

Table of Contents

UNICORE TSI Service Reference Card.....	1
Functional description.....	1
Daemons running.....	1
Init scripts and options (start stop restart ...).....	1
Configuration files location with example or template.....	1
Logfile locations (and management) and other useful audit information.....	1
Open ports.....	1
Possible unit test of the service.....	1
Where is service state held (and can it be rebuilt).....	1
Cron jobs.....	1
Security information.....	2
Access control Mechanism description (authentication & authorization).....	2
How to block/ban a user.....	2
Network Usage.....	2
Firewall configuration.....	2
Security recommendations.....	2
Security incompatibilities.....	2
List of externals (packages not from the OS).....	2
Other security relevant comments.....	2
Utility scripts.....	2

UNICORE TSI Service Reference Card

Functional description

The UNICORE TSI is a Perl daemon running on the frontend of the target resource (e.g. a cluster login node) and provides a simple interface to the operating system, the batch system and the file system of the target resource.

It is the only UNICORE component that runs as root. For each request, the TSI will switch to the requested (non-root!) userid/groupid to perform the work.

Daemons running

The main perl process (called TSI shepherd) forks child processes upon request from the XNJS (which is a part of the UNICORE/X server), which then perform the work.

Init scripts and options (start|stop|restart|...)

The service is started and stopped using shell scripts in the bin/ folder of the installation.

Configuration files location with example or template

Configuration files are located in the installation directory

- conf/tsi.properties
- perl/SharedConfiguration.pm (NEW since 6.3.2 rc1)

Logfile locations (and management) and other useful audit information

Logfiles are by default placed in the logs/ directory in the installation. Usually not much is logged there, but some debug information is returned to the XNJS and can be logged to the XNJS logfile.

Open ports

- the TSI shepherd listener port, configured in the tsi.properties file

Possible unit test of the service

n/a

Where is service state held (and can it be rebuilt)

The TSI is a stateless service.

Cron jobs

n/a

Security information

Access control Mechanism description (authentication & authorization)

By default the TSI listens on a plain TCP socket. For an incoming connection, it is checked that the connection is from one of the hosts that are explicitly configured in the tsi.properties file. Then, the TSI connects to the configured XNJS ports (i.e. performs a callback).

Optionally, the XNJS/TSI connection can be configured to use SSL.

How to block/ban a user

On the TSI itself it is not possible, it should be done on a higher level: either by revoking the certificate, or by removing a user's attributes from the configured attribute sources (e.g. XUADB)

Network Usage

The TSI will receive incoming connections from the XNJS. It will call back the XNJS, i.e. the TSI will open connections to the XNJS machine.

Firewall configuration

n/a

Security recommendations

The TSI runs as root. Thus, the TSI files should be protected by the usual UNIX means.

Security incompatibilities

None known.

List of externals (packages not from the OS)

n/a

Other security relevant comments

n/a

Utility scripts

n/a

-- BerndSchuller - 19-Oct-2010

This topic: EMI > UNICORETsiSRC

Topic revision: r2 - 2011-04-27 - BerndThomasSchullerExCern



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

UNICORETsiSRC < EMI < TWiki

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback