Abstract

SAP security is still a dark world. Very little information can be found on the Net and almost every question related to security assessment of these applications remains unanswered. This paper has the intention of bringing some light into that world, providing the results of a security analysis performed over the SAP RFC interface implementation.

SAP RFC interface is the heart of communications between SAP systems, and between SAP and external software. Almost every system that wants to interact with SAP systems does so using the RFC interface. As stated by SAP: "The RFC library is the most commonly used and installed component of existing SAP software".

This paper describes vulnerabilities discovered in the RFC Library and their security impact. Furthermore, advanced attacks, exploiting default mis-configurations and design flaws in the interface implementation, are presented and explained. Finally, it provides solutions and suggested configurations to protect from described attacks and vulnerabilities..
Below, we provide an example of a simple RFC client program. For more complex and detailed examples,

2.2.1. External RFC Clients

Library. The vast majority of systems that connect with SAP R/3 components have this library installed,

```c
/* Connection error */
if (handle == RFC_HANDLE_NULL) {
    rfc_error_info);
    exception = NULL;
    RFC_INT number=1;
    CUST_NOT_FOUND = 0
    EXCEPTIONS
    DATA = cust_data
    TABLES
    PARCH = parch
    CHANGING
    ZCUST_ID = 100
    EXPORTING
    CALL FUNCTION 'ZCUST_GETMONEY' DESTINATION 'PROD2'
    CALL FUNCTION
```
Security parameters used to regulate the interaction with started and registered external servers are managed. Further on, we will see that remote access to this facility enables attackers obtaining highly valuable information for performing advanced attacks.

The Gateway Monitor access is regulated through the profile parameter.

Security of the Gateway is provided by different means: Work Process (for LU 6.2 communications with IBM mainframes) and the Gateway Monitor (for administration).
AUTORITY_CHECK_RFC function module.

These authentication and authorization procedures depend on the value of the

In SAP Application Servers, it is a different story. Authorization is based in checking whether the user calling RFC call will be processed.

To prevent credential and information sniffing, SAP has developed SNC (Secure Network Communications).

3.1. Traffic Analysis

3. Security Analysis of the SAP RFC Interface Implementation

In External systems, authentication and authorization tasks are responsibility of external software developers.

2.4. Authentication and Authorization Mechanisms

version 7.00) deployed in Microsoft Windows Server 2003, communicating with external clients and servers

The first (and obvious) thing you realize when starting at a network dump is that RFC communications are
An additional point that is worth mentioning is that developer traces (files automatically created) are used.

As described in the previous section, if you want to develop an external RFC server, you would use the RFC PING operation with a fixed key:

```
3.2. Authentication and Authorization Mechanisms and the SRFC Function Group
```

With this information, the possibility of obtaining valid credentials is just limited to the chance of intercepting an automatic authentication and authorization takes place for any function group different from SRFC),

```
•
```

System availability.
```
System Information.
```
LCHECK=0
```

for each CHAR in CLEAR_TEXT_PASS

```
OBFUSCATED_PASS[i] = CHAR
```

parameter in the connection string.
```
```

•
```
auth/
```

© 2007 CYBSEC Security Systems S.A. All rights Reserved.
This function enables the "controlled" remote execution of programs on external servers. The function works as an ACL to regulate RFC_PROGRAM access. To protect from abuse, SAP delivered the group accounts in an external server system, its domains and trusted domains. Developed for internal use by SAP only, this function can be abused to verify the existence of Windows user/3.3.4. RFC_TRUSTED_SYSTEM_SECURITY *

This function, present in both SAP AS and external servers, returns quite a lot of information about the server external RFC server. This function enables the definition of properties of external registered servers. Calling this function with the appropriate parameters, allows an external client to obtain exclusive use of the server. This clearly represents in the ability to execute remote arbitrary commands over the external server system.

3.3.6. RFC_START_GUI *

We have detected that many of these default functions can be abused to perform security sensitive operations over external RFC servers, with impact ranging from information disclosure to remote code execution.

Following, we describe the analyzed functions and the security caveats detected:

No

RfcAllowStartProgram(NULL)

RfcAllowStartProgram("cmd1.exe")

RfcAllowStartProgram()

= All commands are authorized.

RfcAllowStartProgram()

= All commands are authorized.

RfcAllowStartProgram()
We have discovered that some of these validations are also flawed. To make the situation worse, the default function only verifies that the first N bytes of the requested command matches the first N bytes of the command against allowed ones can be abused to execute other programs in the server system:

4.1. Evil Twin

4. Advanced Attacks

The RFC SDK is shipped with many examples. One of them is the function revealed that the process of validating the received program, which was originally in stealth mode, registers itself at the works as an
Just fine. Otherwise, defining the exclusive use of the original server would be necessary, which will allow to...

We will now explain how, the simple fact of being able to register an external RFC server may enable a

One problem with the above described attack, is that the normal flow of communication between the original...


8. Send results back to original client.

7. Attack receives (original) external server processing results.

4. As the server who registered first (the original) is busy with another connection, the call is forwarded to...

5. At this point the attacker is in control of client parameters and tables, being able to log or modify their...

2. Attacker registers an external server with Program ID = EXT1.

1. Attacker opens connection to server with Program ID = EXT1, blocking the connections of other clients.

6. Attacker uses established connection with (original) external server, forwarding the (possibly modified)...

Program ID belonging to two...

Program ID
Protection from these attacks is possible and must be implemented. We are still researching on this subject and effectively controlling the interaction with external servers through the request more information from the client to complete the process. In this case, the server performs a difference: it uses the already established connection with its partner. The roles are temporarily interchanged.

The described vulnerabilities in the RFC Library have been reported to SAP and patches are already available.

There was one situation that was overlooked in the introduction to the SAP RFC Interface: apart from normal attacker to take complete control of a SAP R/3 system. If the client is a SAP Application Server, the authentication method. If the user has SAP_ALL authorizations (or any other privileged roles), you can take complete control of the SAP Application Server.

The following steps describe the attack in detail:

1. Attacker registers an external server with Program ID = EXT1.
2. Attacker registers the callback destination Program ID.
3. Attacker performs a callback over the established connection. Depending on initial user authorizations, the RFC call executes under the privileges of the user who initiated the first call, bypassing any files. Finally, it is callback.