Securing Remote Function Calls (RFC)

SAP Security Recommendations
# Table of Contents

<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Introduction</td>
</tr>
<tr>
<td>7</td>
<td>Securing RFC Destination Configuration</td>
</tr>
<tr>
<td>9</td>
<td>Trusted System Security</td>
</tr>
<tr>
<td>11</td>
<td>Secure Network Communication</td>
</tr>
<tr>
<td>12</td>
<td>Securing RFC Communication on the Server</td>
</tr>
<tr>
<td>14</td>
<td>Limiting Access to RFC Function Modules</td>
</tr>
<tr>
<td>16</td>
<td>Authorization Maintenance for RFC Communication</td>
</tr>
<tr>
<td>19</td>
<td>Activating Switchable Authorization Checks</td>
</tr>
<tr>
<td>22</td>
<td>Securing RFC Communication on the Client</td>
</tr>
<tr>
<td>23</td>
<td>Securing RFC Callback</td>
</tr>
<tr>
<td>26</td>
<td>Securing the RFC Gateway</td>
</tr>
<tr>
<td>26</td>
<td>Access Control for External RFC Servers</td>
</tr>
<tr>
<td>30</td>
<td>Access Control for RFC Proxy Requests</td>
</tr>
<tr>
<td>32</td>
<td>Blocking RFC Communication</td>
</tr>
<tr>
<td>33</td>
<td>RFC Security Monitoring</td>
</tr>
<tr>
<td>34</td>
<td>Summary</td>
</tr>
<tr>
<td>35</td>
<td>Appendix</td>
</tr>
</tbody>
</table>
Remote function call (RFC) is an SAP-proprietary communication protocol used by computer systems, including those running the ABAP® version of the SAP NetWeaver® Application Server component. Most SAP customers run business-critical system communication using RFC technology, with thousands of RFC function modules accessible over the network. Keeping business data that is processed through RFC secure is as important to SAP and its customers as ensuring uninterrupted business operations.
Securing Remote Function Calls

Introduction

Remote function call (RFC) is a communication protocol proprietary to SAP. The network service that manages RFC communication is provided by the RFC gateway. Thousands of RFC function modules are included in SAP software systems. SAP technologies such as the BAPI programming interface and the application link enabling (ALE) interface technology use RFC as the underlying communication protocol. RFC communication is required for all software systems using the ABAP version of SAP NetWeaver Application Server (SAP NetWeaver AS) and is the most powerful and most frequently used integration technology for these software systems.

In many cases the RFC client and RFC server are implemented using the ABAP programming language. ABAP software serves as both RFC client and RFC server, depending on the integration scenario. The RFC client initiates the communication using connection data stored in RFC destinations to access RFC servers. RFC function modules are executed on the RFC server side. RFC clients are often referred to as the calling system and RFC servers as the called system.

External RFC clients and external RFC servers can be implemented using connectors available for different programming languages. Those connectors include the SAP Java Connector, SAP Connector for Microsoft .NET, and the SAP NetWeaver RFC software development kit (SAP NetWeaver RFC SDK), as well as the classic SAP RFC SDK. RFC communication can use Secure Network Communication (SNC) to encrypt communication. Figure 1 depicts the different clients and servers.

Figure 1: RFC Communication

<table>
<thead>
<tr>
<th>SAP NetWeaver® Application Server for ABAP® RFC client</th>
<th>SAP NetWeaver Application Server for ABAP RFC server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>RFC destinations</td>
</tr>
<tr>
<td>Program</td>
<td>RFC destinations</td>
</tr>
</tbody>
</table>

Registered external RFC server

| RFC function module |

External RFC client

<table>
<thead>
<tr>
<th>Program</th>
<th>RFC destinations</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ABAP runtime environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC function module</td>
</tr>
<tr>
<td>RFC function module</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Started external RFC server</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC function module</td>
</tr>
</tbody>
</table>

© 2014 SAP SE or an SAP affiliate company. All rights reserved.
RFC function modules provide, for example, functionality to create or change business partners and contracts, to post financial documents to the general ledger, and much more. The tremendous business functionality exposed through RFC function modules is a huge asset for system integration but also constitutes a large attack surface that must be protected from unauthorized access.

This document focuses on the secure configuration, secure authorization management, and process of monitoring RFC security. Its main focus is on software that uses SAP NetWeaver AS for ABAP 7.0 and higher. It provides an overview of the aspects that must be considered in order to ensure the required access control for all RFC communication. Included are recommendations that should be considered for SAP software running at the customer site.

Most SAP customers run business-critical system communication using RFC technology, with thousands of RFC function modules accessible over the network.

This document does not describe every implementation detail of the controls mentioned. However, it does reference detailed documentation and provides links to that documentation in the appendix.

The baseline for secure configuration of SAP NetWeaver AS for ABAP is covered in an earlier SAP white paper. It includes topics like network filtering, password management, and management of security patches. These are prerequisites for the RFC security controls described in this document. The baseline document includes sections covering aspects of RFC security. This paper expands on those sections.

This document discusses the following aspects concerning the protection of RFC communication:

- The “Securing RFC Destination Configuration” section covers guidelines that need to be considered for a secure setup. RFC destinations store connection information and often include user credentials to access RFC server systems. They also define whether communication is encrypted or not.
- The “Securing RFC Communication on the Server” section focuses on RFC function modules implemented in the ABAP programming language. Access is protected through user logon and authorization checks, but due to the large number of RFC function modules, maintaining user authorizations is a complex task. How to block remote access to RFC function modules not used in a software system is described. Improved solutions to perform RFC authorization management are mentioned. And how to activate switchable authorization checks that were added in RFC function modules is explained.
The “Securing RFC Communication on the Client” section provides information on securing RFC communication outbound from ABAP-based software and going to RFC function modules in other software systems. It explains how to control who is authorized to use RFC destinations that contain user credentials.

The “Securing RFC Callback” section discusses how to protect systems from malicious RFC callback. RFC function modules running in the RFC server of synchronous RFC connections can call back to the RFC client. The RFC server performs user logon and authorization checks. The RFC client that receives the callback does not perform user logon. The RFC callback runs under the privileges of the user that initiated the RFC communication in the RFC client. Authorization checks are performed on that user.

The “Securing the RFC Gateway” section covers access control for registered and started external RFC servers developed by SAP, partners, or customers using the SAP Java Connector, the SAP Connector for Microsoft .NET, or the SAP NetWeaver RFC software development kit. They typically do not perform user authentication or authorization checks. Access must be protected through access control lists. This section also explains access control for RFC proxy requests.

The “RFC Security Monitoring” section includes tips on monitoring RFC security.

RFC communication is required for all software systems based on SAP NetWeaver AS for ABAP. All customers should therefore follow the recommendations provided in this paper to protect their business data.

RFC is a communication protocol proprietary to SAP. The network service that manages RFC communication is provided by the RFC gateway.
Securing RFC Destination Configuration

RFC connections between software systems are configured using RFC destinations and maintained in transaction SM59 in RFC client systems. Inadequate management of RFC destinations could lead to privilege escalation – unintended elevation of user access. For example, SAP_ALL access in production systems could be gained using improperly configured RFC destinations in development systems. The following recommendations focus primarily on ABAP connections (type 3 RFC destinations in transaction SM59).

To manage RFC destinations securely, three different categories are considered:

1. Destinations storing a technical connectivity configuration without stored credentials and without trust relationships between the systems, which require user authentication for each access

2. Destinations with a technical connectivity configuration using stored credentials (such as client, user, and password)

3. Destinations with a technical connectivity configuration using trusted system logon (trusted/trusting RFC)

All three categories of RFC destination are recommended for use between software systems of the same security classification, for example, from one production system to another. They are also recommended for use from systems of higher security classification to systems of lower security classification, for example, from a test system to a development system (see Figure 2).

Access control for RFC callbacks is discussed for these destinations in the “Securing RFC Callback” section.

Figure 2: RFC Destinations

OK: RFC destinations from system of higher security classification to same or lower security classification

Check: RFC destinations of category 2 and 3 as security risks to be used only after risk analysis
As a general guideline, destinations from systems of lower security classification to systems of higher security classification should not store user credentials or use a trusted system logon. For example, a development system should not be a trusted system for a production system. These destinations are only to store technical connectivity configuration data and authenticate the user for each access. RFC destinations for the transport management system are exceptions to this general guideline as long as minimal authorizations are used for the users stored in these destinations. If other such destinations are required, they should be used only after performing a thorough risk analysis. The access control recommendations provided in this document could provide acceptable mitigations for some scenarios.

RFC destinations should be regularly reviewed to determine if they are required for running business processes. Each RFC destination should be assigned to an owner responsible for the destination and what each RFC destination is needed for should be documented. RFC destinations no longer required should be removed. To determine whether an RFC destination is used, an analysis can be performed using the RFC runtime monitor transaction SRTM or the workload monitor transaction ST03N.

Users stored in RFC destinations should be of type SYSTEM. Particularly in production environments, these users should be assigned the minimum authorization required for executing the business scenario from that destination in the target system. It is recommended that dedicated users be used per RFC destination wherever possible. The report RSRFCCHK can be used in all systems locally to analyze RFC destinations that have stored user names and passwords. For centralized monitoring, see “RFC Security Monitoring.”

It is strongly advised that authorization to maintain RFC destinations in transaction SM59 be restricted to authorized administrators. This is controlled primarily through authorization object S_RFC_ADM.

SECURITY RECOMMENDATIONS FROM SAP
- Assign owners to RFC destinations who are responsible for documenting the intended use and security requirements
- Create RFC destinations with stored user credentials or a trusted system logon only from systems of higher security classification to systems of the same or lower security classification
- Delete RFC destinations if they are no longer needed; remove system trust relationships if they are no longer needed (see “Trusted System Security”); use the report RSRFCCHK to list critical RFC destinations
- Restrict authorization to maintain RFC destinations in transaction SM59 by strictly controlling authorization object S_RFC_ADM

© 2014 SAP SE or an SAP affiliate company. All rights reserved.
TRUSTED SYSTEM SECURITY

ABAP-based software systems can trust other ABAP-based software systems. This makes it possible to set up a single sign-on for RFC and HTTP communication between trusted systems and trusting systems.9 Trust relationships are maintained in transaction SMT1 and can be used in transaction SM59 in ABAP connections and in HTTP connections to ABAP-based software systems.

Insecure use of trust relationships could allow malicious users to impersonate other users in remote systems without the need to know their password. This could be accomplished by changing the user name in RFC destinations using a trusted system logon. To maintain trust relationships securely, the following aspects must be considered:

• Maintenance of trust relationships in transaction SMT1 must follow strict guidelines, and authorization for maintenance must be restricted.
• The latest trusted system authentication methods should be used for all trust relationships.
• Authorization in the trusting system that controls the trusted system logon must be restricted.

It was already mentioned in the previous section that trusted systems must have the same or higher security classification as trusting systems. Trust relationships should be reviewed regularly in all business-critical systems, and trust relationships must be removed if they do not conform to this rule. The same rule applies for systems that can issue SAP logon tickets (configured in transaction STRUST). The authorization to maintain trust relationships in SMT1 is controlled using authorization objects S_RFC_TT and S_ADMI_FCD in systems based on SAP NetWeaver AS for ABAP 7.0 and higher.10 Earlier systems use object S_ADMI_FCD.

The security of the authentication mechanism that is used for the trusted system logon was improved with SAP Note 1491645.11 All trust relationships should be updated to use the improved authentication mechanism. The trusted and trusting systems require corrections to their kernel and ABAP code contained in SAP Note 1491645 before the trust relationship can be updated. The reports RS_SECURITY_TRUST_RELATIONS and RS_UPDATE_TRUST_RELATIONS should be used to identify and update trust relationships that use the deprecated authentication method.12 The database table RFCSYSACL holds technical data on these trusted system connections.

Inadequate management of RFC destinations could lead to privilege escalation – unintended elevation of user access.
Access to that table should be restricted (SAP Note 1562697). The table should be assigned to table authorization group TTRL. This can be checked and maintained using transaction SE54. Users should not have S_TABU_DIS or S_TABU_NAM authorization to display or change the data of that table.

The authorization for trusted system logon must be limited in the trusting system using authorization object S_RFCACL. There are two typical use cases for using trusted system logon:

- Often users have the same user name in the trusted and trusting system. This is controlled using the S_RFCACL field for the same user (field RFC_EQUSER = Y). The S_RFCACL field for the trusted user name should be left blank when logging in under the same user name (field RFC_USER = ' ').
- In some cases users in the trusted system should be allowed to log on with a different user name in the trusting system. The S_RFCACL field for the same user documents the use of different user names (field RFC_EQUSER = N).

The S_RFCACL field for the trusted user name contains the allowed user name from the trusted system (field RFC_USER = USERNAME).

The field for the user name should never contain * (the wild card). Any user from the trusted system could impersonate a user having that authorization. Wild cards should also not be used for the fields holding the system ID and the client of the trusted system (fields RFC_SYSID, RFC_CLIENT).

A warning is displayed during role maintenance in transaction PFCG if administrators maintain a wild card for any of these fields. The installation number of the trusted system can be included in the S_RFCACL authorization as of SAP NetWeaver AS for ABAP 7.0 that includes enhancement package 2 (field RFC_INFO = 1234567890).

If a trusted system logon fails, a short dump is created in the trusting system.

Authorization object S_RFCACL is not included in the authorization profile SAP_ALL unless the default configuration is changed in the customizing table PRGN_CUST with key ADD_S_RFCACL. This default configuration should not be changed.

SECURITY RECOMMENDATIONS FROM SAP

- Remove trust relationships in transaction SMT1 if trusted systems have a lower security classification than trusting systems
- Renew trust relationships using outdated authentication methods with reports RS_SECURITY_TRUST_RELATIONS and RS_UPDATE_TRUST_RELATIONS
- Assign table RFCSYSACL to table authorization group TTRL; do not grant display authorizations for that table to end users
- Strictly control S_RFCACL authorizations in trusting systems; do not include the wild card * for S_RFCACL fields RFC_USER, RFC_SYSID, RFC_CLIENT, or RFC_EQUSER
- Do not add the authorization object S_RFCACL to the authorization profile SAP_ALL
SECURE NETWORK COMMUNICATION
RFC does not authenticate client and server cryptographically, nor does it encrypt network communication. Passwords transmitted over the network can be “sniffed.” Due to missing mutual authentication, rogue systems could intercept network traffic, manipulate content, and forward it to legitimate servers (“man in the middle” attacks).

Secure network communication (SNC) provides cryptographically strong mutual authentication, integrity protection of transmitted data, and encryption of network traffic. Its use is highly recommended for all RFC communication traversing untrusted networks to mitigate these risks.

SNC for RFC communication between SAP servers is available to all SAP NetWeaver customers. RFC communication between client and server should be protected using an SNC product, such as the SAP Single Sign-On application (see Figure 3). Blocking unencrypted communication is to be considered after implementing SNC.

Strict access control to transaction STRUST is required in order to protect access to cryptographic keys when using SAP Single Sign-On. The table SSF_PSE_D should be assigned to table authorization group SPSE. End users are not to have display or change access for this table authorization group. File system access from ABAP programs to personal security environment (PSE) files holding cryptographic keys should be restricted.

SECURITY RECOMMENDATIONS FROM SAP
• Implement SNC for RFC communication between end-user networks and server networks when you implement SNC for SAP GUI communication
• Implement SNC for RFC communication between different server networks
• Restrict access to cryptographic keys through transaction STRUST, table SSL_PSE_D, and PSE files
Securing RFC Communication on the Server

Recent releases of SAP ERP application software expose over 40,000 RFC function modules to the network. In a typical customer software system, however, only a few hundred RFC function modules need to be called remotely.

Remote access to RFC function modules is protected by user logon and at least the technical authorization check performed through the authorization object S_RFC. Additional functional authorization checks are performed by many RFC function modules. Wild card authorizations for object S_RFC are very risky since the check on object S_RFC is the main access control for RFC function modules.

The profile parameter auth/rfc_authority_check controls technical authorization checks on object S_RFC. Do not use the value 0 because it deactivates S_RFC authorization checks. The default value 1 is recommended for most systems. It requires user authentication and at least S_RFC authorization for any remote access to RFC function modules on the server from other systems, other clients, or other users. Exceptions are RFC function modules in function group SRFC, which can be accessed anonymously. Additional profile parameter values are available to further restrict access to that function group.

S_RFC authorization checks (see Figure 4) are also performed for internal RFC when profile parameter auth/rfc_authority_check is set to value 9. An internal RFC is used for asynchronous processing and for dynamic load distribution to other work processes or application servers. Internal RFC does not change the user; it maintains the same user context in the same system and the same client. SAP does not recommend this setting for most customer software systems. It requires that you include in S_RFC authorizations RFC function modules that are used internally in the system but which should not be accessed remotely. It also complicates authorization maintenance for object S_RFC.

---

**Figure 4: S_RFC Checks Using auth/rfc_authority_check with Values Between 1 and 8**

<table>
<thead>
<tr>
<th>RFC</th>
<th>Function Call</th>
<th>S_RFC Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Remote call</td>
<td>Is performed</td>
</tr>
<tr>
<td></td>
<td>Internal call</td>
<td>Is not performed</td>
</tr>
<tr>
<td>No</td>
<td>Local call</td>
<td></td>
</tr>
</tbody>
</table>

---
Each function module is assigned to a function group. The authorization check on object S_RFC provides access control on two levels of granularity as of SAP NetWeaver AS for ABAP 7.0 with enhancement package 2. Systems running SAP enhancement package 1 for SAP NetWeaver 7.0 and lower perform checks on the function group level only. Access is granted to execute all RFC function modules assigned to function groups included in the S_RFC authorization. The kernel correction provided in SAP Note 1785761 should be implemented in these systems.

Systems using SAP enhancement package 2 for SAP NetWeaver 7.0 and higher perform authorization checks on the function module level as well. Most roles delivered by SAP and created by customers use access control on the function group level. The “Authorization Maintenance for RFC Communication” section covers how to maintain access on the RFC function module level.

The system trace transaction ST01 has been used for many years for role maintenance for RFC communication. This approach has proven to be difficult for many customers. Many RFC scenarios use highly privileged technical users. Also end users often have S_RFC wild card authorization, allowing them to call all RFC function modules in a system. A cleanup of these authorizations using ST01 traces is difficult.

Improved solutions are available to simplify access control of RFC communication on the server. They are covered in the remainder of this section.

Destinations from systems of lower security classification to systems of higher security classification should not store user credentials.

SECURITY RECOMMENDATIONS FROM SAP

- Use profile parameter auth/rfc_authority_check with value 1 for most systems. Do not use the value 0, as this deactivates the S_RFC authorization check. The value 9 should not be used either, since it requires including RFC function modules in S_RFC authorizations that are only called through internal RFC and not remotely.
- Do not grant wild card authorizations for authorization object S_RFC to users.
LIMITING ACCESS TO RFC FUNCTION MODULES

The unified connectivity (UCON) RFC basic scenario for security is a function of SAP NetWeaver AS for ABAP 7.4 that simplifies remote access control to most RFC function modules in ABAP-based software. Typically less than 5% of all available RFC function modules are used in customer software systems for remote RFC communication. More than 95% can usually be blocked from remote access. System-internal RFC communication for asynchronous processing and load distribution is not restricted by UCON.

UCON protects remote access to RFC function modules by adding a new layer of access control that is independent of authorization checks. When remote users call RFC function modules in a system protected by UCON, the framework checks if the RFC function module is included in a white list. This white list is called the UCON default communication assembly (UCON default CA). If the RFC function module is not included in the UCON default CA, access is denied and the RFC session is terminated with an error message.

Client-dependent activation of UCON runtime checks can be performed. UCON is configured through transaction UCONCOCKPIT (or transaction UCONPHTL in systems released prior to support package 7).

If access is granted by the UCON default CA, an authorization check on object S_RFC is performed as usual. UCON provides first-line protection that is maintained independent of users and roles. Authorization checks on object S_RFC remain the second line of protection. In addition, functional authorization checks are performed during the execution of many RFC function modules.

Without tool support it would be difficult to identify all RFC function modules that need to be remotely accessible. UCON provides a long-term system monitoring and learning mechanism to assist customers in building the UCON default CA. The process is divided into three phases (see Figure 5).

Figure 5: Phases of the UCON RFC Basic Scenario for Security

Logging
Log the RFC function modules called remotely

Evaluation
Simulate UCON runtime checks

Final
Active UCON runtime checks

UCON = Unified connectivity
The first phase is the logging phase. Data is collected without disrupting RFC communication. It is recommended that the UCON data collection be activated as early as possible. The default logging phase is three months, but this can be adjusted to customer needs. It should cover all typical RFC communication, month-end, quarter-end, year-end, and system management activities. Most customers have test systems where most of the RFC integration scenarios can be tested. However, there are some RFC connections that are only used in production systems. For that reason, it is also advisable to activate the UCON data collection in production systems and to use that data to maintain the UCON default CA. All RFC function modules called remotely in the logging phase need to be evaluated, and a decision must be made as to whether these RFC function modules should be added to the UCON default CA. Additional information on the RFC scenario is available, which supports the analysis. For every RFC function module, it can be determined from which RFC client system it was called, which user initiated the RFC communication, over which RFC destination the communication took place, and which user executed the RFC function module within the RFC server. It can also be determined when the RFC function was first called remotely, when it was last called, and how often. After the logging phase, the UCON default CA should include all RFC function modules that will be called remotely.

The second phase is the evaluation phase, which simulates the UCON runtime checks. Calls to RFC function modules that have not been white-listed are allowed but they trigger alerts. Administrators review the alerts and assign the RFC function modules to the UCON default CA if appropriate.

Runtime checks are active in the third phase, which is the final phase. Access to RFC function modules that have not been white-listed is blocked regardless of authorization.

With UCON, it is possible to manage every system individually or to transport UCON default CA and phase information in software system landscapes from development systems to production systems. New RFC function modules developed by customers or created through support packages after finalizing the initial implementation of UCON are assigned to the logging phase. Further information on implementation is available in “UCON RFC Basic Scenario – Guide to Setup and Operations.”

SECURITY RECOMMENDATIONS FROM SAP

• Perform the initial setup for the unified connectivity (UCON) RFC basic scenario for security as early as possible when planning UCON implementation. The earlier this is done, the better the quality of data will be. Set the duration for the logging and evaluation phase to cover all running RFC scenarios. Follow the guide “UCON RFC Basic Scenario – Guide to Setup and Operations.”
• Use the UCON RFC basic scenario for security for business-critical production systems. Typically, remote access to more than 95% of all RFC function modules can be deactivated in these systems.
AUTHORIZATION MAINTENANCE FOR RFC COMMUNICATION

Remote access to RFC function modules in ABAP-based software systems is protected by S_RFC authorization checks. Additional functional authorization checks are performed by many RFC function modules.

To build roles for RFC communication, the following information is needed:
- Which RFC function modules were executed by which RFC server users must be determined.
- Authorizations that are programmatically checked in the RFC function modules must be identified and documented.

The transaction codes to build roles for RFC communication are shown in Figure 6.

Wild card authorizations for object S_RFC are very risky.

The UCONCOCKPIT transaction is available as of SAP NetWeaver AS for ABAP 7.4. All other transactions are available as of SAP NetWeaver AS for ABAP 7.0. The SAP Notes referenced provide information on system requirements.

Figure 6: Transactions Used for Role Maintenance for RFC Communication

<table>
<thead>
<tr>
<th>Activity</th>
<th>Analyze</th>
<th>Build</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activity</strong></td>
<td>Monitor RFC function modules called by user</td>
<td>Document authorization checks by RFC function module</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td>Monitor authorization checks by RFC function module</td>
<td>Build roles</td>
</tr>
<tr>
<td>STAUTHTRACE</td>
<td>STUSOBTRACE</td>
<td>SU24</td>
</tr>
<tr>
<td>UCONCOCKPIT</td>
<td>STFCTRACE</td>
<td>PFCG</td>
</tr>
</tbody>
</table>
Short-Term Analysis of Specific RFC Communication

Short-term analysis of authorization requirements for specific RFC communication scenarios is performed using the transaction STAUTHTRACE (SAP Note 1718059). The analysis through transaction ST01 has been deprecated. The trace includes authorization checks for every RFC function module instead of function groups if S_RFC authorization of the analyzed user is set to function module level authorization (field RFC_TYPE = FUNC). A systemwide trace should be activated in the transaction STAUTHTRACE if the RFC communication uses load balancing (SAP Note 1707841). The trace results show all RFC function modules that were called remotely as well as all authorization checks that were performed during the execution of these RFC function modules. Additional information on authorization requirements for RFC communication using external RFC clients created using connectors is available in SAP Note 460089.

The trace results of the transaction STAUTHTRACE can be used in transaction SU24 to maintain authorization default values for RFC function modules (SAP Note 1707841). This will document which authorization is required to execute which RFC function module. This information can be reused for all roles that include these RFC function modules. This concept has traditionally been used for roles granting access to transactions. SAP provides authorization default values for transactions in transaction SU22. Customers maintain authorization default values in transaction SU24. Initial setup and consolidation of SAP and customer data after system upgrade is done in transaction SU25. The same process of maintaining authorization proposals can also be used for RFC function modules as well as ALE scenarios (SAP Note 1868691). Additional information on this is available in the online documentation.

The authorization default values maintained in transaction SU24 can be used for role maintenance in transaction PFCG if they are part of the role menu. To accomplish this, RFC function modules included in the trace results of STAUTHTRACE must be added to the role menu (SAP Note 1631929). The required S_RFC authorizations are then automatically calculated and all authorization defaults for the RFC function modules are added to the role (SAP Note 1640733). As a result, all authorizations in such roles have either the status “standard” or “maintained” and they reference the RFC function modules that require that authorization. This makes it unnecessary to add authorizations manually. Manually added authorizations do not reference the RFC function module that requires them, which means tracing must be performed again the next time that RFC function module is used in other RFC scenarios.

The short-term system trace analysis can be used to maintain roles for specific RFC scenarios. Additional tools are available for long-term analysis to reduce privileges that have grown excessive over time.

Long-Term Monitoring of All RFC Communication in a System

A long-term authorization trace can be activated using the profile parameter auth/authorization_trace in test and production systems. It is configured using transaction STUSOBTRACE (SAP Note 1854561) and based on the same process used by SAP to maintain authorization default
Securing Remote Function Calls

values (SAP Note 543164). The authorization trace records all programmed authorization checks once per application. The authorization trace can be filtered, for example, to record only authorization checks performed for remote RFC calls (SAP Note 1847663). The trace results are used in transaction SU24 to maintain authorization default values for RFC function modules.

The long-term authorization trace in transaction STUSOBTRACE documents authorization requirements for RFC function modules. It does not document which users require access to these function modules. This information is available in transaction UCONCOCKPIT described in the previous section.

Systems without UCON can extract which users require access to which function modules from the statistical data collected by default in all ABAP-based systems. The kernel version from SAP Note 1964997 should be used in these systems, and the profile parameter stat/rfc/distinct should be set to 1 in order to record all RFC function module names at least once per RFC communication. The statistical data is presented in transaction ST03N. A custom program can be used to extract RFC calls from the statistical data. This data includes external RFC calls and internal RFC calls, so it must be filtered for the external RFC calls. Systems with SAP Note 2080378 can use transaction STRFCTRACE to analyze the remote RFC communication in a similar way to that provided by UCON in later releases.

Transactions STAUTHTRACE and STUSOBTRACE are often used in test systems or production systems to analyze authorization checks. Transactions SU24 and PFCG are used in development systems in order to maintain authorization defaults and roles, which are transported to test and production systems. Transactions SU24 and PFCG make it possible to analyze trace data from remote systems (SAP Note 1861370). An RFC destination without stored user credentials can be used for this.

SECURITY RECOMMENDATIONS FROM SAP

- Use transaction STAUTHTRACE for short-term analysis of authorization checks in specific RFC scenarios
- Set profile parameter auth/authorization_trace to Y or F for a long-term authorization trace in test and productive systems; configure the trace in transaction STUSOBTRACE
- Use the trace results to document authorization defaults for remotely called RFC function modules in transaction SU24 in the development system
- Analyze users or groups of users that require access to certain RFC function modules using transaction UCONCOCKPIT, STRFCTRACE, STAUTHTRACE, or a custom report, depending on system release
- Build roles in transaction PFCG with authorization defaults of RFC function modules in the role menu to calculate S_RFC authorization automatically; include authorization defaults for all RFC function modules
ACTIVATING SWITCHABLE AUTHORIZATION CHECKS

Many RFC function modules can be sufficiently protected by the technical S_RFC check, but often they perform no additional functional authorization checks in their program code. Other RFC function modules require additional functional authorization checks to be executed securely. If it is determined that a required check is missing, new authorization checks will have to be implemented in existing RFC function modules.

Authorization checks that are newly introduced in existing RFC function modules through SAP Notes or through support packages can interrupt business-critical system communication if legitimate users do not have the newly introduced authorization.

To enable a nondisruptive evolution of authorization checks, SAP introduced switchable authorization checks in all software systems based on SAP NetWeaver AS for ABAP 7.0 and higher. It is handled by the switchable authorization check framework (SACF) and is configured in transaction SACF.

All switchable authorization checks are delivered inactive to ensure that business operations remain undisrupted after support packages or SAP Notes have been technically implemented. Switchable authorization checks use so-called authorization scenario definitions that describe the check. The authorization scenario definition references documentation for the check and lists all authorization objects that are checked. Switchable authorization checks in the ABAP program code can use the same authorization scenario in different places in the code. All checks using the same scenario are switched at once. Authorization scenario definitions created by SAP and the programmed checks are delivered through SAP Notes, support packages, or enhancement packages.

At runtime, the checks require a so-called productive authorization scenario to be active. Productive authorization scenarios are created from authorization scenario definitions. They hold the settings that determine whether the checks will be actively performed or if they are only logged for analysis. For logging purposes the check results are written to the security audit log. The events DUO and DUP log successful and unsuccessful switchable
authorization checks. The event DUQ logs changes to productive authorization scenarios. The security audit log should be active, and these events should be included in an active static filter for all users in all clients in order to enable the logging. More information on security audit log settings is available in SAP Note 539404. Analysis of the security audit log events in the transaction SM20 or in the report RSAU_SELECT_EVENTS makes it possible to adjust the roles of affected users before the productive authorization scenario is set to active. The earlier the logging is activated, the better it is for the data quality.

Productive authorization scenarios are never delivered by SAP. They are always created in customer development systems and transported in the customer software system landscape. If partners or customers need to implement switchable authorization checks, they can implement authorization scenario definitions in the customer name space. Figure 7 shows how authorization scenario definitions, productive authorization scenarios, and programmed switchable authorization checks based on these scenarios work.

**Figure 7: Switchable Authorization Checks**

SAP, partner, or customer development system

<table>
<thead>
<tr>
<th>SAP, partner, or customer development system</th>
<th>Customer development system</th>
<th>Test and production systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>SACF</td>
<td>SACF</td>
<td>SACF</td>
</tr>
<tr>
<td>Scenario definition</td>
<td>Scenario definition</td>
<td>Switchable authorization check framework</td>
</tr>
<tr>
<td>Development</td>
<td>Configuration</td>
<td>Switchable authorization check in ABAP code</td>
</tr>
<tr>
<td>Workbench for ABAP®</td>
<td>Runtime</td>
<td>Switchable check defined (inactive)</td>
</tr>
<tr>
<td>Switchable authorization check in ABAP® code</td>
<td></td>
<td>Developers create scenario definitions and implement switchable checks.</td>
</tr>
</tbody>
</table>

Switchable check defined (inactive)

**Developers** create scenario definitions and implement switchable checks.

Switchable check configured (logging only, active check, inactive)

**Administrators** create productive scenarios that are used for authorization checks at runtime.

SACF = Switchable authorization check framework
SACF integrates seamlessly into the authorization upgrade activities in transaction SU25 in systems based on SAP NetWeaver AS for ABAP 7.3 with enhancement package 1.49 Transaction SACF_COMPARE can be used in all other systems in standard maintenance to configure multiple authorization scenarios after the software is installed or upgraded. With transaction SACF, authorization scenarios are maintained individually.

The switchable authorization check framework is typically installed through support packages. SAP Note 205452250 lists all other required SAP Notes if installation is to be performed through individual SAP Notes. Security notes from SAP indicate in the note title that they provide new switchable authorization checks. Switchable authorization checks are also used by SAP for functionality unrelated to RFC. Additional information on SACF is available in SAP Note 1922808.51 Changes to the authorization concept are typically not part of the regular patch management and must be planned. The security notes providing new switchable authorization checks in RFC function modules should be reviewed together with S_RFC authorizations in one project and revisited for upgrades and new system installations.

At runtime, the checks require a so-called productive authorization scenario to be active.

SECURITY RECOMMENDATIONS FROM SAP

• Activate the security audit log using profile parameter rsau/enable = 1. In transaction SM19, include events DUO, DUP, and DUQ in an active static filter for all users in all clients. Add additional filters by increasing the value of profile parameter rsau/selection_slots, if required.
• Combine S_RFC authorization refinement and activation of switchable authorization checks in RFC function modules in one project.
• Use transaction SACF_COMPARE after new system installation to create active productive authorization scenarios for all definitions. This activates all switchable authorization checks.
• Use transaction SACF_COMPARE after upgrades to create productive authorization scenarios with logging status for all new definitions. The status logging supports analysis if the scenario is used and indicates which users are affected. Activate the productive scenario after the logging phase, if required. Set it to inactive if it is not required. Analyze productive authorization scenarios that were changed by the upgrade.
Securing RFC Communication on the Client

RFC destinations stored in RFC client systems often contain logon data for RFC server systems. The RFC destinations that should contain no logon data were mentioned in the “Securing RFC Destination Configuration” section.

All users in the client system use the same RFC user stored in the RFC destination to execute RFC function modules in the server. They do not need to have the authorizations in the server themselves, nor do they require a user in the RFC server system, strictly speaking.

Misuse of RFC destinations with stored credentials of highly privileged users is known as RFC hopping. Insecure setup of RFC destinations could allow access to be made to a development system, which could enable access to a test system, and finally to a business-critical production system.

Access control for RFC communication on the RFC client side is provided by authorization objects S_ICF and S_DEVELOP. RFC destinations in RFC client systems with stored access credentials for business-critical systems should be categorized in authorization groups. This is maintained for each RFC destination in transaction SM59 in the “Logon & Security” tab. RFC destinations can be assigned to these authorization groups. In the RFC client system, a technical authorization check on authorization object S_ICF is then performed by the RFC runtime environment at the kernel level. It verifies that the user in the RFC client system is authorized to use RFC destinations of a given authorization group. By default, RFC destinations are not assigned to authorization groups. The authorization check on object S_ICF does not take place until an authorization group is assigned. Systems running central user administration (CUA) can, for example, assign RFC connections to managed systems to an authorization group. Authorizations to use these RFC destinations can be limited to CUA administrators.

RFC destinations are configured per system and can be used by users in any client of the system if they are not properly protected by authorization groups. One example is using RFC destinations in transaction SE37 for test execution of function modules. Transaction SE37 can start function modules through an RFC destination or in the same system and client locally. During test execution of function modules, an authorization check is performed on authorization object S_DEVELOP. An additional check on authorization object S_ICF is performed if the test execution uses an RFC destination that is assigned to an authorization group. Test execution should be strictly controlled in systems that have RFC destinations with stored access credentials to business-critical systems. Access to transaction SE37 should generally not be granted to end users in production systems. Emergency procedures should exist for support processes that may require test execution.

SECURITY RECOMMENDATIONS FROM SAP

- Configure authorization groups for RFC destinations to business-critical systems; this results in S_ICF authorization checks made for users that use them
- Strictly control S_DEVELOP authorization for test execution of function modules in production systems and in all systems with RFC destinations storing access credentials of business-critical systems
Securing RFC Callback

So far this document has covered RFC communication, where RFC clients execute functionality in RFC servers but not the other way around. However, the server can perform an RFC callback to the client during synchronous RFC calls. It uses the open RFC connection to the RFC client and does not require another user logon. This is done using the predefined destination BACK55 which exists in all ABAP-based software systems. RFC callback can also be performed by external RFC servers created with any connector or received by external RFC clients created with the SAP NetWeaver RFC SDK. RFC callback executes RFC function modules on the client in the context of the user that initiated the synchronous RFC communication. All RFC function modules that this user is authorized to execute in the RFC client can be executed by the server through RFC callback. Asynchronous RFC, transactional RFC, queued RFC, and background RFC do not support RFC callback (see Figure 8).

RFC callback can pose risks to business-critical systems when initiating RFC communication using highly privileged users to other systems with a lower trust level. For example, batch jobs are in many cases executed by highly privileged system users. These batch jobs could perform RFC communication to remote systems. Malicious remote systems could misuse the high privileges of the batch user using the RFC callback. For this reason, it is advised that the following access control be implemented for all business-critical systems.

RFC callback always performs S_RFC authorization checks and potentially additional functional authorization checks on the user that initiated the RFC communication. The authorization management for users initiating RFC communication should therefore follow the same guidelines as outlined in the “Authorization Maintenance for RFC Communication” section on authorization management.

Figure 8: RFC Callback

[Diagram showing the process of RFC callback]
RFC callback white lists improve access control for RFC callback. They are maintained for RFC destinations in systems that initiate RFC communication and which might receive RFC callbacks. The white lists are maintained for RFC destinations of connection types ABAP and TCP/IP. One white list for the RFC destination `DYNAMIC_DEST_CALLBACK_WHITELIST` can be maintained for dynamic RFC destinations. For many RFC destinations, RFC callback is not needed. If RFC callback is needed, only a few RFC function modules are typically called back. White-list entries define tuples of RFC function modules called in the target of an RFC destination and RFC function modules that are allowed to be called back. A learning and simulation phase supports maintenance of RFC callback white lists for existing RFC destinations. It is recommended that white lists be configured at least for RFC destinations in business-critical systems when connecting to systems of lower security classification. Logging of RFC callbacks should be activated in all systems to provide traceability.

RFC callback white lists are provided in SAP Note 1686632. The checks are controlled through the profile parameter `rfc/callback_security_method`, which has by default the value 1. This setting is nondisruptive as long as there is no active RFC callback white list and it can be used to log RFC callbacks to the security audit log. The security audit log should be active and events DUI, DUJ, and DUK should be included in an active static filter for all users in all clients to enable the logging. The events DUI and DUJ provide information of which RFC callbacks were executed successfully and which were rejected. The event DUK is used for the simulation phase. It logs RFC callbacks that would be rejected after activation of inactive white lists.
RFC callback white lists are maintained in transaction SM59 in systems based on SAP NetWeaver AS for ABAP 7.3 with enhancement package 1 with SAP Note 1686632. The logged RFC callbacks stored in the security audit log are analyzed programmatically and RFC callback white lists are generated from the logs. The white list is then displayed in the “Logon & Security” tab of RFC destinations. SAP Note 2058946 explains the maintenance of RFC callback white lists in other systems in standard maintenance.

Creating an RFC callback white list is performed in three phases:

- In the logging phase, the profile parameter rfc/callback_security_method is set to 1. Security audit log is configured and DUI events are written for all successful RFC callbacks. The RFC callback white lists are created using these logs in transaction SM59 or as explained in SAP Note 2058946. The white lists are not yet set to active.
- In the simulation phase, the profile parameter rfc/callback_security_method is set to 2 and inactive white lists are tested. RFC callback that is not covered by an inactive white list is alerted using the security audit log event DUK. All these events must be reviewed and added to the white list if appropriate.
- After the simulation phase, RFC callback white lists are activated. RFC callbacks that are not included in the white lists are rejected after activation. DUJ events are written to the security audit log for rejected callbacks.

The profile parameter rfc/callback_security_method value 3 can be used once white lists have been fully maintained for all RFC destinations of connection types ABAP and TCP/IP. With this parameter setting, all active and inactive white lists are actively evaluated.

In addition to using white lists, RFC callback can be prevented programmatically before initiating synchronous RFC communication. This is done by calling function module RFC_CALLBACK_REJECTED locally in the RFC client before performing the RFC call. A programmatically rejected RFC callback always has precedence over the white list.

SECURITY RECOMMENDATIONS FROM SAP

- Implement SAP Note 1686632 in all systems
- Activate the security audit log and include events DUI, DUJ, and DUK in an active static filter for all clients and users in all systems
- Do not set the profile parameter rfc/callback_security_method to 0 in business-critical systems, as this deactivates logging of RFC callback and active white-list checks; use parameter value 1 or 2 as long as there are RFC destinations with inactive white lists
- Create and activate RFC callback white lists for RFC destinations in business-critical systems when connecting to systems of lower security classification
Securing the RFC Gateway

ACCESS CONTROL FOR EXTERNAL RFC SERVERS

The “Securing RFC Communication on the Server” section focused on controlling access to RFC function modules implemented in the ABAP programming language. This requires a logon to the ABAP-based software system and at the least S_RFC authorization.

As mentioned earlier, external RFC servers can be implemented using connectors such as the SAP NetWeaver RFC SDK. Those external RFC servers can register at an RFC gateway or they can be started by an RFC gateway. The RFC gateway is the technical component of the application server that runs the network services managing all RFC communication. By default, these external RFC servers provide RFC function modules to all RFC clients that can access the RFC gateway.

External RFC servers typically do not have any user management. They could identify RFC clients using SNC, but in most cases they do not perform any authentication and, for that reason, cannot perform authorization checks. The access control for registered and started RFC servers is provided by access control lists (ACLs) managed by the RFC gateway. Figure 9 shows the different types of RFC servers – RFC function modules implemented in ABAP as well as registered or started external RFC servers.

Figure 9: ABAP RFC Function Modules Versus Started and Registered RFC Servers
RFC gateway ACLs for external RFC servers should be maintained for all systems based on SAP NetWeaver AS for ABAP and SAP NetWeaver AS for Java as well as stand-alone RFC gateway installations. If RFC gateway access control is not maintained, malicious RFC clients could start external RFC servers without a logon and execute operating system commands on the server, download files from the server, manipulate files on the server, or access the database of the application server. Malicious external RFC servers could register at remote RFC gateways to receive RFC calls intended for legitimate external RFC servers or to perform a malicious RFC callback to the RFC client.

There are two RFC gateway access control lists that must be maintained for external RFC servers. Maintaining these ACL files has no impact on RFC communication to RFC function modules in the ABAP server.

- ACL file `reginfo` controls the registration of external RFC servers and the accessing of registered external RFC servers.
- ACL file `secinfo` controls the starting of external RFC servers.

Registered external RFC servers are used far more often than started external RFC servers. These RFC servers often run in their own environment and register at a remote RFC gateway. They keep the connection to the RFC gateway open. RFC clients access these servers through this open connection. Very often the only RFC client is the ABAP-based software system where the external RFC server is registered. This is configured in transaction `SM59` in TCP/IP connections with the technical setting “Registered Server Program.” One example for this use case is the search and classification engine TREX of the SAP NetWeaver technology platform. Registered external RFC servers are a very common integration technology and are being developed by SAP and partner companies.

Started external RFC servers are executables on the server and are started by the RFC gateway upon request by RFC clients. Typically RFC clients are the application servers of a system. External RFC servers are not started in most systems by remote RFC clients. One example is the start of the external RFC server `SAPXPG`, which is used through transaction `SM49` to execute operating system commands on the same application server. Accessing started external RFC servers is configured in transaction `SM59` in TCP/IP connections with the technical setting “Start on Explicit Host” and RFC gateway options that point to a remote RFC gateway.

The access control for registered and started RFC servers is provided by access control lists (ACLs) managed by the RFC gateway.
RFC gateway ACL files `secinfo` and `reginfo` do not exist after system installation. Unless the profile parameter `gw/acl_mode` is set to 1, there are no restrictions to register or start external RFC servers. New system installations add this setting to the default profile (DEFAULT.PFL). It restricts registration and starting external RFC servers to the application servers of a system. ACL files must be adjusted in case additional remote connections are required.

Systems that were installed before `gw/acl_mode` was introduced must be configured to ensure security. A system upgrade does not automatically enable the settings. Guidelines for maintaining ACL files `secinfo` and `reginfo` are available on the SAP Help Portal site and in SAP Note 1408081. A tool for the initial ACL setup is available through transaction SMGW (menu Goto, Expert Functions, External Security).

Secinfo ACL entries generated by the tool for started external RFC servers block remote RFC clients. Only internal access from application servers of the same system is granted. In most customer systems, there are no other ACL entries needed. RFC gateway logging should be enabled at least for rejected RFC clients.

On the other hand, registered external RFC servers typically register from remote systems. The tool evaluates all external RFC servers that are registered on RFC gateways of all application servers of the current system. It also evaluates all local TCP/IP connections that access registered RFC servers. It adds these external RFC servers to the ACL file `reginfo` and grants access to register from remote systems. Most registered external RFC servers are always registered, and they restore the registration if the connection is lost. RFC gateway logging should be enabled and used to identify registered external RFC servers that are not identified at initial setup. The logs need to be reviewed regularly, and ACL files need to be adjusted if necessary.

The generated ACL controls registration of external RFC servers. It does not limit access of remote RFC clients to registered RFC servers. The ABAP-based software system where the external RFC server is registered is in many cases the only legitimate RFC client accessing that registered RFC server. RFC client access is logged for registered external RFC servers, with `reginfo` entries containing ACCESS restrictions. Again, the RFC gateway logging can be used to identify RFC clients accessing registered RFC servers.

Gateway logging actions `S` and `s` are used for starting, registering, or accessing external RFC servers.

Uninterrupted business operation is as important as security. ACL simulation is available to test the ACL settings in a nondisruptive way before activating them. The simulation mode is available as of version 7.21 of the kernel of SAP NetWeaver AS for ABAP and is configured through the profile parameter `gw/sim_mode` (SAP Note 1689663). Entries that are missing in the ACL are not rejected but logged using an RFC gateway logging action `Z`.

After initial generation, ACL changes are performed directly in the files. ACL changes do not require a system restart. They can be activated dynamically either for individual application servers or globally for all application servers in transaction SMGW.
In summary, the following profile parameters should be set:

- **gw/logging** (SAP Note 910919) should be activated. Log files of multiple systems with multiple application servers can be stored on a network share when using variables like $(SAPSYSTEMNAME) and $(SAPLOCALHOST) in the file names. The logging action should include actions SsMPXZ. External RFC server communication is logged using action S if simulation is off and action Z if simulation is on. Logging is limited to rejected requests if the lowercase s is included in the logging actions.

- **gw/acl_mode** (SAP Note 1480644) should be set to 1 for security by default in case of nonexistent ACL files.

- **gw/sec_info** and **gw/reg_info** (SAP Note 1408081) hold the path to the ACL files. It is recommended that those settings be changed to a path on the global directory that can be accessed by all application servers. This eliminates the need to maintain the files for all application servers individually.

- **gw/reg_no_conn_info** (SAP Note 1444282) must be set to 1 or higher odd numbers explained in the SAP Note. This ensures that bypass of ACL files as described in SAP Note 1298433 is not possible.

- **gw/sim_mode** (SAP Note 1689663) can temporarily be set to 1 to test ACL files, but should not be used indefinitely. The simulation mode does not deny access, but it logs access that is not included in the ACL using logging action Z. The log files must be analyzed and missing entries added to the ACLs.

- **gw/monitor** (SAP Note 64016) should be set to 1 to restrict RFC gateway monitoring to local access (gw/monitor = 1). This is the default configuration setting and should not be changed to 2 for remote access.

Once RFC gateway access control is configured and active, it should be monitored in order to make sure that restrictions remain in place. Monitoring can be performed using the SAP EarlyWatch Alert service, which includes a section on RFC gateway security settings as explained in SAP Note 863362. In addition, the configuration validation functionality of the SAP Solution Manager application management solution can be used to monitor the RFC gateway security settings. This is covered in the “RFC Security Monitoring” section.

**SECURITY RECOMMENDATIONS FROM SAP**

- Activate RFC gateway logging in all systems to provide an audit trail of access to external RFC servers
- Generate RFC gateway ACL files secinfo and reginfo for all systems
- Use the simulation mode to test ACL settings nondisruptively; evaluate the RFC gateway log frequently in the simulation phase; add missing ACL entries, if required; activate the ACL settings by turning off simulation
- Monitor secure settings using SAP EarlyWatch Alert or the configuration validation functionality of SAP Solution Manager
ACCESS CONTROL FOR RFC PROXY REQUESTS
RFC gateways can proxy requests to RFC function modules in other ABAP-based software systems. Requests to registered or started RFC servers cannot use RFC gateway in proxy mode. Using an intermediate RFC gateway as proxy can be configured on the RFC client side in transaction SM59 for all ABAP connections by setting RFC gateway options in the technical settings. Also, external RFC clients created with the SAP Java Connector or the classic SAP RFC SDK can use an intermediate RFC gateway as proxy to call RFC function modules in other ABAP-based software systems.

Many customers segment networks and restrict RFC communication between networks using firewalls. For example, the RFC gateway of an SAP ERP software system might be accessible to the end-user network. The SAP ERP software could have RFC connections to other ABAP-based software systems. Access to those systems might be blocked from the end-user network by a firewall. Malicious users could try to access the other ABAP system by using the RFC gateway of the SAP ERP software system in proxy mode.

Proxy requests should be restricted if RFC communication is filtered between network segments. Most RFC gateways do not have to proxy RFC requests from external RFC clients.

Access control for RFC proxy requests is configured in the ACL file prxyinfo. The file location is set using profile parameter gw/prxy_info. A single ACL entry is sufficient for most customer systems:

```
P SOURCE=local.internal DEST=*
```

This configuration permits all application servers of a system to proxy requests of other application servers of the same system. All remote proxy requests are blocked.

The ACL settings in file prxyinfo can be tested using the simulation mode that is configured through profile parameter gw/sim_mode. It logs proxy requests that are not included in prxyinfo instead of rejecting them.

RFC gateways can proxy requests to RFC function modules in other ABAP-based software systems.
In summary, the following profile parameters should be set:

- **gw/logging** (SAP Note 910919) should be activated in the same way as explained in the “Access Control for External RFC Servers” section. Logging actions S and Z are also used to log proxy requests. Logging can be limited to rejected proxy requests using lowercase s.
- **gw/prxy_info** (SAP Note 910918) holds the path to the ACL file. A global directory accessible to all application servers allows the use of a single file for a system.
- **gw/reg_no_conn_info** (SAP Note 1444282) must be set to 1 or higher odd numbers. To restrict proxy requests from external RFC clients, the value of the profile parameter must be increased by 128 as explained in SAP Note 1848930. The parameter value 129 (1 + 128) enables protection against bypass of ACL files (SAP Note 1298433) and ACL checks for proxy requests of external RFC clients.
- **gw/sim_mode** (SAP Note 1689663) can be set temporarily to 1 in order to test the ACL file prxyinfo. Proxy requests are logged using action Z in RFC gateway logging instead of rejecting requests. gw/sim_mode applies to secinfo, reginfo, and prxyinfo as well.

### SECURITY RECOMMENDATIONS FROM SAP

- Activate RFC gateway logging in all systems to provide an audit trail of proxy requests
- Restrict proxy requests in all systems that can connect to other firewalled systems; allow internal proxy requests from application servers of the same system; proxy requests of remote RFC client systems are not needed in most systems
- Use simulation mode to test ACL settings nondisruptively; evaluate the RFC gateway log frequently in the simulation phase; add missing ACL entries, if required

Proxy requests should be restricted if RFC communication is filtered between network segments.
BLOCKING RFC COMMUNICATION
The RFC gateway access control lists secinfo, reginfo, and prxyinfo restrict certain aspects of RFC communication. They are required for secure system operation in all ABAP-based software systems and systems with a stand-alone RFC gateway. In some systems, however, it might be necessary to completely block all RFC communication from certain clients or networks.

This type of general access control to network services is typically performed using firewalls. It can also be configured in the RFC gateway using the access control list configured with profile parameter gw/acl_file. Clients and networks in this access control list can be explicitly denied or allowed access for RFC communication through that RFC gateway.

Testing this ACL file is not covered by the RFC gateway simulation mode.

SECURITY RECOMMENDATIONS FROM SAP
• Use firewalls to restrict access to network services provided by the RFC gateway to legitimate RFC clients
• Use the RFC gateway ACL gw/acl_file to supplement firewall settings and block RFC communication from certain clients or networks

RFC does not authenticate client and server cryptographically, nor does it encrypt network communication.
RFC Security Monitoring

RFC destinations can be analyzed in every system using transaction SM59 or report RSRFCCHK. However, complex software system landscapes typically have hundreds of RFC destinations in dozens of systems. Monitoring individual RFC destinations in each system is effort intensive. However, central RFC security monitoring functionality can provide an overview of the security status of the entire RFC integration landscape.

Functionality for monitoring RFC destination settings is available through the configuration validation functionality of SAP Solution Manager. The monitoring function periodically collects all RFC destination details of managed systems through diagnostic agents. Relevant system profile parameters and access control lists are also collected. Predefined RFC security templates are available, which can be adjusted to customer needs. The data from managed systems is compared against the template, and noncompliant settings are identified and reported in dashboards.

Configuration validation functionality is part of the work center for root cause analysis of SAP Solution Manager. The following checks should be used to monitor RFC security:

- RFCDES_TYPE_3_CHECK checks ABAP connections to see if user names and passwords are stored in the destination, if a trusted system logon is used, which authorizations RFC users have in target systems, and whether secure network communication is used.
- ABAP_INSTANCE_PAHI verifies required profile parameter settings.
- GW_REGINFO and GW_SECINFO checks RFC gateway access control lists.

An overview of the current security status can be provided in a security dashboard and alerts on noncompliance can be collected in the alert in-box.

RFC destinations should be regularly reviewed to determine if they are required for running business processes.

SECURITY RECOMMENDATIONS FROM SAP

Use the configuration validation functionality of SAP Solution Manager to monitor RFC security centrally.
Summary

Well-configured RFC access control is essential to maintain the security of business data in systems based on SAP NetWeaver AS for ABAP. SAP advises all customers with these software systems to determine whether the security recommendations provided in this document have been implemented in their software system landscapes and to take action in cases where access controls are missing.

Several sections discuss the configuration for logging RFC communication. These logs serve as audit trails and should be activated at least in all business-critical production systems. The logging data is analyzed by tools, which simplifies the work of maintaining access controls. The earlier logging is activated, the better the data quality for these tools. Some controls provide simulation modes to test controls nondisruptively before they are activated.

Central monitoring tools provide customers with insight into the current RFC security status and help keep required access controls in place over time.

If you have questions concerning any material discussed in this document or would like to request RFC security consulting, please use the contact information provided in SAP Note 1682316.
Appendix

1. SAP Help Portal site – Ports of SAP NetWeaver Application Server for ABAP
   https://help.sap.com/saphelp_nw74/helpdata/en/4e/c26cdc58e968b9e100000000a42189e/frameset.htm
2. SAP Service Marketplace extranet – Connectors: Communication between SAP Systems and other SAP or non-SAP Systems
   https://service.sap.com/connectors
3. SAP Community Network – Secure Configuration of SAP NetWeaver Application Server Using ABAP
   https://scn.sap.com/docs/DOC-17149
4. SAP Service Marketplace – White papers providing security recommendations from SAP
   https://service.sap.com/securitywp
5. SAP Help Portal – CTS User Administration and Authentication
   https://help.sap.com/saphelp_nw74/helpdata/en/6b0d99f34d11d3a65100000e835363f/frameset.htm
6. SAP Note 1150764 – Where-used list for RFC destinations
   https://service.sap.com/sap/support/notes/1150764
7. SAP Note 622464 – Change: Password change req. entry for “SYSTEM” user type
   https://service.sap.com/sap/support/notes/622464
8. SAP Help Portal – Authorization Object S_RFC_ADM
   https://help.sap.com/saphelp_nw74/helpdata/en/48/8d1c05ae444e6ee100000000a421937/frameset.htm
9. SAP Help Portal – Maintaining Trust Relationships between SAP Systems
   https://help.sap.com/sap/helions/48/956eb194cc73e9e100000000a42189b/frameset.htm
10. SAP Help Portal – Authorization Object S_RFC_TT
11. SAP Note 1491645 – Unauthenticated system access via RFC or HTTP
    https://service.sap.com/sap/support/notes/1491645
12. SAP Note 1498973 – Renewing trust relationships to a system
    https://service.sap.com/sap/support/notes/1498973
13. SAP Note 1562697 – Authorization group for trust relationship
    https://service.sap.com/sap/support/notes/1562697
14. SAP Help Portal – Authorization Object S_RFCACL
15. SAP Note 128447 – Trusted/trusting systems
    https://service.sap.com/sap/support/notes/128447
16. SAP Note 1416085 – PFCG: Authorization maintenance for object S_RFCACL
    https://service.sap.com/sap/support/notes/1416085
17. SAP Help Portal – Configuring SNC: Using RFC from AS ABAP
    https://help.sap.com/saphelp_nw74/helpdata/en/b2/cadb21ecd949d0a1ff84ff18e4032b/frameset.htm
18. SAP Knowledge Base Article 2057374 – Using SNC Client Encryption (SCE) for encrypting SAP GUI Connection
    https://service.sap.com/sap/support/notes/2057374
19. SAP Help Portal – SAP NetWeaver Single Sign-On 2.0
    https://help.sap.com/sso
20. SAP Note 1690662 – Option: Blocking unencrypted SAPGUI/RFC connections
    https://service.sap.com/sap/support/notes/1690662
21. SAP Note 1485029 – Protect read access to key tables
    https://service.sap.com/sap/support/notes/1485029
22. SAP Note 1497104 – Protect access to PSE files by additional AUTHORITY-CHECK
    https://service.sap.com/sap/support/notes/1497104
23. SAP Note 931252 – Security Note: Authority Check for Function Group SRFC
    https://service.sap.com/sap/support/notes/931252
24. SAP Note 1769064 – Additional values for auth/rfc_authority_check
    https://service.sap.com/sap/support/notes/1769064

© 2014 SAP SE or an SAP affiliate company. All rights reserved.
Securing Remote Function Calls

25. SAP Note 931251 – Security Note: Authority Check for Function Modules
   https://service.sap.com/sap/support/notes/931251
26. SAP Note 1785761 – Missing authorization check in RFC
   https://service.sap.com/sap/support/notes/1785761
27. SAP Community Network – Unified Connectivity (UCON)
   https://scn.sap.com/docs/DOC-53844
28. SAP Help Portal – Unified Connectivity
   https://help.sap.com/saphelp_nw74/helpdata/en/ab/35e1c69f744d69a4fcf4ca93284e0c/frameset.htm
29. SAP Community Network – SAP Insider: Secure Your System Communications with Unified Connectivity
   https://scn.sap.com/docs/DOC-51003
30. SAP Note 1904562 – Unified Connectivity Logging Prerequisites
   https://service.sap.com/sap/support/notes/1904562
31. SAP Community Network – UCON RFC Basic Scenario – Guide to Setup and Operations
   https://scn.sap.com/docs/DOC-57565
32. SAP Help Portal – Creating an Authorization Concept for RFC
33. SAP Note 1718059 – New functions for the trace evaluation
   https://service.sap.com/sap/support/notes/1718059
34. SAP Note 1707841 – STAUTHTRACE: System-wide trace evaluation
   https://service.sap.com/sap/support/notes/1707841
35. SAP Note 460089 – Minimum authorization profile for external RFC programs
   https://service.sap.com/sap/support/notes/460089
36. SAP Note 1868691 – ALE: Creating authorizations for system users
   https://service.sap.com/sap/support/notes/1868691
37. SAP Help Portal – From the Programmed Authorization Check to a Role
   https://help.sap.com/saphelp_nw74/helpdata/en/fc/27cdec46fa4f77b8ac6b3d5169b72e/frameset.htm
38. SAP Note 1631929 – Using trace evaluation to maintain menus and authorizations
   https://service.sap.com/sap/support/notes/1631929
39. SAP Note 1640733 – PFCG: Additional S_RFC authorization
   https://service.sap.com/sap/support/notes/1640733
40. SAP Note 1854561 – Authorization trace with filter (auth/authorization_trace)
   https://service.sap.com/sap/support/notes/1854561
41. SAP Note 543164 – Significance of auth/authorization_trace values
   https://service.sap.com/sap/support/notes/543164
42. SAP Note 1847663 – STUSOBTRACE: Filter for authorization trace
   https://service.sap.com/sap/support/notes/1847663
43. SAP Note 1964997 – ST: Enhancement of kernel statistics for RFC subrecords
   https://service.sap.com/sap/support/notes/1964997
44. SAP Community Network – How to get RFC call traces to build authorizations for S_RFC for free!
   https://scn.sap.com/community/security/blog/2010/12/05/how-to-get-rfc-call-traces-to-build-authorizations-for-srfc-for-free
45. SAP Note 2080378 – STRFCTRACE
   https://service.sap.com/sap/support/notes/2080378
46. SAP Note 1861370 – PFCG: Menu maintenance via authorization trace
   https://service.sap.com/sap/support/notes/1861370
47. SAP Help Portal – Performing Authorization Checks Based on Scenarios
   https://help.sap.com/saphelp_nw74/helpdata/en/a9/a721a34a4b4e2fa5c12947022c7d76/frameset.htm
48. SAP Note 539404 – FAQ: Answers to questions about the Security Audit Log
   https://service.sap.com/sap/support/notes/539404
Securing Remote Function Calls

49. SAP Note 2068580 – SU25 | New comparison functions
   https://service.sap.com/sap/support/notes/2068580

50. SAP Note 2054522 – SP implementation dependency with BASIS (SACF) corrections
   https://service.sap.com/sap/support/notes/2054522

51. SAP Note 1922808 – SACF: FAQ – Supplementary application information
   https://service.sap.com/sap/support/notes/1922808

52. SAP Help Portal – Controlling Access to RFC Destinations

53. SAP Note 587410 – Test environment: Activity 16 (Execute) in S_DEVELOP
   https://service.sap.com/sap/support/notes/587410

54. SAP Help Portal – Central User Administration (CUA)
   https://help.sap.com/saphelp_nw74/helpdata/en/8d/270bea613d2443bad6ce0524f08ca0/frameset.htm

55. SAP Help Portal – RFC Destinations

56. SAP Note 1686632 – Positive lists for RFC callback
   https://service.sap.com/sap/support/notes/1686632

57. SAP Note 1968729 – SAL: Message definition for RFC callback
   https://service.sap.com/sap/support/notes/1968729

58. SAP Help Portal – Logon and Security

59. SAP Note 2058946 – Maintenance of callback positive lists before Release 7.31
   https://service.sap.com/sap/support/notes/2058946

60. SAP Note 1515925 – Preventing RFC callbacks during synchronous RFC
   https://service.sap.com/sap/support/notes/1515925

61. SAP Help Portal – Interfaces to External RFC Programs

62. SAP Help Portal – Security Settings in the Gateway

63. SAP Note 1408081 – Basic settings for reg_info and sec_info
   https://service.sap.com/sap/support/notes/1408081

64. SAP Note 1425765 – Generating sec_info reg_info
   https://service.sap.com/sap/support/notes/1425765

65. SAP Note 910919 – Setting up Gateway logging
   https://service.sap.com/sap/support/notes/910919

66. SAP Note 1480644 – gw/acl_mode versus gw/reg_no_conn_info
   https://service.sap.com/sap/support/notes/1480644

67. SAP Note 1444282 – gw/reg_no_conn_info settings
   https://service.sap.com/sap/support/notes/1444282

68. SAP Note 1298433 – Bypassing security in reginfo & secinfo
   https://service.sap.com/sap/support/notes/1298433

69. SAP Note 1689663 – GW: Simulation mode for reg, sec, and prxy_info
   https://service.sap.com/sap/support/notes/1689663

70. SAP Note 64016 – Using the SAP Gateway monitor GWMON
   https://service.sap.com/sap/support/notes/64016

71. SAP Note – 863362 – Security checks in SAP EarlyWatch Alert
   https://service.sap.com/sap/support/notes/863362

72. SAP Community Network – ConfVal_Security
Securing Remote Function Calls

73. SAP Note 910918 – GW: Parameter gw/prxy_info
   https://service.sap.com/sap/support/notes/910918

74. SAP Note 1848930 – GW: Strong gw/prxy_info check
   https://service.sap.com/sap/support/notes/1848930

75. SAP Note 1495075 – Access control lists (ACL)
   https://service.sap.com/sap/support/notes/1495075

76. SAP Help Portal – Configuration Validation

77. SAP Community Network – ConfVal_Security

78. SAP Help Portal – Configuring Security Apps
   https://help.sap.com/saphelp_sm71_sp12/helpdata/en/83/c485de06724172b2b45f635e49332e/frameset.htm

79. SAP Community Network – ConfVal_Alerting
   https://wiki.scn.sap.com/wiki/display/TechOps/ConfVal_Alerting

80. SAP Note 1682316 – Consulting: Optimizing RFC User Authorizations
    https://service.sap.com/sap/support/notes/1682316