RCE Administration and Security Guide

Build 9.1.0.0201911201302_SNAPSHOT

Table of Contents

List of Figures

Chapter 1. Preface

This document is an *early draft* of the RCE Administration and Security Guide which is currently focused on the security of cross-organization setups using the "SSH Uplink" feature. This feature is part of RCE 10.0.

Chapter 2. Security Properties of RCE Features

This section provides a concise overview of the security properties of various RCE features, especially those related to network activity, security, and information transmission. It is intended to support security reviews and decisions before deploying and using RCE in your organization. For a general overview of RCE's features and operation, please refer to the standard RCE User Guide (available at `https://rcenvironment.de/pages/documentation/documentation.html` for the latest 9.x release) and the RCE Website (`https://rcenvironment.de`).

2.1. Security Properties of RCE's Default Network Connections

The default network connections of RCE are designed for close collaboration within a working group, and are only intended to be used within trusted networks. If you require secure connections across the boundaries of trusted networks (especially across the internet), or collaborate with external partners, the encrypted and authenticated SSH Remote Access or SSH Uplink features should be used instead.

A default RCE installation does not open a port for accepting standard connections unless explicitly configured to do so.

For a general overview of RCE's default network approach, please refer to the standard RCE User Guide (available at `https://rcenvironment.de/pages/documentation/documentation.html` for the latest 9.x release).

If RCE is configured to accept incoming network connections, the port number(s) to be used can be freely chosen.

Incoming connections can be restricted to certain IPv4 addresses. However, these filters are static, so this feature is only useful for either restricting connections to localhost for local multi-instance setups, or setting up networks between few, static, and long-living nodes. Both setups are fairly untypical, but may be of use in certain situations.

TODO.

- Describe which features can be remotely accessed via the default network
- Describe the relation to the the tool authorization system

2.2. Security Properties of RCE's SSH Server Port

RCE offers a built-in SSH server port, which is disabled by default: An RCE installation does not open a port for accepting SSH connections unless explicitly configured to do so. This port is provided using the Java library Apache SSHD (`https://mina.apache.org/sshd-project/`).

Version of Library Apache SSHD

In RCE 9.x, the latest 1.x version (1.7.0) is being used. RCE 10.x will use the latest 2.x version (currently 2.2.0).

If RCE's SSH server port is enabled, the port number may be freely chosen.

The accounts used to log into this SSH port are completely independent of system accounts; RCE provides its own account management.

Account passwords are never saved in plaintext. For login verification, only salted BCrypt hashes are stored. SSH key files are supported.

For each SSH account, a single authorization role is selected, which defines which actions are permitted for this account (e.g. workflow monitoring).

These SSH accounts, each with its assigned role and its password hash or SSH public key string, are stored in a JSON file within the instance's "profile" directory The location of this profile directory can be customized.

Accounts can be added or removed by using a provided text mode UI, or by manually inserting or deleting entries in the JSON file.

Connecting to RCE's SSH port does not create or allow any TCP port forwardings; this feature of standard SSH is disabled.

Each RCE instance automatically creates its own SSH server key pair once the SSH port is enabled. The key data is stored in the RCE instance's profile directory.

Client-side strict host key checking is disabled, and a warning is logged when the server-side key has changed. As the SSH login only provides a first line of security, with the actual service security provided by the RCE authorization system, even a successful MITM attack would not have a significant impact. Additionally, without access to the real login credentials, an attacker would have to perform successful MITM attacks on all incoming connections to the relay server, effectively replacing it completely. Such a scenario is highly unlikely, especially because even on success, such an attack would not impact the security of the service authorization system itself (as noted above).

RCE's SSH port supports three modes of operation:

- a custom command shell for RCE administration commands; Unlike standard SSH, this feature never provides a direct system shell.
- the deprecated "SSH Remote Access" feature (available in RCE 8.x and 9.x; to be replaced in RCE 11.0);
- the upcoming "SSH Uplink" feature (to be released in RCE 10.0)

The latter SSH Uplink feature is the recommended mode for offering tools as services to users outside of your organization. Unlike "SSH Remote Access", this mode was specifically designed for this purpose. Further, a special SSH authorization role will be provided to restrict SSH accounts to this mode exclusively. Notably, this will also completely disable access to the interactive command shell.

2.3. Security Properties of RCE's Uplink Feature

This protocol was specifically designed to allow different organizations to provide tool execution services to each other. As this naturally involves creating network connections over the boundaries of organization's networks, security is the top priority its its design and implementation. For a general overview of its network approach, please see the "administration" section below.

From an administrator's point of view, the main novelty compared to the older "SSH Remote Access" feature is that it is designed to connect different organizations over a shared "relay" server that can be placed outside the organization's internal network. This eliminates the need to open any incoming network ports in the organization's firewall(s). Connections are only established from the internal network to the outside (e.g. the internet), but are never required in the opposite direction.

It is of course also possible to use this feature completely inside the organization's network, for example for securely providing tool execution services between different departments.

Although the current implementation is built on top of the SSH protocol, the actual Uplink protocol is not tied to it. SSH is used as the default transport mechanism as it provides well-tested encryption and login authorization. Technically, this mechanism could be replaced by any other that provides similar features.

For example, the Uplink protocol could be expanded/adapted to support TLS connections, which would provide support for CA-based server certificates.

TODO.

- outline what data is sent as part of the protocol, and where it is transmitted to
- describe interaction with RCE's Tool Integration
- describe the interaction with local/default networks, especially regarding tool forwarding

2.4. Security Properties of RCE Tool Integration

For general information about RCE's Tool Integration concepts and configuration, please refer to the RCE User Guide (available here for the latest 9.x release).

From a security perspective, the service-providing side of Tool Integration consists of a dedicated user (typically an administrator or engineer familiar with the tool) defining a static script that controls the tool's invocation. Typical tasks in this script are custom input pre-processing or conversion, invoking one or more command-line tools, and assembling the final (external) data output.

All inputs and outputs of the service representing the tool is explicitly defined as part of the Tool Integration setup. Inputs and outputs are standardized to data types like Integer, Float, String, File, Directory, Vector, Matrix, or Table.

Beyond input data and (optionally) tool properties, the caller of the tool has no means of influencing the tool's execution.

The tool script is executed by passing it to bash on Linux or cmd on Windows. (To reiterate: This script is static, and always defined by a dedicated user on the providing side, never by the caller.)

The only data that is implicitly published from a tool's execution is the standard and error output of the executing tools, as well as execution time data.

If this is undesired, standard shell/batch features can typically be used to suppress this output

TODO.

- Describe tool properties

2.5. Security Properties of RCE's Tool Authorization System

Tools can be assigned to any set of authorization groups, which makes them accessible to any user that is a member of that group.

This assignment is defined on the machine where this tool is "integrated" (ie, where it is defined to be actually run; see the "Tool Integration" section in the RCE User Guide for details).

There is no central authority for group membership. This is an intentional design decision to keep the authorization system flexible and lightweight.

This decentralized approach has, among other things, the benefit of removing administrative overhead when setting up groups between different organizations, for example in the typical use case of a research project with partners from different organizations.

Each group membership is defined by access to a secret 256 bit key. From a user's perspective, this key works like a password required to access a specific group. These group keys are imported into RCE instances as text strings, after which they are available on this instance until they are removed.

The text snippets used to give users access to a certain group should be transported in a secure way, for example via an intranet page with access control.

Users can create any number of authorization groups using their RCE clients.

Due to the decentralized nature of the authorization groups, there is no global password change mechanism, or a central way of revoking group memberships. If this is a concern, periodic group rotation should be used. This can be realized, for example, by using an intranet page with organization-specific access control where the current group access codes are posted.

The group keys that were created or imported into an RCE instance are stored in an encrypted form using the Secure Storage implementation of the Eclipse platform (`https://www.eclipse.org/`). Each user's secure RCE data is encrypted with a random 256 bit master key that is stored in the user's home directory. This way, even if a user accidentally shares his/her whole RCE profile directory to a public location, nobody else can gain access to the stored group keys.

RCE provides console commands for listing all active authorization groups and their assignments, which can be used for automated auditing.

Chapter 3. RCE Administration

3.1. SSH Uplink Networks

To exchange tool computation services between organizations, some sort of network connection must be established. As many organizations are (understandably) reluctant to open ports into their networks, the Uplink approach is based on connecting to a shared coordination/forwarding server called a "relay".

This relay server is typically placed outside of any organization's protected network, e.g. on a rented server or in the DMZ of one of the involved organizations.

Due to the exposed nature of this relay server, it is designed to be secure by default. There is only one way of connecting to it, which is the encrypted and authenticated SSH protocol. The protocol transmitted over SSH is designed to be concise and easily audited.

Development is focused on placing as little trust as possible into the relay server. Technical steps are being taken to limit what data can be monitored at the relay server.

Important: These features are not complete in preview versions of this feature! Only use release versions to pass sensitive data, and review the corresponding version of this security guide.

A typical Uplink setup between two or more organizations involves:

The Uplink relay server (a specially configured RCE instance) in a location that is accessible from all organization's networks.

This relay server opens a SSH port, with one or more sets of credentials for each organization. Typically, only a single login is needed for each project group within an organization; see below.

An Uplink SSH gateway node (again, a specially configured RCE instance) in each organization's network.

This node is the only one that actually establishes an SSH connection to the relay.

It is typically centrally administered, and not used for any other end-user work.

A benefit of this setup is that the SSH login credentials are only required to be present on that machine. Additionally, this reduces the number of SSH logins that must be configured on the server side, as it is effectively shared by all users with access to this (internal) SSH gateway.

This node opens one or more default RCE network connection ports towards the internal network. Access to this can be restricted using standard internal network policies as needed.

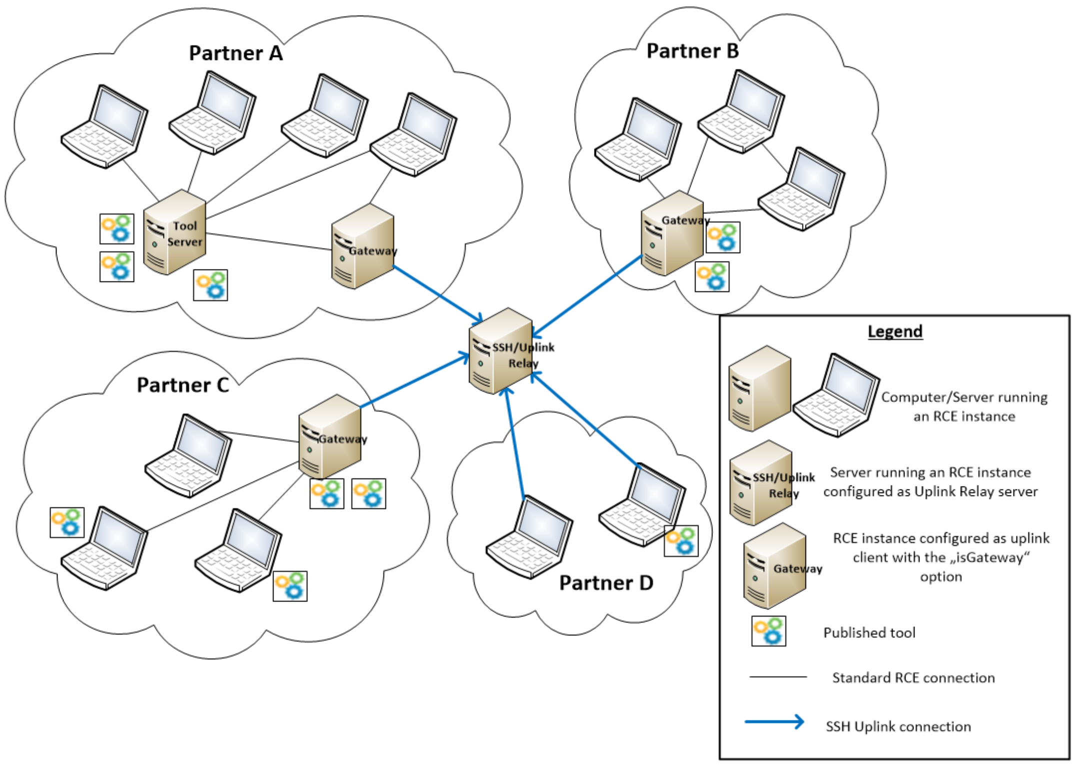
End users can then use their individual RCE instances to connect to the SSH gateway node (using RCE's default network features).

Once this setup is complete, users can access tool compute services published by their or one of the other organizations in the form of virtual tools within their RCE networks.

3.2. Example of a Cross-Organization Network

The following figure gives an example of how such a cross-organization network could be structured:

Figure 3.1. Example RCE network


Partner A
Tool Server
Gateway
Partner B
Gateway
SSH/Uplink Relay
Partner C
Gateway
Partner D
Legend
Computer/Server running an RCE instance
SSH/Uplink Relay
Server running an RCE instance configured as Uplink Relay server
Gateway
RCE instance configured as uplink client with the „isGateway" option
Published tool
Standard RCE connection
SSH Uplink connection

The four project partners in the example all have an internal network of RCE instances which are connected by standard RCE connections. Uplink connections to a relay server are used to connect between the different partners. The relay server is located outside of the organizations networks, and only the relay server has to be reachable via SSH over the internet. Typically, for each organization one RCE instance (called SSH gateway) established an SSH connections to this relay server. All other instances in the institution's internal network can be connected to it by standard RCE connections and still publish tools to the other partners/ access tools published by other partners.

Each institution in the example has a different internal setup, all of which are possible:

- Partner A has a dedicated RCE server where the published tools are located, which is connected to the SSH gateway by an RCE connection. All other RCE users in the internal network are connected to this server
- Partner B has put all the tools directly on the SSH gateway instance.
- In Partner C's network, some tools are located on the SSH gateway, but some tools are also published by users directly on their own machines. As long as they are connected to the SSH gateway also those tools can be published to the other partners.
- Partner D has no tool server, instead the users' computers connect directly to the relay server.

3.3. Notes and Recommendations: RCE's Default Network Connections

For default connections, an arbitrary number of incoming network ports can be configured.

3.4. Notes and Recommendations: RCE's SSH Port

When creating SSH accounts using the built-in administration text mode UI, there is an issue when entering passwords including the "@" character. This does not weaken security in any way, but can be confusing for end users when their correctly entered password does not work. It may be advisable to inform users about this.

Hashing SSH passwords with external BCrypt tools and entering them into the JSON file containing SSH accounts (either manually or via scripts) should theoretically work, but is not supported, and is therefore not known to work.

Chapter 4. Future Work and Outlook

Lorem ipsum dolor sit amet