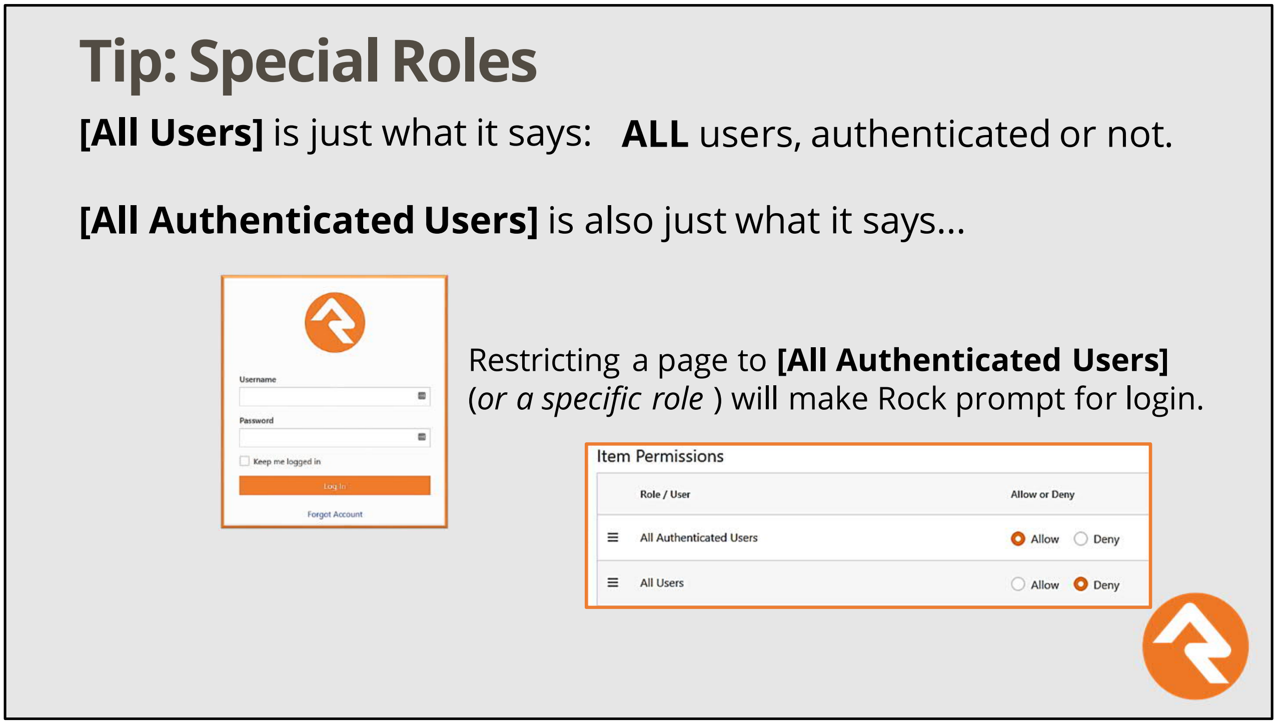
Be VERY careful when inactivating users…



Security Roles

Name	Group Type	Description	Role	Added	System	Active	
Book Review Team	Serving Team	BookStore Review Serving Team.	Member	2 Minutes Ago	✓		×
Check-In Central	Serving Team	People in this group have access to the Check-in Central menu/pages. Staff should NOT be added to this group, it is only for volunteers. Staff that need access should be added to RSR Staff Check-in Central role (contact IT.)	Member	2 Minutes Ago	✓		×
Outreach Leaders	Serving Team	Outreach Leaders serving Team.	Member	2 Minutes Ago	✓		×
RSR - Staff Workers	Security Role	Used to give rights to the organization's staff members. NOTE: DEFAULT of allowing this role to edit person fields/attributes has been removed. See RSR - People Edit role for this access.	Member	2 Minutes Ago	✓		×

By default, all Group Types are configured to INACTIVATE a person in any group they are a member of WHEN the person themselves in made inactive. This can cause unexpected consequences when you “play” with making yourself (the Rock admin) inactive., because it inactivates you from the Rock Admin and/or Staff Workers roles… and now you can’t even log into Rock!



Tip: Special Roles

[All Users] is just what it says: **ALL** users, authenticated or not.

[All Authenticated Users] is also just what it says…

Restricting a page to **[All Authenticated Users]** (*or a specific role*) will make Rock prompt for login.

Item Permissions

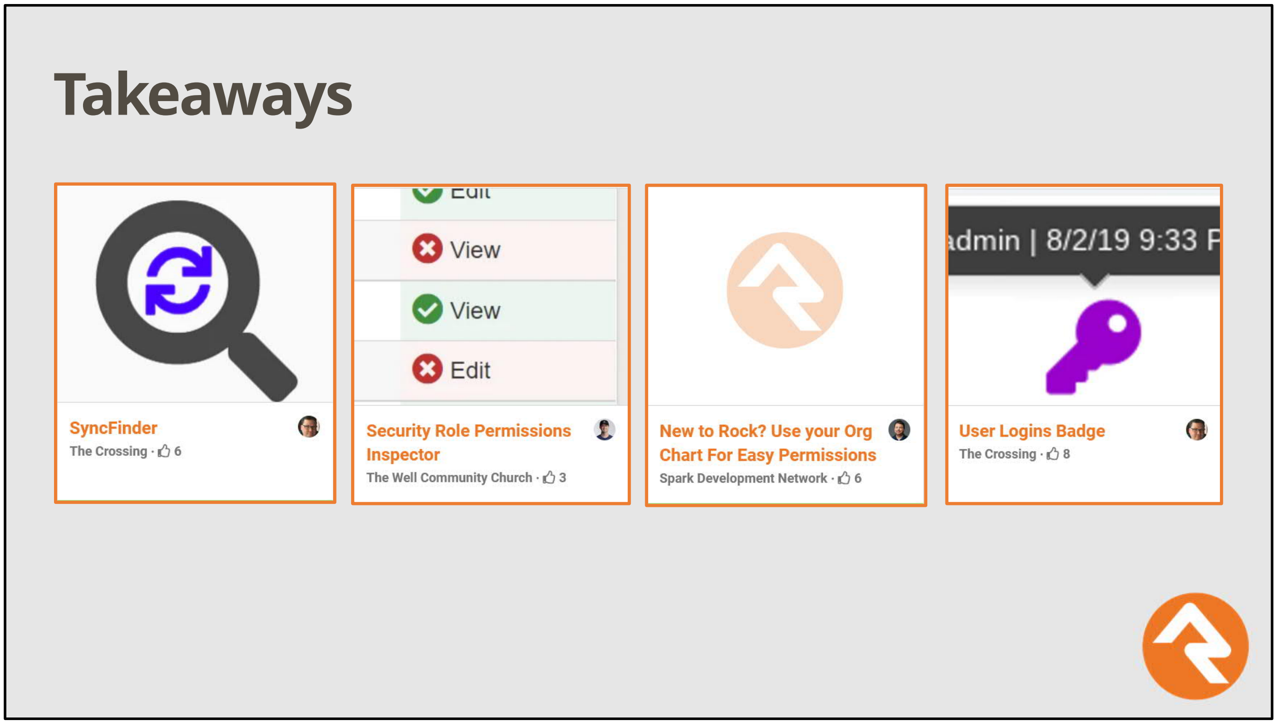
	Role / User	Allow or Deny
≡	All Authenticated Users	● Allow ○ Deny
≡	All Users	○ Allow ● Deny

[All Users] applies to ALL users in the system, regardless if they are authenticated or not.

ALL USERS VIEW is the equivalent of "this is a public page" that anyone can see.

[ALL AUTHENTICATED USERS] will cause Rock to prompt for login and ANY credentials will work... Restricting View to a specific Role will also prompt for login, but only people in that role will be allowed access.

[ALL UN-AUTHENTICATED USERS] is the inverse of AUTHENTICATED USERS, but has limited usefulness.



Takeaways

SyncFinder
The Crossing · 👍 6

Security Role Permissions Inspector
The Well Community Church · 👍 3

New to Rock? Use your Org Chart For Easy Permissions
Spark Development Network · 👍 6

User Logins Badge
The Crossing · 👍 8

Here are some Rock recipes that are involved with Security some way.

THANK YOU!

Hit me up in Rocket.Chat if you have specific security questions!