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..
As many legacy and internally developed systems may already be installed (and successfully working) before protocol for communication between SAP systems and between SAP systems and external (non-SAP) SAP R/3 is implemented in an organization, SAP must provide a way to communicate with these partners. The direct transfer of data between systems fitted well the basic requirements of data communication.

We shall now differentiate between two types of RFC connections: Connections between SAP systems and External systems.

In SAP Application Servers (SAP AS from now on), RFC services are implemented as Originally, SAP implemented IBM's CPI-C interface to communicate with other systems. This protocol enabled...
The vast majority of systems that connect with SAP R/3 components have this library installed, therefore, this type of connection can be composed of an external system (client) calling a function module in a SAP AS. In case a SAP AS wants to call a RFC function module implemented in an external server, very little changes.
These authentication and authorization procedures depend on the value of the

2.4. Authentication and Authorization Mechanisms
developed with the RFCSDK for Windows and Linux, versions 6.40 and 7.00.

clear-text. You can quickly identify logon information, parameter and table names and values, etc:
As previously described, authentication and authorization of RFC calls are based on the value of the

As described in the previous section, if you want to develop an external RFC server, you would use the RFC

Analyzing different traffic dumps we discovered that the obfuscation algorithm turned out to be a simple XOR

As commented in [6], there are some RFC functions which are installed by default in every external RFC

RFC communication (without SNC).

credentials can be obtained.

operation with a fixed key:

0x55, 0xc5, 0xe5, 0xd4, 0xb, 0xa0, 0xdd, 0xd6, 0xf5, 0x21, 0x32, 0xf, 0xe2, 0xcd, 0x68, 0x4f, 0x1a, 0x50,

3.2. Authentication and Authorization Mechanisms and the SRFC Function Group

identify the first two pieces in the above dump, while there seems to be no clue for the password. The reason

is that the password is obfuscated.

If the RFC call is performed by a program using SAP's RFC library, the program will first call the RFC PING

Library to enable your server to communicate with RFC client partners.
Analysis of this function revealed information disclosure and buffer overflow vulnerabilities, which would allow a buffer overflow vulnerability in the processing of received parameters was also detected in this function, with the same consequences as the above case.

function works as an ACL to regulate RFC START PROGRAM access:

To protect from abuse, SAP delivered the

Developed for internal use by SAP only, this function can be abused to verify the existence of Windows user/

RFC_TRUSTED_SYSTEM_SECURITY

This function, present in both SAP AS and external servers, returns quite a lot of information about the server

RFC_SYSTEM_INFO

This function enables the definition of properties of external registered servers. Calling this function with the

SYSTEM_CREATE_INSTANCE

default. Analysis of this function spotted a buffer overflow vulnerability which, if properly exploited, would result

To allow starting SAPGUI on Front-end systems, this function is also present in every external RFC server by

Calling this function, an attacker can obtain information about installed (accessible) RFC functions in an

RFC_GET_DOCU

This function can be used to analyze availability of RFC interfaces, both in SAP Application Servers and

RFC_PING

We have detected that many of these default functions can be abused to perform security sensitive operations

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

over external RFC servers, with impact ranging from information disclosure to remote code execution.

RfcAllowStartProgram(NULL)

RfcAllowStartProgram("cmd1.exe")

= Execution of "cmd1.exe" is authorized.
In this section, we are going to describe different attacks we have developed, which abuse default mis-

configuration of this file is to allow everything.

sniffing (if SNC is not used) or having remote access to the Gateway Monitor.

allowed command, where N is the length of the

following format:

• rfcexec.sec
• RFC_REMOTE_FILE
• RFC_MAIL
• RFC_RAISE_ERROR

Program ID

Registration

rfcexec

program, which was originally

rfcexec

works as an