



XRootD

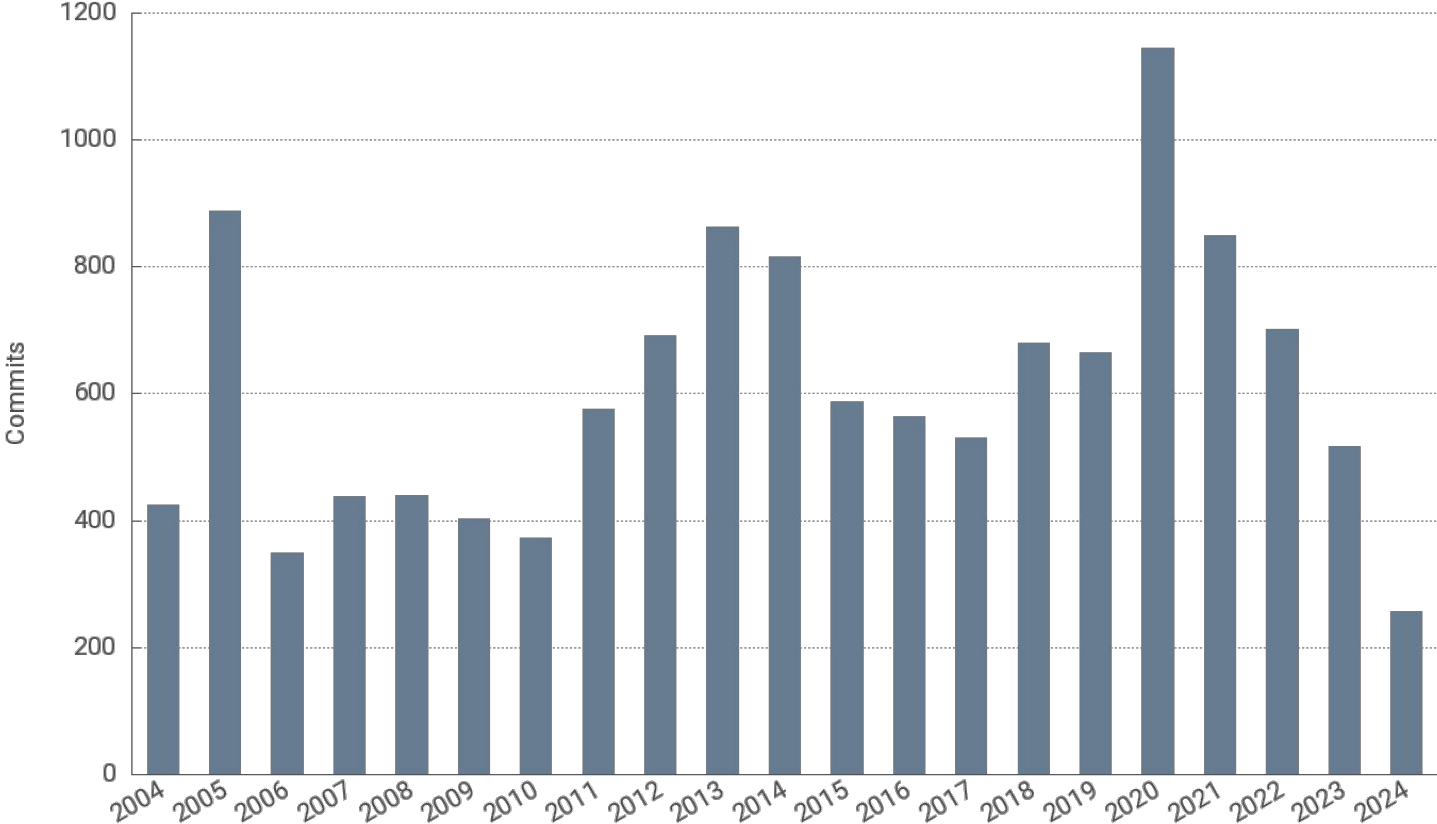
Catching Up with XRootD (Part 2)

XRootD GitHub Statistics

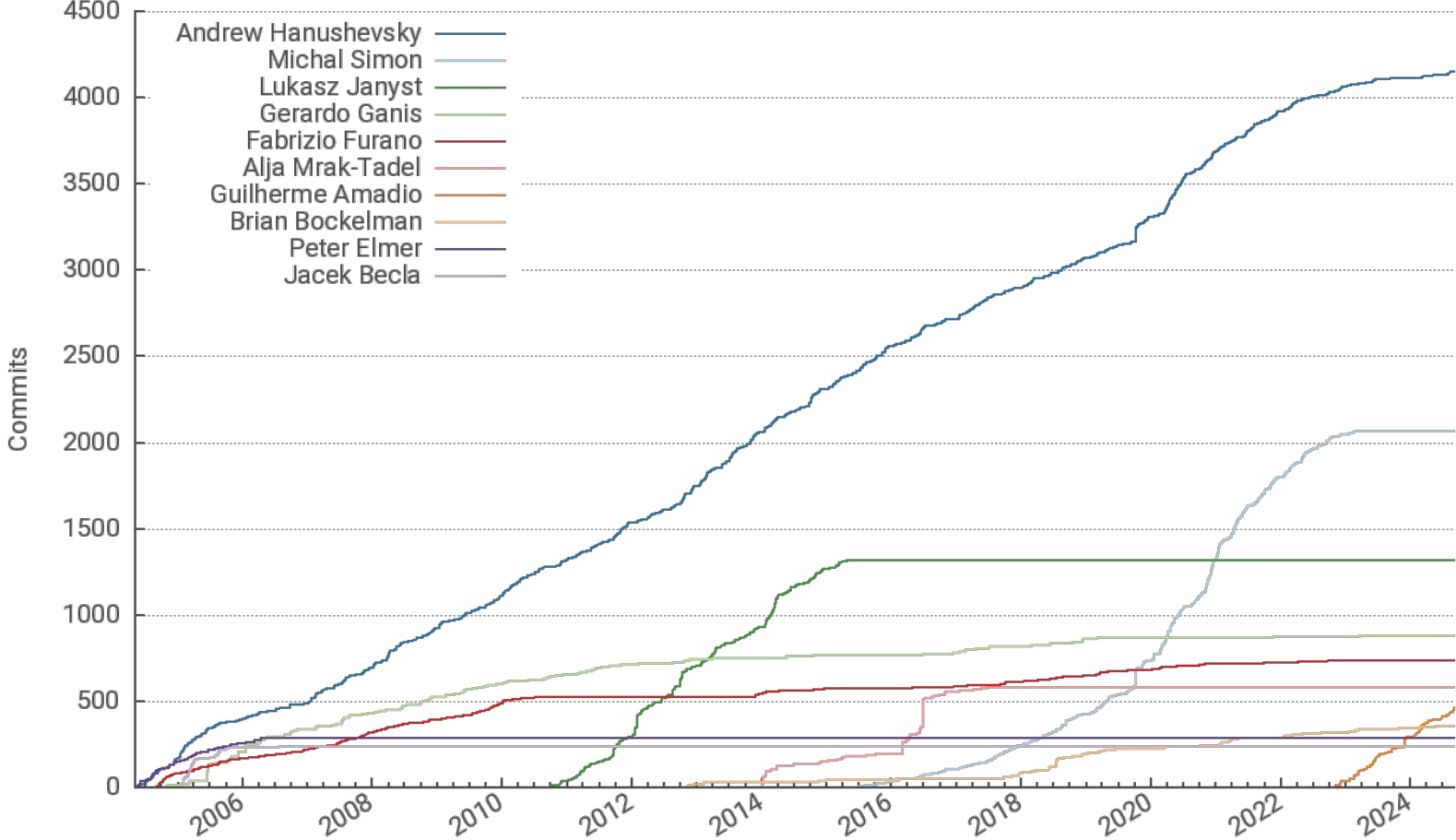
- ▶ **Releases:**
 - From 5.5.4 to 5.7.1 – 14 releases in total
- ▶ **Pull Requests:**
 - 204 pull requests since Jan 2023, 4 currently open
- ▶ **Issues:**
 - Total: 89 open, 1239 closed issues
 - Stats for 2023/2024: 254 issues opened, 194 closed (~78%)
 - Many issues are really just questions
 - Average of ~2 weeks to close
- ▶ **Contributors:**
 - From 2 to 12 contributors each month (2023 – 2024)
 - Average around 5~10 contributors per release
 - Total contributors: ~100
- ▶ **Most contributions by CERN, SLAC, UCSD**

Domains	Total (%)
cern.ch	6578 (51.58%)
slac.stanford.edu	4643 (36.40%)
ucsd.edu	740 (5.80%)
cse.unl.edu	285 (2.23%)
physics.uu.se	123 (0.96%)

20 Years of XRootD Commit Activity



20 Years of XRootD Commit Activity



XRootD in HEP Community

- ▶ **Core component of HEP software ecosystem**
 - Depended on by CTA, FTS, EOS, ROOT, Rucio, experiment frameworks, etc
- ▶ **Exabytes of data processed each year**
- ▶ **Needs security, stability, scalability, sustainability**
 - Code scanning (CodeQL), security policy setup on GitHub (allows private bug reporting)
 - Continuous effort to improve testing infrastructure
 - Measure and expand test coverage, use static analysis tools, ABI tracker, automatic testing in CI
 - Performance analysis of production workloads to guide performance optimizations
 - Lower barrier for contributors and users as much as possible
 - Make it as easy as possible to configure, build, run tests, and create packages

XRootD 5.7.x Highlights

▶ XRootD 5.7.0

- Moved baseline C++ standard to C++17
- Updated min/default RSA bits to 2048
- Restrict renegotiation for TLSv1.2 and earlier
- Redact authz tokens from server logs
- HTTP header parsing now case insensitive for better compatibility
- Make more client errors recoverable to allow retry in more situations
- Several improvements in XrdScitokens plugin (new auth strategies)
- Reintegrated XrdCeph into main repository, now uses `-DENABLE_CEPH=ON` option
- Better performance for GSI authentication by avoiding duplicate work (needed on Alma 9)
- Enable `SSL_OP_IGNORE_UNEXPECTED_EOF` option if available, increase default timeouts
- XRootD now includes a sample shell script for third-party copy (TPC) configuration
- Erasure coding plugin now enabled by default and using OS-provided isa-l library
- Completed migration of test suite to GoogleTest
- Updated Doxygen documentation

XRootD Sample Shell Script for Third Party Copy

```
#!/bin/bash

OPTS=(" ${@:1:$#-2} ")
shift $(( $# - 2 ))

SRC=$1
DST=$2

if [[ -n "${XRD_XROOTD_ORIGIN}" ]]; then
    DST="root://${XRD_XROOTD_ORIGIN}/${DST}"
fi

SAFE_SRC=${SRC/\? *}
SAFE_DST=${DST/\? *}











logger -p info -t xrdcp-tpc "start transfer: ${SAFE_SRC} =>
${SAFE_DST}"

xrdcp --server "${OPTS[@]}" "${SRC}" "${DST}"
STATUS=$?

if [[ ${STATUS} -eq 0 ]]; then
    logger -p info -t xrdcp-tpc \
        "transfer: ${SAFE_SRC} => ${SAFE_DST} completed successfully"
else
    logger -p err -t xrdcp-tpc \
        "transfer: ${SAFE_SRC} => ${SAFE_DST} FAILED [exit code: ${STATUS}]"
fi

exit ${STATUS}
```

Popular content

Content	Views	Unique visitors
 xrootd/xrootd: The XRootD centr...	179	76
 Issues	127	24
 Pull requests	70	18
 Server responded with an error: [...	61	22
 xrootd/utis/xrdcp-tpc.sh at master	57	13
 xrootd/src at master	47	12
 xrootd/src/XrdHttp/XrdHttpProtoc...	34	1
 Tags	29	4
 xrootd/docs/INSTALL.md at master	24	4
 xrootd/src/XrdHttp/XrdHttpProtoc...	23	1

XRootD Doxygen Documentation

Before

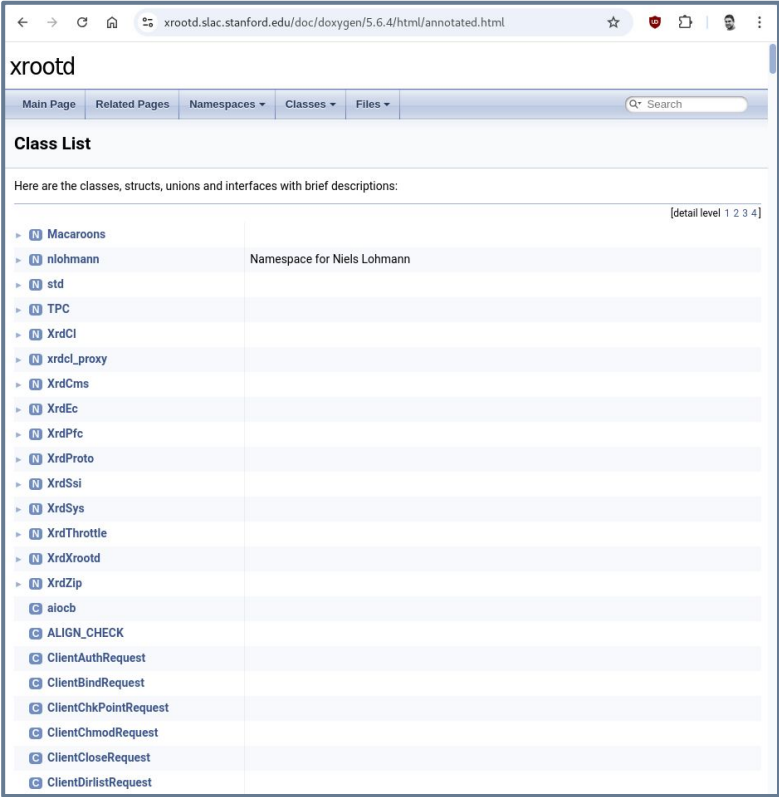
The screenshot shows a browser window with the URL `xrootd.slac.stanford.edu/doc/doxygen/5.6.4/html/annotated.html`. The page title is "xrootd". Below the navigation bar, there is a "Class List" section. A heading reads "Here are the classes, structs, unions and interfaces with brief descriptions:". Below this, there is a list of classes and namespaces, including "Macaroons", "niohmann", "std", "TPC", "XrdCl", "xrdecl_proxy", "XrdCms", "XrdEc", "XrdPfc", "XrdProto", "XrdSsi", "XrdSys", "XrdThrottle", "XrdXrootd", "XrdZip", "aiocb", "ALIGN_CHECK", "ClientAuthRequest", "ClientBindRequest", "ClientChkPointRequest", "ClientChmodRequest", "ClientCloseRequest", and "ClientDirlistRequest".

After

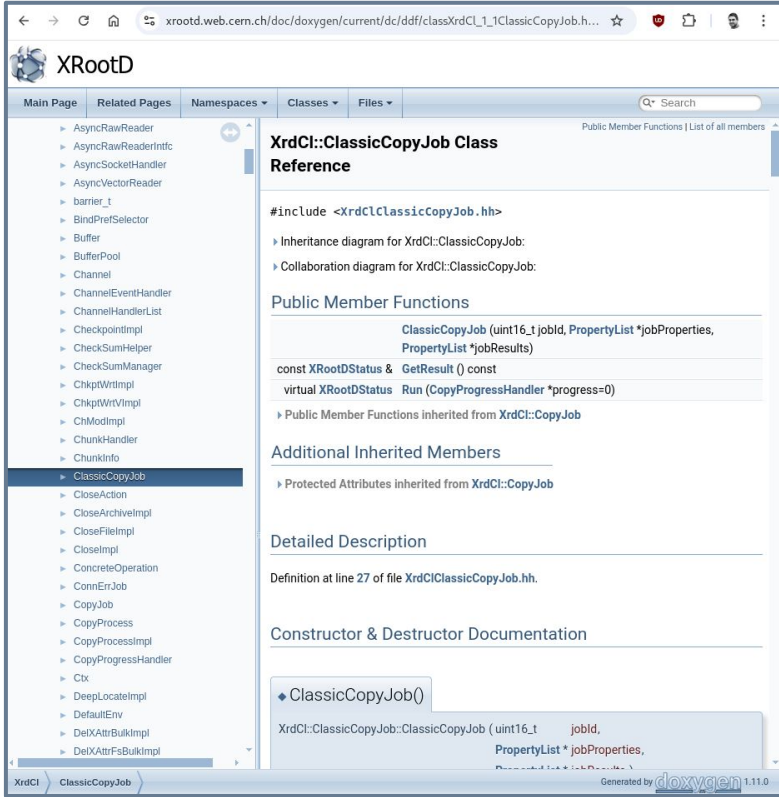
The screenshot shows a browser window with the URL `xrootd.web.cern.ch/doc/doxygen/current/`. The page title is "XRootD". The navigation bar includes "Main Page", "Related Pages", "Namespaces", "Classes", and "Files". The main content area features the XRootD logo and the heading "XRootD: eXtended ROOT Daemon". Below this, there is a paragraph describing the project: "The XRootD project provides a high-performance, fault-tolerant, and secure solution for handling massive amounts of data distributed across multiple storage resources, such as disk servers, tape libraries, and remote sites. It enables efficient data access and movement in a transparent and uniform manner, regardless of the underlying storage technology or location. It was initially developed by the High Energy Physics (HEP) community to meet the data storage and access requirements of the BaBar experiment at SLAC and later extended to meet the needs of experiments at the Large Hadron Collider (LHC) at CERN. XRootD is the core technology powering the EOS distributed filesystem, which is the storage solution used by LHC experiments and the storage backend for CERNBox. XRootD is also used as the core technology for global CDN deployments across multiple science domains." Below this, there is a section titled "Documentation" with the text: "General documentation such as configuration reference guides, and user manuals can be found on the XRootD website at <http://xrootd.org/docs.html>." There is also a section titled "Supported Operating Systems" with the text: "XRootD is officially supported on the following platforms:" followed by a list item: "RedHat Enterprise Linux 7 or later and their derivatives". The footer of the page indicates it was "Generated by doxygen 1.11.0".

XRootD Doxygen Documentation

Before



After



XRootD Doxygen Documentation

The screenshot shows the XRootD Doxygen documentation interface. On the left is a navigation pane with a tree view of classes and files. The main content area displays the signature and details for the `ChMod()` method. The signature is: `XRootDStatus XrdCl::FileSystem::ChMod (const std::string & path, Access::Mode mode, ResponseHandler * handler, uint16_t timeout = 0)`. Below the signature, it states: "Change access mode on a directory or a file - async". The "Parameters" section lists: `path` (file/directory path), `mode` (access mode, or'd `Access::Mode`), `handler` (handler to be notified when the response arrives), and `timeout` (timeout value, if 0 the environment default will be used). The "Returns" section states: "status of the operation". Below this, it says "Definition at line 1462 of file XrdClFileSystem.cc." and shows the corresponding C++ code snippet from lines 1466 to 1485.

XRootD

Main Page | Related Pages | Namespaces | Classes | Files

- ▶ DelXAttrFsBulkImpl
- ▶ DelXAttrFsImpl
- ▶ DelXAttrImpl
- ▶ DirectoryList
- ▶ DirListFlags
- ▶ DirListImpl
- ▶ DlgEnv
- ▶ EcHandler
- ▶ EcPgReadResponseHandler
- ▶ EcPluginFactory
- ▶ Env
- ▶ ExOpenFuncWrapper
- ▶ FcntlAction
- ▶ FcntlImpl
- ▶ File
- ▶ FileImpl
- ▶ FileOperation
- ▶ FilePlugin
- ▶ FileStateHandler
- ▶ **FileSystem**
- ▶ FileSystemData
- ▶ FileSystemImpl
- ▶ FileSystemOperation
- ▶ FileSystemPlugin
- ▶ FileSystemUtils
- ▶ FileTimer
- ▶ FinalOperation
- ▶ ForkHandler
- ▶ FreeSpace
- ▶ FSExecutor
- ▶ FunctionWrapper
- ▶ FunctionWrapper< void >
- ▶ FutureWrapper
- ▶ FutureWrapper< void >
- ▶ FutureWrapperBase
- ▶ Fwd
- ▶ FwdStorage

◆ ChMod() [1/2]

XRootDStatus XrdCl::FileSystem::ChMod (const std::string & path,
Access::Mode mode,
ResponseHandler * handler,
uint16_t timeout = 0)

Change access mode on a directory or a file - async

Parameters

- path** file/directory path
- mode** access mode, or'd `Access::Mode`
- handler** handler to be notified when the response arrives
- timeout** timeout value, if 0 the environment default will be used

Returns

status of the operation

Definition at line 1462 of file XrdClFileSystem.cc.

```
1466     {  
1467         if( pPlugin )  
1468             return pPlugin->ChMod( path, mode, handler, timeout );  
1469  
1470         std::string fPath = FilterXrdClCgi( path );  
1471  
1472         Message *msg;  
1473         ClientChmodRequest *req;  
1474         MessageUtils::CreateRequest( msg, req, fPath.length() );  
1475  
1476         req->requestid = KXR_chmod;  
1477         req->mode = mode;  
1478         req->dlen = fPath.length();  
1479         msg->Append( fPath.c_str(), fPath.length(), 24 );  
1480         MessageSendParams params; params.timeout = timeout;  
1481         MessageUtils::ProcessSendParams( params );  
1482         XRootDTransport::SetDescription( msg );  
1483  
1484         return FileSystemData::Send( pImpl->fsdata, msg, handler, params );  
1485     }
```

XRootD 5.7.x Highlights

▶ XRootD 5.7.1

- Allow `cconfig` to write out combined config file
- Harden `systemd` service units for better security
- Fix memory leaks in Python bindings
- XRootD security policy described in [SECURITY.md](#)
- Ensure correct certificate is used when passed via `cgi` with `xrd.gsiusrproxy=...`
- Finished work on redacting token information from logs (client-side)
- Increased test coverage with more server/client setups
- Enabled CodeQL code scanning tool on GitHub
- Introduced abi-tracker reports to ensure full ABI backward compatibility

▶ XRootD on Alma 9 / RHEL 9

- Need to pay attention to `ulimit` settings to avoid running out of memory
- HTTP 1.1 vs HTTP 2.0 incompatibilities (case-insensitive headers from `curl`)
- Bad performance from OpenSSL 3.0 affects GSI + VOMS extraction configurations

XRootD Python Bindings Improvements

- ▶ Improved Python 3.x support
 - No more need to use raw strings, fixed file iteration by line, bug fixes
- ▶ Python 2.x support to be dropped along with CentOS 7 support
- ▶ Rewritten Python packaging
 - Integrated into CMake build
 - Compliant with PEP 517
 - Customizable builds (CXXFLAGS)
- ▶ Tests converted to Python 3.x
 - TODO: Integrate running Python tests with CTest (partially done)
- ▶ Machinery to build Python wheels is ready, need to find solution for linking OpenSSL

Code Scanning with CodeQL on GitHub

The screenshot shows the GitHub interface for the repository 'xrootd / xrootd'. The navigation bar at the top includes 'Code', 'Issues 89', 'Pull requests 4', 'Discussions', 'Actions', 'Projects', 'Security 194', 'Insights', and 'Settings'. The 'Security' tab is highlighted with a red box. On the left sidebar, 'Advisories' and 'Code scanning 194' are also highlighted with red boxes. The main content area is titled 'Code scanning' and shows a status message: 'All tools are working as expected'. Below this is a search bar with the query 'is:closed branch:master' and a button to 'Clear current search query, filters, and sorts'. A summary bar indicates '194 Open' and '83 Closed' issues. The list of issues includes:

- Potential use after free** Critical master
#4 closed as false positive last week • Detected by CodeQL in src/XrdSut/XrdSutPFCache.cc:272 3 weeks ago
- Likely overrunning write** Critical master
#183 closed as fixed last week • Detected by CodeQL in src/XrdHttp/XrdHttpSecurity.cc:210 3 weeks ago
- Potential use after free** Critical master
#3 closed as fixed last week • Detected by CodeQL in src/XrdSecztn/XrdSecProtocolztn.cc:670 3 weeks ago
- Potential double free** Critical master
#2 closed as fixed last week • Detected by CodeQL in src/XrdSecgsi/XrdSecProtocolgsi.cc:5835 3 weeks ago

Code Scanning with CodeQL on GitHub

The screenshot shows a GitHub interface for a repository named 'xrootd'. The navigation bar includes 'Code', 'Issues 89', 'Pull requests 4', 'Discussions', 'Actions', 'Projects', 'Security 194', 'Insights', and 'Settings'. The main content area displays a 'Code scanning alerts / #3' section with a title 'Potential use after free' and a 'Dismiss alert' button. A 'Fixed' badge is present, indicating the issue was resolved in the 'master' branch last week. The code snippet shows a C++ function in 'src/XrdSecZtn/XrdSecProtocolZtn.cc' at line 670. The code allocates memory for 'Entity.creds' and then attempts to access it after it has been freed. A message states: 'Memory may have been previously freed by call to free.' Below the code, there is a 'CodeQL' section with a 'Show paths' button. A table at the bottom provides details about the tool, rule ID, and query. On the right side, there are sections for 'Severity' (Critical), 'Affected branches' (master), 'Tags' (reliability, security), and 'Weaknesses' (CWE-416).

src/XrdSecZtn/XrdSecProtocolZtn.cc:670

```
667     Entity.credslen = strlen(tResp->tkn);
668     if (Entity.creds) free(Entity.creds);
669     Entity.creds = (char *)malloc(Entity.credslen+1);
670     strcpy(Entity.creds, tResp->tkn);
```

Memory may have been previously freed by call to free.

CodeQL [Show paths](#)

```
671     if (!Entity.name) Entity.name = strdup("anon");
672     return 0;
673 }
```

Tool	Rule ID	Query
CodeQL	cpp/use-after-free	View source

This rule finds accesses through a pointer of a memory location that has already been freed (i.e. through a dangling pointer). Such memory blocks have already been released to the dynamic memory manager, and modifying them can lead to anything from a segfault to memory

Severity
Critical

Affected branches
✓ master

Tags
reliability security

Weaknesses
🚨 CWE-416

Code Scanning with CodeQL on GitHub

The screenshot shows a GitHub CodeQL alert for a 'Potential use after free' issue. The alert is titled 'Potential use after free' and is located in the file 'XrdSecProtocolztn.cc' at line 668. The alert is marked as 'Fixed' and has a 'Dismiss alert' button. The alert is shown in a modal window with a close button (X) and a 'Source' button. The modal window contains three steps explaining the issue:

Potential use after free 6 steps in XrdSecProtocolztn.cc

Step 1 pointer to free output argument

```
src/XrdSecztn/XrdSecProtocolztn.cc:668
665     if (!tokenlib || validated)
666     {
667         Entity.credslen = strlen(tResp->tkn);
668         if (Entity.creds) free(Entity.creds);
669         Entity.creds = (char *)malloc(Entity.credslen+1);
670         strcpy(Entity.creds, tResp->tkn);
671         if (!Entity.name) Entity.name = strdup("anon");
```

Step 2 *Entity [post update] [creds]

```
src/XrdSecztn/XrdSecProtocolztn.cc:668
665     if (!tokenlib || validated)
666     {
667         Entity.credslen = strlen(tResp->tkn);
668         if (Entity.creds) free(Entity.creds);
669         Entity.creds = (char *)malloc(Entity.credslen+1);
670         strcpy(Entity.creds, tResp->tkn);
671         if (!Entity.name) Entity.name = strdup("anon");
```

Step 3 *this [post update] [Entity, creds]

file:

This rule finds acc... blocks have already been released to the dynamic memory manager, and modifying them can lead to anything from a segfault to memory

Tool: CodeQL, Rule: cpp/u...

Memory may have... CodeQL Show p...

667 En... 668 if... 669 En... 670 st... 671 if... 672 re... 673 }

Dismiss alert

Code Scanning with CodeQL on GitHub

Code scanning alerts / #3

Potential use after free

Dismiss alert ▾

Fixed in `master` last week

```
src/XrdSeczn/XrdSecProtocolztn.cc:670
```

```
667     Entity.credslen = strlen(tResp->tkn);
668     if (Entity.creds) free(Entity.creds);
669     Entity.creds = (char *)malloc(Entity.credslen+1);
670     strcpy(Entity.creds, tResp->tkn);
```

Memory may have been previously freed by [call to free](#).

CodeQL [Show paths](#)

```
671     if (!Entity.name) Entity.name = strdup("anon");
672     return 0;
673 }
```

First detected in commit 3 weeks ago

- [CMake] Update CMake minimum requirement and supported versions

src/XrdSeczn/ XrdSecProtocolztn.cc:670 on branch `master`

Fixed in branch `master` last week

- XRootD 5.7.1

Tool	Rule ID	Query
CodeQL	cpp/use-after-free	View source

This rule finds accesses through a pointer of a memory location that has already been freed (i.e. through a dangling pointer). Such memory blocks have already been released to the dynamic memory manager, and modifying them can lead to anything from a segfault to memory

reliability security

Weaknesses

🛡️ CWE-416

XRootD ABI Tracker

ABI Tracker (XRootD)

API/ABI changes review

Version	Date	Soname	Change Log	Backward Compatibility	Added Symbols	Removed Symbols	Headers Diff	Package Diff
master	2024-09-01 17:01	1/2/3	changeLog	100%	0	0	1	N/A
5.7.1	2024-09-03	1/2/3	changeLog	99.97%	4 new	1 removed	7	N/A
5.7.0	2024-07-01	1/2/3	changeLog	99.49% added 1 object	302 new	1 removed	20	N/A
5.6.9	2024-03-08	2/3	changeLog	99.73%	0	0	27	N/A
5.6.0	2023-06-30	2/3	changeLog	96.56%	7 new	0	23	N/A
5.5.5	2023-05-07	2/3	changeLog	57.92% changed SONAME added 8 objects removed 2 objects	1212 new	177 removed	113	N/A
4.12.9	2022-03-21	1/2	changeLog	N/A	N/A	N/A	N/A	N/A

Maintained by [Guilherme Amadio](#). Last updated on Wed Sep 4 12:06 2024.

Generated by [ABI Tracker](#), [ABICC](#) and [ABI Dumper](#) tools.

← → ↻ 🏠 🌐 xrootd.web.cern.ch/abi-tracker/compat_report/xrootd/5.5.5/5.6.0/8b06b/abi_compat_report.html#Type_Problems_High

Problems with Data Types, High Severity 1

XrdOfs.hh

[-] class XrdOfs 1

	Change	Effect
1	Size of this class has been increased from 704 bytes to 728 bytes.	1) An object of this class can be allocated by the applications and old size will be hardcoded at the compile time. Call of any exported constructor will break the memory of neighboring objects on the stack or heap. 2) The memory layout and size of subclasses will be changed.

[+] affected symbols: 31 (8.2%)

[to the top](#)

Problems with Data Types, Medium Severity 3

XrdOfs.hh

[-] class XrdOfs 2

	Change	Effect
1	Field dMask has been added at the middle position of this structural type.	1) Size of the inclusive type has been changed. 2) Layout of structure fields has been changed and therefore fields at higher positions of the structure definition may be incorrectly accessed by applications.
2	Field fMask has been added at the middle position of this structural type.	1) Size of the inclusive type has been changed. 2) Layout of structure fields has been changed and therefore fields at higher positions of the structure definition may be incorrectly accessed by applications.

[+] affected symbols: 31 (8.2%)

XrdSecEntity.hh

[+] class XrdSecEntity 1

[to the top](#)

Problems with Data Types, Low Severity 3

XrdOfs.hh

[+] class XrdOfs 1

XrdSecEntity.hh

[+] class XrdSecEntity 1

XrdSfsInterface.hh

[+] struct XrdSfsFScnt 1

XRootD Packaging and Distribution

- ▶ **New official XRootD DEB and RPM repositories:**

<https://xrootd.slac.stanford.edu/dload.html#official-deb-repositories>

<https://xrootd.slac.stanford.edu/dload.html#official-rpm-repositories>

- ▶ **XRootD is available via official channels in most distributions already**
 - Alma, Arch, CentOS, Debian, Fedora, Gentoo, Manjaro, Raspbian, Rocky, Ubuntu, etc
- ▶ **XRootD is also available on repositories that work across other OSs**
 - Homebrew, macPorts, Nix, Spack, conda, etc
- ▶ **Installation via official channels strongly encouraged**
- ▶ **CentOS 7 (and Python 2.x) support to be dropped likely by the end of the year**

XRootD Future Plans

- ▶ **Continuously improve overall security, stability, and performance**
 - Address problems shown by static analysis and other QA tools
 - Benchmark studies on 100G and 200G networks (CHEP 24)
 - Expand and improve current testing infrastructure
- ▶ **Plans for XRootD 6.0**
 - Initially planned for 2023/2024, has not been necessary so far
 - Existing changes to timeout handling now require ABI breakage
 - Improved error handling strategy, make error messages clearer
 - Review long term stability of HTTP support in XrdCl
 - Move to C++20 standard as baseline (base language, ranges, co-routines, std::format)
- ▶ **Ideas for new features**
 - Native plugin for OAuth2 (for use with CERN SSO, with SWAN as main use case)
 - Native support for SSH-based authentication in addition to Kerberos, GSI, etc
 - Investigate support for remote direct memory access (RDMA) with Nvidia DPUs



XRootD

XRootD 5.5.x

- ▶ **XRootD 5.5.2**
 - Support GCC 13, OpenSSL 3.x
 - Enable ZTN authentication with macaroons-based tokens
 - Extend number of parallel copies from 4 to 128 (--parallel option to xrscp)
- ▶ **XRootD 5.5.3**
 - Support user-provided script for computing checksums
- ▶ **XRootD 5.5.4**
 - ZTN plugin enabled by default
 - Fixes for authentication failures across daylight savings change
 - Support certificates with dates beyond year 2049
- ▶ **XRootD 5.5.5**
 - Enable XrdClProxy plugin to work with pgRead
 - Fix creation of zip archives with many entries
 - Fix for mixing of reused file handles coming from external table (seen on EOS AMS)

XRootD 5.6.0

▶ XRootD 5.6.0

● Server

- Make `maxfd` configurable (default is 256k)
- Use `SHA-256` by default for signatures and message digests
- Switch to a fixed set of DH parameters (compatibility with OpenSSL 1.0.2)
- Allow specification of minimum and maximum creation modes
- Better detection of private IPV6 addresses (check also for `unique local address`)
- Include token information in the monitoring stream (subject, user, vorg, role, groups)

● XCache

- New function for file eviction
- Allow origin to be a locally mounted directory (e.g. XCache for Ceph/Lustre)

● Client

- New subcommand for `xrdfs cache` to allow for cache evictions
- Do not enforce TLS when `--notlsok` option is used in combination with `root://` URL
- Increase default number of parallel event loops to 10 (affects XCache)

XRootD 5.6.x

▶ XRootD 5.6.1

- Use kernel provided uuid on macOS
- Set `RPATH` that works for binaries and libraries on macOS

▶ XRootD 5.6.2

- HTTP: Fix chunked `PUT` creating empty files
- Scitokens: Update maximum header size and line length in INI files
- Fix template for default ZTN token location
- Change the thread-id returned to OpenSSL 1.0.x to improve performance
- Insert CRLs containing critical extensions at the end of the bundle
- XrdClHttp: Add `pgWrite` support to the HTTP client plugin
- Export `readv` comma separated limits via `XRD_READV_LIMITS` environment variable
- Python: Allow build customization via environment variables (e.g. `CXX`, `CXXFLAGS`)
- Fix promotion of `root://` URLs to use TLS encryption (bug introduced in 5.6.0)

XRootD 5.6.x

▶ XRootD 5.6.3

- Create environment file within `xrd.adminpath`
- Return an error if `xrdfs` rm fails to delete any file
- Initial packet marking support in HTTP TPC

▶ XRootD 5.6.4

- Use full certificate chain for verification
- Migrate tests to GoogleTest and run without containers
- Add integrity check for headers and fix header dependency issues
- Fixes on SPARC architecture and GNU/Hurd (external contributions)
- Fix crash on `pss.origin` directive without specifying a port (uses protocol default)

▶ XRootD 5.6.5

- Support GCC 14
- Export project version in `XRootDConfig.cmake` module

XRootD 5.6.x

▶ XRootD 5.6.6

- Use full certificate chain for verification
- Migrate tests to GoogleTest and run without containers
- Add integrity check for headers and fix header dependency issues
- Fixes on SPARC architecture and GNU/Hurd (external contributions)
- Fix crash on `pss.origin` directive without specifying a port (uses protocol default)

▶ XRootD 5.6.7

- Fix crash at teardown when using copies with multiple streams
- Fix TPC initialization to take into account control stream (was always using 2 streams before)

▶ XRootD 5.6.8

- Only claim to be TLS capable if TLS initialization succeeds (`--notlsok` no longer needed)
- Create `CDash` dashboard for XRootD and enable submissions in `test.cmake`
- Fix build on FreeBSD

XRootD 5.6.x

▶ XRootD 5.6.9

- Python
 - Fix iteration over a file with Python3
 - Fix crash with raw strings in prepare call
- HTTP TPC
 - Fix 500 server response code if X-Number-Of-Streams > 100
- XrdScitokens
 - Add stat permissions to create, modify and write operations
 - Allow creation of parent directories if necessary
 - Fix bug when scope includes basepath or /