ServiceNow Application Developer

Securing Applications Against Unauthorized Users > Scripting Security

Both the client-side and server-side APIs have methods for scripting security.

The client-side ***GlideUser* (*g_user*) API (https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=c_GlideUserAPI)** has these methods:

- *hasRole()*
- *hasRoleExactly()*
- *hasRoleFromList()*
- *hasRoles()*

The client-side API methods can be used in any client-side script such as Client Scripts and UI Policy scripts. Client-side security is the easiest security to break. Do not depend on client-side scripts to secure sensitive data.

The server-side ***GlideSystem* (*gs*) API (https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=c_GlideSystemScopedAPI)** has these methods:

- *getUser()*
- *getUserID()*
- *getUserName()*
- *hasRole()*
- *isLoggedIn()*
- *isInteractive()*

- *getSession()*

The server-side ***GlideElement*** **API (https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=c_GlideElementScopedAPI)** has methods to check whether a user's role allows them to access the associated GlideRecord(s):

- *canCreate()*
- *canRead()*
- *canWrite()*

The server-side methods can be used in any server-side script such as Business Rules or Script Includes. Server-side scripted security is more secure than client-side scripted security. Any user with access to scripting fields can see the scripts and see what the security checks are.

Neither client-side nor server-side scripts are part of the *Debug Security Rules* module. When security is scripted outside of Access Controls, it must be debugged independently of the Access Controls.

For the highest level of security, use Access Controls to protect sensitive data.