
CERN

EOS Third-Party-Copy support for delegated X509 and tokens (Macaroons/SciTokens)

Elvin Sindrilaru
on behalf of the **EOS team**

Outline

- Why all the excitement around tokens and HTTP?
- TPC transfers using X509 delegated credentials
- Support for Macaroons and SciTokens
- HTTP TPC support

A bit of context …

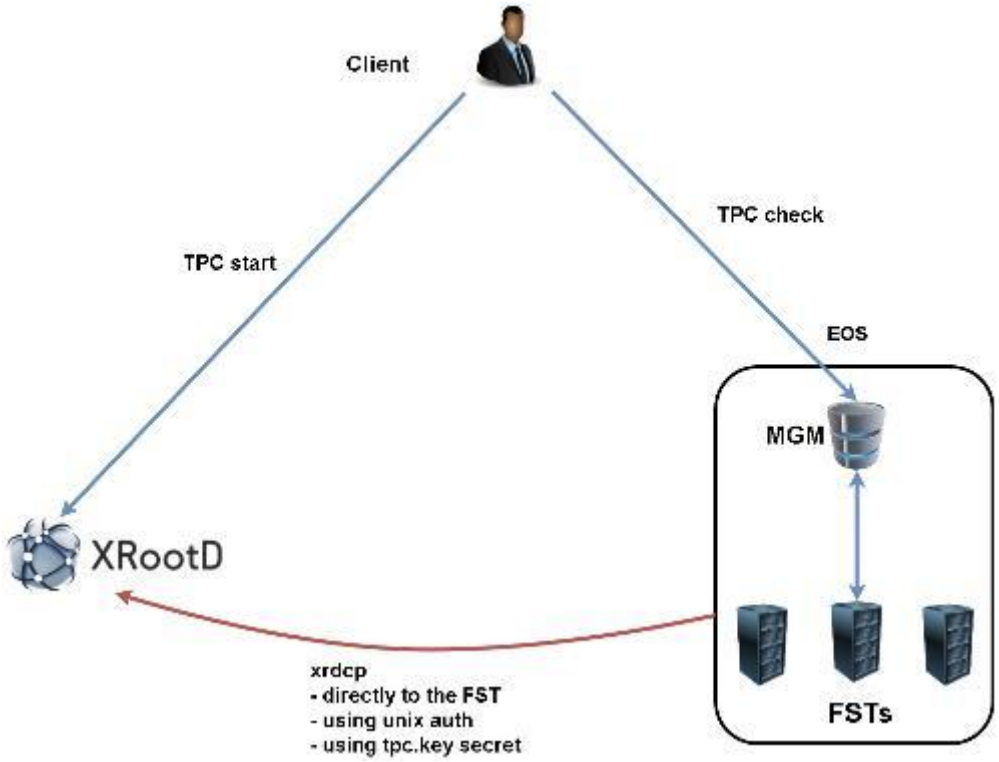
- **Globus Toolkit** open source support discontinued – need alternative for GridFTP

- **WLCG DOMA** (Data Organization Management Access) Group looking into alternatives
  - **XRootD TPC** with delegation
  - **HTTP(S) TPC** with macaroon support

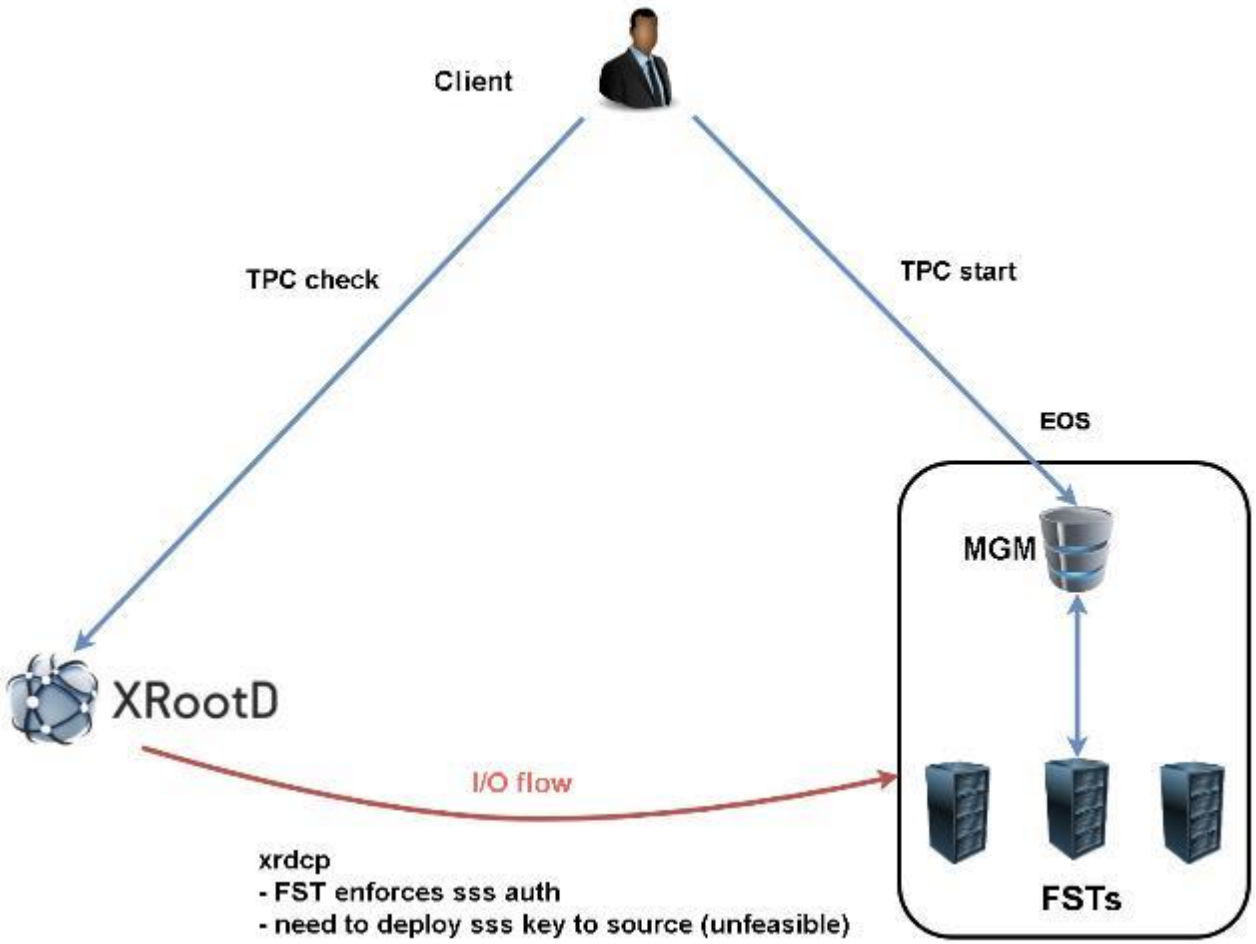
- EOS enforces authentication only at the MGM
  - FSTs rely on the (sss) encrypted opaque info
  - FSTs **enforce SSS authentication** for outgoing connections

TPC transfer with EOS as source

- XRootD TPC is a **pull** based model i.e. destination copies from source

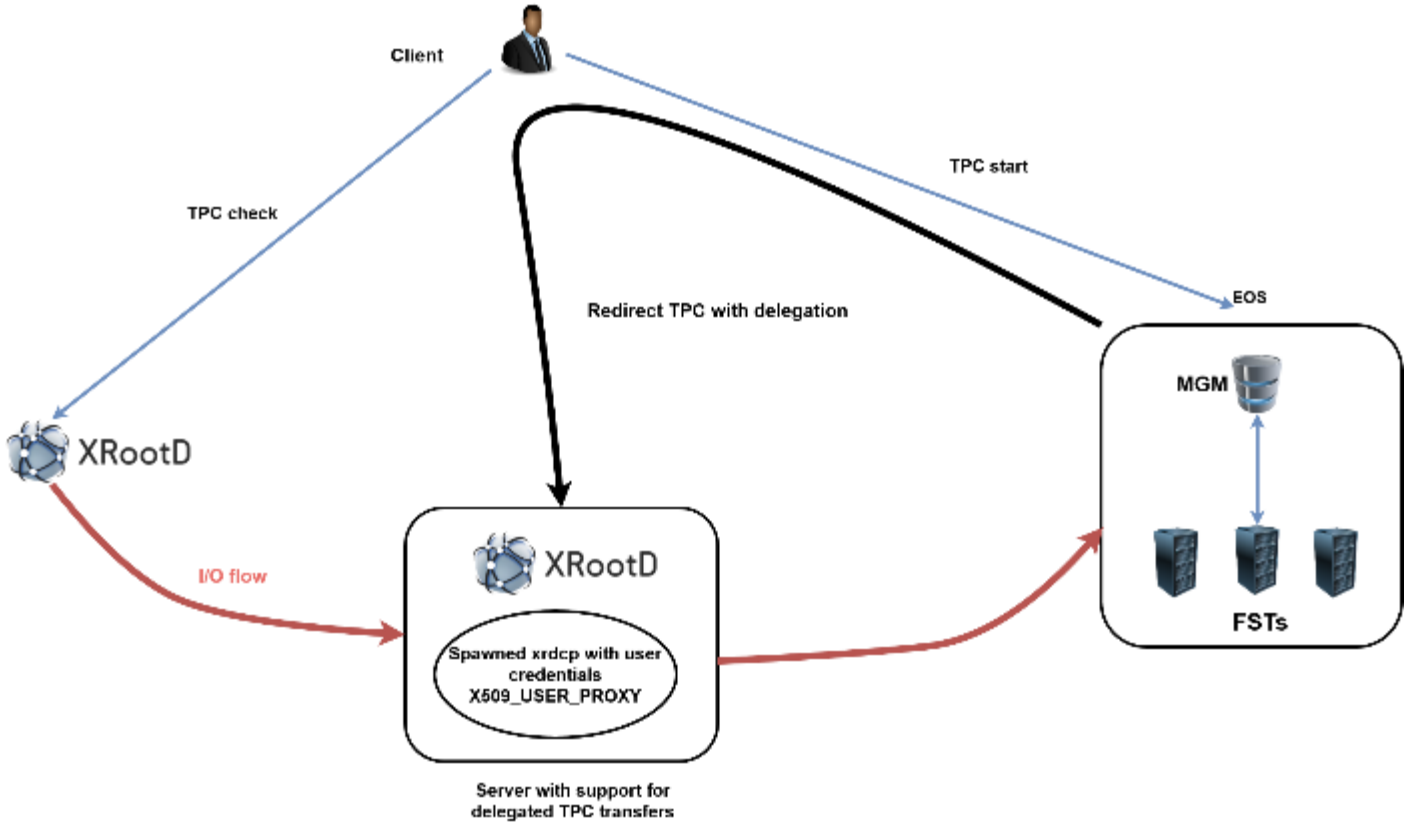


TPC transfer with EOS as destination



XRootD TPC with delegation

- Deploy an **XRootD vanilla gateway server** capable of doing delegated TPC transfers
  - Only **GSI delegation** is supported



XRootD TPC with delegation - configuration

- MGM side one line configuration addition

```
// Add to /etc/xrd.cf.mgm file
ofs.tpc redirect delegated eosgateway.cern.ch:1094
```

- XRootD vanilla PSS server playing the gateway role

```
// XRootD PPS config /etc/xrootd/xrootd-tpc.cfg
sec.protocol  gsi -dlgpxy:1 -exppxy:=creds -crl:1 -moninfo:1 -cert:/etc/grid-security/daemon/gridftp-cert.pem
-key:/etc/grid-security/daemon/gridftp-key.pem -gridmap:/etc/grid-security/grid-mapfile -d:1 -gmapopt:2
sec.protbind  * gsi
ofs.tpc  autorm fcreds gsi =X509_USER_PROXY ttl 60 60 xfr 9 pgm /usr/local/bin/xrootd-third-party-copy.sh
```

EOS support for token authorization

- Based on **XrdHttp** and work done in the **DOMA TPC** working group
  - Using custom http external handler **libEosMgmHttp.so**
- **Macaroon** tokens supported by **libXrdMacaroons.so** that comes by default with XRootD
- **SciTokens** support, install extra packages:
  - **eos-scitokens**, **eos-scitokens-debuinfo** from eos-depend repository
- Other useful packages:
  - **xrdhttpvoms** – support client proxy certificates and VOMS
  - **x509-scitokens-issuer**, **x509-scitokens-issuer-client** include macaroon-init tool for obtaining macaroons using X509
  - **python2-macaroons** – for inspecting the contents of macaroons

EOS MGM config for token support

```
# Load and enable HTTP(S) access on port 9000 on the current instance
xrd.protocol XrdHttp:9000 /usr/lib64/libXrdHttp.so
# Directory containing CA certificates to be used by the server
http.cadir /etc/grid-security/certificates/
# File containing the x509 server certificate
http.cert /etc/grid-security/daemon//hostcert.pem
# File containing the x509 server private key
http.key /etc/grid-security/daemon/hostkey.pem
# Path to the "grid map file" to be used for mapping users to specific identities
http.gridmap /etc/grid-security/grid-mapfile
# Load the XrdHttpVOMS security extractor plugin that is able to deal with
# proxy certificats and VOMS credentials
http.secxtractor libXrdHttpVOMS.so
# Load the XrdTpc external handler which deals only with COPY and OPTIONS http
# verbs and provides the default HTTP TPC functionality
http.exthandler xrdtpc /usr/lib64/libXrdHttpTPC.so
# Load the EOS specific HTTP external handler libEosMgmHttp.so and also specify
# the option is HTTP traffic is to be redirected to HTTP(S)
http.exthandler EosMgmHttp /usr/lib64/libEosMgmHttp.so eos::mgm::http::redirect-to-https=0
# The following two external library plugins are used to provide support for
# token based authentication with Macaroons and SciTokens.
mgmofs.macaroonslib /usr/lib64/libXrdMacaroons.so /opt/eos/lib64/libXrdAccSciTokens.so
# Base64-encoded secret key used for generating macroon
macaroons.secretkey /etc/eos.macaroon.secret
# Mandatory sitename configuration for the XrdMacaroons library which is also
# embedded in the macaroons attributes
all.sitename eosdev
```

EOS FST config for (TPC) token support

- **No changes** per se for token support
- But there are some changes needed for **HTTP TPC** support

```
# Enable the XrdHttp plugin and listen on port 9001 for connections
xrd.protocol XrdHttp:9001 /usr/lib64/libXrdHttp.so
# Load the libEosFstHttp external handler
http.exthandler EosFstHttp /usr/lib64/libEosFstHttp.so none
# Load the XrdTpc external handler which deals with COPY and OPTIONS http
# verbs and provides the default HTTP TPC functionality
http.exthandler xrdtpc /usr/lib64/libXrdHttpTPC.so
```

Practical examples(1) – X509

- XRootD TPC with delegated credentials

```
# Set the path for X509 user "foo"
export X509_USER_CERT=/home/foo/.globus/usercert.pem
export X509_USER_KEY=/home/foo/.globus/userkey.pem
XrdSecPROTOCOL=gsi,unix xrdcp --tpc delegate only root://eos1.cern.ch//src root://other.world.com//dst
```

- CURL (direct) transfer using X509 credentials

```
curl -L -v --capath /etc/grid-security/certificates --cert ~/.globus/usercert.pem --cacert ~/.globus/usercert.pem
--key ~/.globus/userkey.pem https://e0.cern.ch:9000//eos/dev/replica/file1.dat --upload-file /etc/passwd
```

Practical examples(2) – Macaroons TX

- CURL (direct) transfer using macaroons

```
# Make sure the following environment variables point to the client
# certificate and private key
X509_USER_CERT=/home/esindril/.globus/usercert.pem
X509_USER_KEY=/home/esindril/.globus/userkey.pem
# Use the macaroon-init tool to request a macaroon
macaroon-init https://esdss000.cern.ch:9000//eos/ 60 DOWNLOAD,UPLOAD
MDAxNGxvY2F0aW9uIGVvc2RldgowMDM0aWRlbnRpZmllciBiYzhiZWRmZC0wNzJjLTRmZWEtYjNiYy0wNDJjZjczZDhiYjMKMDAxNmNpZCBuYW1lOm
VzaW5kcmlsCjAwMWZjaWQgYWN0aXZpdHk6UkVBRF9NRVRBREFUQQowMDI4Y2lkIGFjdGl2aXR5OkRPV05MT0FELFVQTE9BRCxNQU5BR0UKMDAxM2Np
ZCBwYXRoOi9lb3MvCjAwMjRjaWQgYmVmb3JlOjIwMjAtMDEtMjlUMTU6MTM6MzVaCjAwMmZzaWduYXR1cmUgNm15NCbrb62KCIvxxDlSgrwgMZKjG
PrO7NwxFQwIycK
# Export the token as an environment variable for easier use later on
export
MACAROON=MDAxNGxvY2F0aW9uIGVvc2RldgowMDM0aWRlbnRpZmllciBiYzhiZWRmZC0wNzJjLTRmZWEtYjNiYy0wNDJjZjczZDhiYjMKMDAxNmNpZ
CBuYW1lOmVzaW5kcmlsCjAwMWZjaWQgYWN0aXZpdHk6UkVBRF9NRVRBREFUQQowMDI4Y2lkIGFjdGl2aXR5OkRPV05MT0FELFVQTE9BRCxNQU5BR0U
KMDAxM2NpZCBwYXRoOi9lb3MvCjAwMjRjaWQgYmVmb3JlOjIwMjAtMDEtMjlUMTU6MTM6MzVaCjAwMmZzaWduYXR1cmUgNm15NCbrb62KCIvxxDlS
grwgMZKjGPrO7NwxFQwIycK
# Use the curl command to trigger the transfer (download) and properly
# populate the header information with the authentication information
curl -v -L -H "Authorization: Bearer $MACAROON" https://esdss000.cern.ch:9000/eos/dev/replica/file1.dat
```

What is inside my macaroon?

Practical examples(3)

- Inspect the contents of a macaroon

```
>>> import macaroons
>>> secret = open("/etc/eos.macaroon.secret", 'r').read()
>>> mtoken =
"MDAxNGxvY2F0aW9uIGVvc2RldgowMDM0aWRlbnRpZmllciBiYzhiZWRmZC0wNzJjLTRmZWEtYjNiYy0wNDJjZjczZDhiYjMKMDAxNmNpZCBuYWllO
mVzaW5kcmlsCjAwMWZjaWQgYWN0aXZpdHk6UkVBRF9NRVRBREFUQQowMDI4Y2lkIGFjdGl2aXR5OkRPV05MT0FELFVQTE9BRCxNQU5BR0UKMDAxM2N
pZCBwYXRoOi9lb3MvCjAwMjRjaWQgYmVmb3JlOjIwMjAtMDEtMjlUMTU6MTM6MzVaCjAwMmZzaWduYXR1cmUgNm15NCbrb62KCIvxxDlSgrwgMZKj
GPrO7NwxFQwIycK"
>>> M = macaroons.deserialize(mtoken)
>>> print M.inspect()
location eosdev
identifier bc8bedfd-072c-4fea-b3bc-042cf73d8bb3
cid name:esindril
cid activity:READ_METADATA
cid activity:DOWNLOAD,UPLOAD,MANAGE
cid path:/eos/
cid before:2020-01-29T15:13:35Z
signature b8d9b5e4d09badbeb628222fc710e54a0af080c64a8c63eb3bb370c454302327
```

Practical examples(4) – SciTokens TX

- Requires an **IAM(Identity and Access Management)** provider and a client (**oidc-agent**)

```
# Start the oidc-agent in the background
eval $(oidc-agent)
oidc-gen WLCG-<your_username> -w decive
# Put as issuer https://wlcg.cloud.cnaf.infn.it/ and configure the set of
# scopes as "max". Then connect the agent to the IAM provide which will
# prompt you for the password you set up earlier.
oidc-add WLCG_<your_username>
# Request a token from the IAM and save it as an environment variable for
# later use
export SCI_TOKEN=`oidc-token WLCG_<your_username>`
# Trigger a HTTP download using the SciToken information
curl -v -L -H "Authorization: Bearer $SCI_TOKEN" https://esdss000.cern.ch:9000/eos/dev/replica/file1.dat
```

Minimum version requirements

- Support for everything presented so far requires:
  - **XRootD >= 4.11.1**
  - **EOS >= 4.6.8**
  - **XRootD client >= 4.11.1**

- **HTTP TPC** support requires XRootD 4.11.2 and a new EOS release

Reference setup and configuration

- EOS setup and example commands
  - http://eos-docs.web.cern.ch/eos-docs/configuration/http_tpc.html
- Macaroons description
  - https://github.com/rescrv/libmacaroons


CERN