SAPinsider

This article appeared in the Jul▪Aug▪Sep 2005 issue of *SAP Insider* and appears here with permission from the publisher, Wellesley Information Services (WIS), **www.WISpubs.com**.

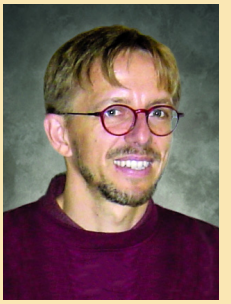
WIS

Secure the RFC Connections in Your SAP System Landscape

Regular Feature

Security Strategies

Sarah Maidstone, SAP AG



Frank Buchholz, SAP AG

Most discussion surrounding security today tends to focus on the changing security paradigm that goes hand in hand with increasingly open IT architectures. And rightly so, since opening up previously closed systems makes certain security aspects more vital than ever before. Too often, though, companies fail to review existing security measures when making fundamental changes to their security infrastructure.

In this article, we'll look at the consequences of increasingly complex and open system landscapes with respect to one tried and tested technology, Remote Function Call (RFC), SAP's interface protocol for cross-system communication. Many customers do not fully understand the risks involved in continuing to use RFC connections across their SAP landscapes, or how to implement a strong authorization concept to mitigate or avoid these risks.

> ✔ **Note!**
> Specific functionality in this article refers to SAP NetWeaver 2004 and applications based on it, though the basic security principles described here apply to earlier releases as well.

Before changing or expanding your system infrastructure — for example, if you're considering adding new SAP systems like SAP Enterprise Portal or mySAP CRM — be sure to re-examine the robustness of your RFC.

Revisiting RFC Security

A Remote Function Call involves calling a function module that runs in a different system (server) from the program that calls it (client). The information on how to reach the remote system — including data that describes the network connection, and authentication data for the RFC user — is stored in what is known as the *RFC destination*. While RFCs are most commonly used between two SAP systems, it is also possible to call specially programmed functions from non-SAP systems. This type of connectivity is generally used to replicate data — for example, when sharing master or transaction data, or when SAP BI reads information from other business systems using RFC and then produces reports based on that stored data.

RFCs are also used when centralizing certain system management functions — such as for SAP Solution Manager or Central User Administration, which take a sweeping, cross-system approach to SAP data and activities. Using these applications, an administrator logs on to one central SAP system but performs actions in all connected systems.

When you grant access across system boundaries, however, there is always a certain element of risk involved. In the case of RFC, there are two primary vulnerabilities:

1. In logon data for service users stored in the RFC destination
2. In the reliance on authorizations to repel any potential attack

Depending on the type of application in each case, you either need to configure a service user for the RFC destination, or to ensure that the appropriate user information is forwarded — and these users have to be assigned to appropriate authorizations. We'll step through these processes later in the article.

The Risks of Remote System Connection

Let's examine these vulnerabilities in more detail.

1. Logon Data for Service Users Stored in the RFC Destination

The first threat is in exposing service user data stored in the RFC destination. Consider, for example, that you need to run a report of open financial accounts on a nightly basis. To run this report, your data warehouse needs to pull data from your FI system automatically. Though there's not a human user directly keying in his or her user ID and password, your BI system still needs to log on to your FI system to access the necessary data, so a "service user" logs in as a generic BI system user.

A service user account is a technical user master data record where the user ID and password are stored in the client system.

Where service users are used, the logon data (user ID and password) for that service user is then stored in the RFC destination. Whenever another program uses the RFC destination, this logon data is used for user authentication. The service user's authorizations are also applied in the target system. This means that, regardless of the user's actual identity and authorizations within the calling system, he or she takes on the identity and authorizations of the service user once inside the target system. Here's where the vulnerability lies: An attacker could try to use the defined RFC destination to call other RFC function modules. The authorizations granted to the service user must block this type of attack.

So it follows that customers who choose to use service users for RFC destinations should carefully limit the authorizations granted to those service users. For a look at exactly what this involves, see the sidebar, "How to Determine the Right Authorizations: The Security Audit Log."

2. Reliance on Authorizations to Repel Any Potential Attack

Given that most system landscapes already include firewalls and gateways in one form or another, why do RFC connections still need to rely on authorizations to repel potential attacks? In an SAP landscape, any user can attempt to create an RFC connection between any two SAP systems, or even from his or her PC to an SAP system, using the RFC Software Development Kit (SDK)[1].

For example, an end user can use an RFC connection to retrieve reports from the backend data system through an Excel front end of SAP BI. In this case, even well-configured firewall rules, application gateway rules, and gateway access control rules would not normally provide any protection; if RFC connections are used at all in the landscape, then firewalls and gateways are configured to allow them through.

This is why SAP's robust, fine-grained authorization concept plays such a central role — because it's

How to Determine the Right Authorizations: The Security Audit Log

The Security Audit Log is a record of security-related system information such as configuration changes or unsuccessful logon attempts. To determine precisely which RFC-related authorizations are needed for each user, customers can activate the Security Audit Log for RFC calls (see **Figure 1**) in the test and production systems for a couple of months, and use the log results to build roles that contain the right authorizations. These roles are very useful, since customers should never assign any user full authorizations for the S_RFC authorization object. The performance and storage of the Security Audit Log have been optimized so that it is also suitable for use in production systems.

For more information on how to activate the Security Audit Log, visit **http://help.sap.com**, click the *Documentation* tab, the *SAP NetWeaver* tab, and enter the *SAP NetWeaver '04 (SPS 11)* documentation. Once in the SAP Library, follow this navigation path: *SAP NetWeaver* → *Security* → *System Security* → *System Security for SAP Web AS ABAP Only* → *Security Audit Log (BC-SEC)*.

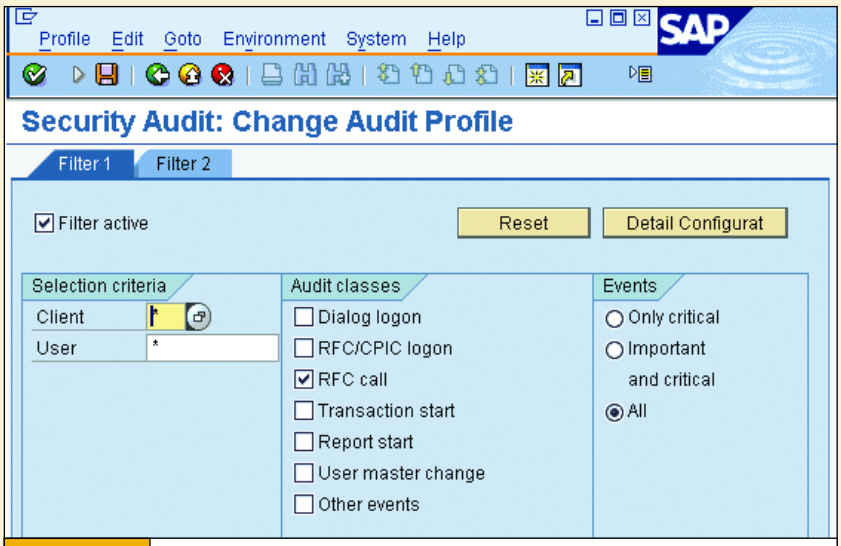


Figure 1 Security Audit Log: Configuration Setting to Analyze RFC Calls

[1] The RFC Software Development Kit, or RFC SDK, is an SAP plug-in for RFC. With RFC SDK's Remote Function Call API, users can remotely call ABAP function modules from C programs, as well as receive call requests from an ABAP program by the CALL FUNCTION interface.

the job of the SAP system to do all authorization checks, which the firewall isn't able to perform anymore. This means that authorizations, including those of service users if they are used to establish RFC connections, have to be configured very carefully.

Use Authorization Objects to Secure Your Remote Connections

Establishing a solid, secure authorization concept for RFC connections in your enterprise involves making these three authorization objects household names: S_ICF, S_RFCACL, and S_RFC. To ensure the security of your remote connections, confirm that these objects are all configured properly and are being used to their full advantage:

☑ S_ICF (available in SAP Web Application Server release 6.20 and higher)

If an SAP system wants to establish an RFC connection to another SAP system, S_ICF is carried out in the calling, or client, system to determine whether the user is allowed to call function modules using the RFC destination. This object could be found in the authorization profile of the user who is logged into the calling system. The user can use only those RFC destinations that are configured in the S_ICF object.

☑ S_RFCACL (available from SAP R/3 release 4.0)

Where trusted RFC is in place (see sidebar "Using RFC Trusted System Networks"), S_RFCACL is used on the server, or target system, to determine whether the user logged on to the client is permitted to access the server.

☑ S_RFC (available from SAP R/3 release 4.0)

Of course, it's not enough to restrict access on a system level. It's also important to define which function

Using RFC Trusted System Networks

To mitigate the risk of storing logon information in an RFC destination, it is possible to establish a trust relationship between the calling system (system A in **Figure 2**) and the destination (system B). In this case, the destination accepts connections from dedicated calling systems, known as trusted systems. No password is needed to access the destination from the trusted system. Since it is also possible to map individual users and their authorizations between the two systems, it is therefore easy to control the user's authorizations in the trusting target system. Unlike situations where service users are used, log files here will provide information about exactly which user accessed the target system, and what actions they performed there.

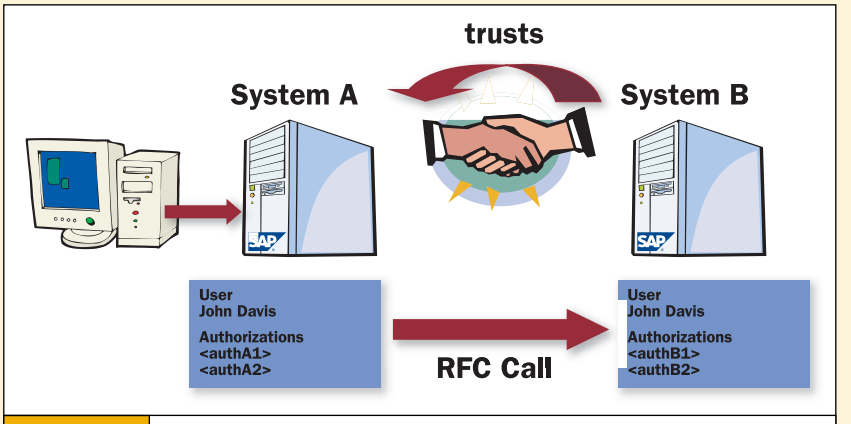


Figure 2 An RFC Trusted System Network

Using trust relationships in RFC connections, however, involves both opportunities and risks. In a trusted RFC relationship, the RFC destinations trust the user management and administration of the calling system, so that users only need to authenticate themselves once when they communicate with trusting systems. If you have sound user management in place — where users are only assigned rights they actually need and haven't been given far-reaching authorizations they don't need — you can increase the level of security for your RFC connections by doing away with the need to store passwords in the RFC destination. By performing authorization checks and producing change documents under the user ID of the calling user in the RFC destination, customers are able to control and audit the RFC connections effectively.

> ✔ **Note!**
> A prerequisite for successfully using a trusted relationship is that the user has the corresponding authorization object S_RFCACL in the trusting server system.

It is again imperative, however, that the relevant authorization objects are properly configured and managed to prevent abuse of this trust relationship. If, for example, you establish a trust relationship in a test system where anyone can create users, and most users have far-reaching authorizations, these rights would be transferred to the trusting system, opening your system up to potential abuse and outside attacks.

modules can be accessed remotely. This is where the authorization object S_RFC comes into play. S_RFC controls which specific RFC functions in the target system can be executed by a service user or individual user. It is the most important authorization object for protecting the target system because, while S_ICF works only on the client system and S_RFCACL is used only in the case of trusted RFC, S_RFC is *always* used, protecting the server system at all times.

To restrict access at this level, customers need to be familiar with their SAP system landscape and know what RFC destinations are available within it. Normally, the "service" type of user should be used for service users, although the "communication" type is also possible. To determine which RFC-related authorizations are needed for each user, customers can activate the Security Audit Log (see sidebar back on page 82).

For easy reference, **Figure 3** offers a quick overview of these three essential authorization objects and their availability.

Authorization Object	Description	For Use When...	Availability
S_ICF	Authorization check in the client system used to determine whether the logged-on user is permitted to use the RFC destination to call function modules by RFC	You want to connect an SAP system to another SAP system	SAP Web AS release 6.20 and higher
S_RFCACL	Authorization check in the server system used to determine whether the logged-on user in the client system can log on to the server system with the desired user ID	Trusted RFC systems are in place	SAP R/3 release 4.0 and higher
S_RFC	Authorization check in the server system used to determine whether the user can execute the RFC function module in the target system	You need to determine which function modules can be accessed remotely	SAP R/3 release 4.0 and higher

Figure 3 Essential Authorization Objects for Secure RFC Connections

Beyond Secure RFC Connections

Over and above correctly configuring the RFC connections themselves, other security issues still remain. In an SAP client system, transactions and authorization concepts are in place to ensure that only the permitted users have access to that data. But when it comes to external (non-SAP) programs — a third-party identity management system, for example — they often store data and authentication information that corresponds to the RFC destinations. Since *anyone* could use this data to log on to a system, the data must be protected. A certain level of protection is normally provided by the operating system, where customers can ensure via configuration that only the RFC client is able to access the file. To control access to the registered RFC server programs, customers can edit the gateway security settings.

In addition, many J2EE applications use RFC connections to access servers in an SAP backend system. Most RFC destinations are available centrally in the J2EE destination service. Calls can be made both by individual users, with a mechanism similar to trusted RFC and based on SAP Logon Tickets, and by service users. In both cases, to avoid errant users logging on to your systems, it is still essential for customers to implement tight authorizations for S_RFC as described earlier.

Conclusion

Particularly for those customers who have been relying on RFC connections for some time and are gradually opening up their systems — for example, if you're upgrading to SAP Web Application Server 6.20 or newer — it's vital to review the traditional security measures in place. It may be that your old standbys are no longer strong enough, given the changing nature of IT processes and system interaction. Your company information is just too important to risk to outdated security practices.

A good place to start is by checking how robust the current RFC authorization concept is, using the steps we've outlined in this article. For more detailed information, see the presentation "Tips and Tricks for Setting Up an Authorization Concept to Secure the RFC Connections in an SAP System Landscape" at SAP's Service Marketplace (**www.service.sap.com**). ■

Sarah Maidstone has been a security product manager since 2002, and speaks regularly on security at SAP conferences. Sarah has seven years of experience in various roles at SAP and holds an MA(Hons) degree in English Language and German.

Frank Buchholz joined SAP in 1994. With a strong focus on security, he worked in HR quality management before participating in the development of Secure Network Communications and the Audit Information System. After assuming the development lead role for maintaining and improving user and authorization management functions (ABAP), Frank joined the SAP NetWeaver Product Management Security team as Security Architect in the fall of 2003.