



XRootD/SciToken-Based Access to Lustre Storage at GSI





A First Step Toward Data Federation for FAIR

Mohammad Al-Turany, Sören Fleischer, Thorsten Kollegger, Anar Manafov, Rouven Spreckels

- **Goal** Make reading+writing files available via both CLI and web browser.
 - Implement authentication using GSI Weblogin and SciTokens. Weblogin is an account at GSI intended for employees, apprentices, external companies, guests, guest scientists.
 - Utilize POSIX ownership and permissions to manage access rights.

Block Diagram



Service Configuration

📃 🖈 🛛 xrootd-http.cfg (~/daten/GSI/git.gsi.de/dc/common/grid-ansible/playbooks/punch2/files/xrootd) - Pluma 🗸 🔿 🗙	📃 🖈 💿 scitokens.cfg (~/daten/GSI/git.gsi.de/dc/common/grid-ansible/playbooks/punch2/files/xrootd) - Pluma 🗸 🔨 🗙	📄 🖈 xrootd-privileged@.service (~/daten/GSI/git.gsi.de//griansible/playbooks/punch2/files/xrootd) - Pluma 🗸 🔨 🗙
File Edit View Search Tools Documents Help	File Edit View Search Tools Documents Help	File Edit View Search Tools Documents Help
D De Open → 🖹 Save 🛱 5 Undo C 🕺 D 🖻 🔍 O.	📭 🖪 Open 🔻 🖺 Save 🛱 5 Undo C 🐰 🗈 🖻 🔍 🔍	다. 🔓 Open 🔻 🖹 Save 🛱 5 Undo C 🐰 🗈 🖹 🔍 🔍
🖹 scitokens.cfg 🖹 xrootd-http.cfg 🖹 xrootd-privileged@.service	🖹 scitokens.cfg 🔕 🖹 xrootd-http.cfg 🔕 🖹 xrootd-privileged@.service 🔕	🖹 scitokens.cfg 🔕 🖹 xrootd-http.cfg 🔕 🖹 xrootd-privileged@.service 🔇
<pre>1 all.export / r/w 2 3 all.adminpath /var/spool/xrootd 4 all.pidpath /run/xrootd 5 6 oss.localroot /data/xrootd 7 8 if exec xrootd 9 xrd.protocol XrdHttp:80 libXrdHttp.so 10 fi 11 12 ofs.osslib ++ libXrdMultiuser.so 13 14 ofs.authorize 15 http.header2cgi Authorization authz 16 ofs.authlib libXrdAccSciTokens.so config=/etc/xrootd/scitokens.cfg 17 scitokens.trace all 18 19 continue /etc/xrootd/config.d/</pre>	<pre>1 Global 2 audience = https://id.gsi.de/realms/wl 3 4 [Issuer https://id.gsi.de/realms/wl] 5 issuer = https://id.gsi.de/realms/wl 6 base_path = / 7 map_subject = True 8 default_user = xrootd 9 name_mapfile = /etc/xrootd/mapfile</pre>	<pre>1 [Unit] 2 Description=XRootD xrootd daemon instance %I 3 Documentation=man:xrootd(8) 4 Documentation=https://xrootd.org/docs.html 5 Requires=network-online.target 6 After=network-online.target 7 8 [Service] 9 # Note "-R xrootd" here instructs xrootd to drop privileges to the xrootd Unix user. 10 ExecStart=/usr/bin/xrootd -1 /var/log/xrootd/xrootd.log -c /etc/xrootd/xrootd-%i.cfg -k fifo -s /var/run/xrootd/xrootd-%i.pid -n %i -R xrootd 11 Environment=XC_ENABLE_MULTIUSER=1 12 Type=simple 13 Restart=on-abort 14 RestartSec=0 15 KillMode=control-group 16 LimitNOFILE=65536 17 WorkingDirectory=/var/spool/xrootd 18 19 # These provide xrootd with the ability to switch UIDs/GIDs for reading/writing files. 2 Capabilities=CAP_SETGID+p CAP_SETGID CAP_DAC_OVERRIDE 2 [Install] 24 RequiredBy=multi-user.target </pre>
.ini 🔻 Tab Width: 4 💌 Ln 1, Col 1 INS	.ini ▼ Tab Width: 4 ▼ Ln 1, Col 1 INS	.ini ▼ Tab Width: 4 ▼ Ln 6, Col 28 INS

Web Browser

The user initially does not have a token in their browser.

- (a) When the user initially browses to the web frontend, the
- JavaScript code is downloaded to the user's browser.
- b The code, which is now running in the user's browser communicates with the web backend. Since the user's browser does not yet have a token, they do not see any data, but they have the option to log in.
- © When the user clicks log in, they browse to id.gsi.de, where they can supply their credentials. If the authentication succeeds, id.gsi.de answers to the user's browser with an HTTP redirect response to the web backend with an ephemeral one-time code in the GET parameters.



Keycloak is configured to require 2FA.

User Workflows



(C)

The user initially does not have a token.

The user sends an HTTPS request to id.gsi.de, specifying their GSI Weblogin username and password. If everything is correct, the server responds with a token.



D *	tokio : bash — Konsole	~ ^ X
📮 New Tab 🚬 🔲 Split View 🗸		🗅 Copy 🗐 Paste 🔍 Find 🚍
[user1@f13 tokio]\$ curl -H "\$TOK fsjdklfjsdlf 12345 [user1@f13 tokio]\$	<pre>KEN" http://punch2.gsi.de/sfleisch/blabla.txt</pre>	0

f XRootD downloads the public key of id.gsi.de and verifies that the token is signed by id.gsi.de.

(b) The code in the user's browser makes the aforementioned

HTTP request with the one-time code to the web backend.

(d) The web backend sends the one-time code to keycloak, which, given correct inputs, returns the tokens. These are returned in an HTTP response to the user's browser.

(b) Subsequents requests from the user's browser to the REST API contain the token.

(e) The web backend requests the list of files in the specified directory from the XRootD server.

🔵 🔵 🖤 Data Lake UI		× +				~		
\leftrightarrow \rightarrow C \bigcirc punch2.g	gsi.de:517	73/browse/%2FUsers%2Fanar%2FBIG_DATA			Rela	aunch to update :		
DATA LAKE UI	Browse	e files and folders Documentation	More 🥆			Log In		
 Navigator One Settings 	~	Initial Directory / BIG_DATA Current page: 500 (0 folders, 500 files), cumulative file size: 11.50 kB Total: 400000 (0 folders, 400000 files), cumulative file size: 9.20 MB						
		< 1 2	3 4 5	6 ••• 800 > G	Bo to 1			
		Name ≑		Size 🌲	Date 🌲	Туре 🌲		
		aaaaokvn.txt		23.00 B	2024-09-30 09:40:01	file		
		aaadhncb.txt		23.00 B	2024-09-30 09:39:36	file		
		aaadpmzx.txt		23.00 B	2024-09-30 09:39:21	file		
		aaadzbmz.txt		23.00 B	2024-09-30 09:39:52	file		
		aaafdntl.txt		23.00 B	2024-09-30 09:40:36	file		
		aaageugo.txt		23.00 B	2024-09-30 09:40:32	file		
		aaaglurq.txt		23.00 B	2024-09-30 09:40:26	file		
		aaahruzd.txt		23.00 B	2024-09-30 09:40:10	file		
0		aaaiunfb.txt		23.00 B	2024-09-30 09:39:26	file		

- 9 XRootD reads the mapfile to determine which UNIX username belongs to the Weblogin username specified in the token.
- (h) XRootD performs the actual file I/O on the user data as the UNIX username found in the mapfile. The permission check is then performed by the Linux kernel according to POSIX rules.

(root) punch2 — Konsole 📑 New Tab 🚺 Split View Copy 📄 Paste 🔍 Find... 🗮 tokio : bash × (root) punch2 241015 14:18:08 150418 XrootdBridge: unknown.31:28@[::ffff:140.181.13.191] login as nobody 241015 14:18:08 150418 multiuser_UserSentry: Anonymous client; no user set, cannot change FS UIDs 241015 14:18:08 150418 multiuser_UserSentry: Anonymous client; no user set, cannot change FS UIDs 241015 14:18:08 150418 scitokens_Access: Trying token-based access control 241015 14:18:08 150418 scitokens_Access: Cached token mapped_username=sfleisch, subject=sfleisch, issuer=https://id.gs i.de/realms/wl, authorizations=/:read,dir,stat 241015 14:18:08 150418 scitokens_Access: Grant authorization based on scopes for operation=read, path=/sfleisch/blabla 241015 14:18:08 150418 scitokens_Access: Request username sfleisch 241015 14:18:08 150418 multiuser_UserSentry: Switching FS uid for user sfleisch 241015 14:18:08 150418 multiuser_UserSentry: Switching FS uid for user sfleisch 241015 14:18:08 150418 multiuser_Open: Will not create checksum 241015 14:18:08 150418 multiuser_Open: Will not create checksum 241015 14:18:08 150364 XrootdXeq: unknown.31:28@[::ffff:140.181.13.191] disc 0:00:00 241015 14:18:08 150364 multiuser_UserSentry: Anonymous client; no user set, cannot change FS UIDs 241015 14:18:08 150364 multiuser_UserSentry: Anonymous client; no user set, cannot change FS UIDs

work in progress	aaajbwte.txt	23.00 B	2024-09-30 09:40:17	file
p1 091 000	🖹 aaakgonm.txt	23.00 B	2024-09-30 09:39:43	file

f XRootD downloads the public key of id.gsi.de and verifies that the user-supplied token is signed by id.gsi.de
9 XRootD reads the mapfile to determine which UNIX username belongs to the Weblogin username specified in the token.
i The links to the actual files are pointing directly to the XRootD server, which is contacted directly by the user's browser when they click on a file.
h The XRootD executable performs the actual file I/O on the user data as the UNIX username found in the mapfile. The permission check is then performed by the Linux kernel according to POSIX rules.

