Item 34
**git_comments:**

1. coverity[ctor_dtor_leak]

**git_commits:**

1. **summary:** TS-3984: CID 1328817: Resource leaks (CTOR_DTOR_LEAK) in multiplexer plugin
   **message:** TS-3984: CID 1328817: Resource leaks (CTOR_DTOR_LEAK) in multiplexer plugin Suppressing the error

**github_issues:**

**github_issues_comments:**

**github_pulls:**

**github_pulls_comments:**

**github_pulls_reviews:**

**jira_issues:**

1. **summary:** CID 1328817: Resource leaks (CTOR_DTOR_LEAK) in multiplexer plugin
   **description:** {code} ** CID 1328817: Resource leaks (CTOR_DTOR_LEAK) /plugins/experimental/multiplexer/dispatch.cc: 39 in Request::Request(const std::basic_string<char, std::char_traits<char>, std::allocator<char>>&, tsapi_mbuffer *, tsapi_mloc *)()

   ---

   *** CID 1328817: Resource leaks (CTOR_DTOR_LEAK) /plugins/experimental/multiplexer/dispatch.cc: 39 in Request::Request(const std::basic_string<char, std::char_traits<char>, std::allocator<char>>&, tsapi_mbuffer *, tsapi_mloc *)() 33 34 extern Statistics statistics; 35 36 extern size_t timeout; 37 38 Request::Request(const std::string &h, const TSMBuffer b, const TSMLoc l) CID 1328817: Resource leaks (CTOR_DTOR_LEAK) The constructor allocates field "io" of "Request" but there is no destructor. 39 : host(h), length(TSHttpHdrLengthGet(b, l)), io(new ats::io::IO()) 40 { 41 assert(!host.empty()); 42 assert(b != NULL); 43 assert(l != NULL); 44 assert(io != NULL); {code}

**jira_issues_comments:**

1. Commit 5bb63b298755f6364cdeddce4c9a07b3136dae8c in trafficserver's branch refs/heads/master from [~bcall] [ https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=5bb63b2 ] TS-3984: CID 1328817: Resource leaks (CTOR_DTOR_LEAK) in multiplexer plugin
2. Hi, where this report comes from? is this a static analysis tool?
3. It is from coverity
4. Commit 9c83d811784b42273ed40666214dbe911ddc88cd in trafficserver's branch refs/heads/master from [~bcall] [ https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=9c83d81 ] Revert "TS-3984: CID 1328817: Resource leaks (CTOR_DTOR_LEAK) in multiplexer plugin" This reverts commit 5bb63b298755f6364cdeddce4c9a07b3136dae8c. This really isn't a leak it is deleted in another place.
5. Commit d478ae53f20f470dada60fa38b9c4e33b41bf1ac in trafficserver's branch refs/heads/master from [~bcall] [ https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=d478ae5 ] TS-3984: CID 1328817: Resource leaks (CTOR_DTOR_LEAK) in multiplexer plugin Suppressing the error
6. I'm fine with this for now, but we really ought to address this, If I understand, it copies the pointer out of the class, and then delete's the data elsewhere. That seems like a really bad anti-pattern, leading to incomprehensible code.
7. I think the best solution is to migrate to a smart point where the ownership is handled and abstracted by the class. I will work on a patch for that. Thanks for reporting the problem.
8. GitHub user dmorilha opened a pull request: https://github.com/apache/trafficserver/pull/316 TS-3984: CID 1328817: multiplexer patch to use std::auto_ptr @bcall @sc0ttbeardsley @zwoop a fix for multiplexer, I am not sure how coverity flags it... please let me know your thoughts... You can merge this pull request into a Git repository by running: $ git pull https://github.com/dmorilha/trafficserver multiplexer-patch Alternatively you can review and apply these changes as the patch at: https://github.com/apache/trafficserver/pull/316.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #316 ---- commit 04c9f66e436ec489fe434f38715c5e3cf09a6987 Author: Daniel Vitor Morilha <dmorilha@yahoo-inc.com> Date: 2015-10-30T01:42:31Z TS-3984: CID 1328817: multiplexer patch to use std::auto_ptr ----
9. Github user bryancall commented on the pull request: https://github.com/apache/trafficserver/pull/316#issuecomment-153545093 I got a bunch of errors applying this patch: dispatch.cc:46:26: error: comparison between NULL and non-pointer ('std::auto_ptr<ats::io::IO>' and NULL) [-Werror,-Wnull-arithmetic] (__builtin_expect(!(io != __null), 0) ? __assert_rtn(__func__, "dispatch.cc", 46, "io != NULL") : (void)0); ~~ ^ ~~~~~~ dispatch.cc:46:26: error: invalid operands to binary expression ('std::auto_ptr<ats::io::IO>' and 'long') (__builtin_expect(!(io != __null), 0) ? __assert_rtn(__func__, "dispatch.cc", 46, "io != NULL") : (void)0); ~~ ^ ~~~~~~ dispatch.cc:221:38: error: comparison between NULL and non-pointer ('std::auto_ptr<ats::io::IO>' and NULL) [-Werror,-Wnull-arithmetic] (__builtin_expect(!(iterator->io != __null), 0) ? __assert_rtn(__func__, "dispatch.cc", 221, "iterator->io != NULL") : (void)0); ~~~~~~~~~~~~ ^ ~~~~~~ dispatch.cc:221:38: error: invalid operands to binary expression ('std::auto_ptr<ats::io::IO>' and 'long') (__builtin_expect(!(iterator->io != __null), 0) ? __assert_rtn(__func__, "dispatch.cc", 221, "iterator->io != NULL") : (void)0); ~~~~~~~~~~~~ ^ ~~~~~~ dispatch.cc:234:38: error: comparison between NULL and non-pointer ('std::auto_ptr<ats::io::IO>' and NULL) [-Werror,-

Wnull-arithmetic] (__builtin_expect(!(iterator->io != __null), 0) ? __assert_rtn(__func__, "dispatch.cc", 234, "iterator->io != NULL") : (void)0); ~~~~~~~~~~~~ ^ ~~~~~~ dispatch.cc:234:38: error: invalid operands to binary expression ('std::auto_ptr<ats::io::IO>' and 'long') (__builtin_expect(!(iterator->io != __null), 0) ? __assert_rtn(__func__, "dispatch.cc", 234, "iterator->io != NULL") : (void)0); ~~~~~~~~~~~~~ ^ ~~~~~~ In file included from dispatch.cc:26: In file included from ./dispatch.h:27: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../include/c++/v1/memory:1673:31: error: no matching constructor for initialization of 'Request' ::new((void*)__p) _Up(std::__1::forward<_Args>(__args)...); ^ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../include/c++/v1/memory:1600:18: note: in instantiation of function template specialization 'std::__1::allocator<Request>::construct<Request, Request>' requested here {__a.construct(__p, std::__1::forward<_Args>(__args)...);} ^ /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../include/c++/v1/memory:1453:14: note: in instantiation of function template specialization 'std::__1::allocator_traits<std::__1::allocator<Request> >::__construct<Request, Request>' requested here {__construct(__has_construct<allocator_type, _Tp*, _Args...>(), ^ /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../include/c++/v1/vector:1609:25: note: in instantiation of function template specialization 'std::__1::allocator_traits<std::__1::allocator<Request> >::construct<Request, Request>' requested here __alloc_traits::construct(this->__alloc(), ^ dispatch.cc:205:7: note: in instantiation of member function 'std::__1::vector<Request, std::__1::allocator<Request> >::push_back' requested here r.push_back(Request(host, buffer, location)); ^ ./dispatch.h:51:8: note: candidate constructor (the implicit copy constructor) not viable: expects an l-value for 1st argument struct Request { ^ dispatch.cc:38:10: note: candidate constructor not viable: requires 3 arguments, but 1 was provided Request::Request(const std::string &h, const TSMBuffer b, const TSMLoc l)

10. Github user dmorilha commented on the pull request: https://github.com/apache/trafficserver/pull/316#issuecomment-153567597 In our internal version everything worked fine. I will check it tomorrow. Thanks for reporting.
11. Github user bryancall commented on the pull request: https://github.com/apache/trafficserver/pull/316#issuecomment-153580444 Compile with -Werror or look at the warnings.
12. Github user dmorilha commented on the pull request: https://github.com/apache/trafficserver/pull/316#issuecomment-154508604 I think I got it all fixed now.
13. Github user asfgit closed the pull request at: https://github.com/apache/trafficserver/pull/316
14. [~dmorilha] I was able to remove the const_cast from Request &Request::operator=(Request &r) by removing the const in the incoming argument. However, I get errors when trying to do the same with Request::Request(const Request &r). I will commit as is, but if you have a suggestion on getting rid of the const_cast I think it would be cleaner code. Here is the error I am getting: {code} In file included from dispatch.cc:26:0: dispatch.h:54:8: warning: 'template<class> class std::auto_ptr' is deprecated [-Wdeprecated-declarations] std::auto_ptr<ats::io::IO> io; ^ In file included from /usr/include/c++/5.1.1/memory:81:0, from dispatch.h:27, from dispatch.cc:26: /usr/include/c++/5.1.1/bits/unique_ptr.h:49:28: note: declared here template<typename> class auto_ptr; ^ In file included from /usr/include/c++/5.1.1/x86_64-redhat-linux/bits/c++allocator.h:33:0, from /usr/include/c++/5.1.1/bits/allocator.h:46, from /usr/include/c++/5.1.1/memory:63, from dispatch.h:27, from dispatch.cc:26: /usr/include/c++/5.1.1/ext/new_allocator.h: In instantiation of 'void __gnu_cxx::new_allocator<_Tp>::construct(_Up*, _Args&& ...) [with _Up = Request; _Args = {Request}; _Tp = Request]': /usr/include/c++/5.1.1/bits/alloc_traits.h:256:4: required from 'static std::_Require<typename std::allocator_traits<_Alloc>::__construct_helper<_Tp, _Args>::type> std::allocator_traits<_Alloc>::_S_construct(_Alloc&, _Tp*, _Args&& ...) [with _Tp = Request; _Args = {Request}; _Alloc = std::allocator<Request>; std::_Require<typename std::allocator_traits<_Alloc>::__construct_helper<_Tp, _Args>::type> = void]' /usr/include/c++/5.1.1/bits/alloc_traits.h:402:16: required from 'static decltype (_S_construct(__a, __p, (forward<_Args>)(std::allocator_traits::construct::__args)...)) std::allocator_traits<_Alloc>::construct(_Alloc&, _Tp*, _Args&& ...) [with _Tp = Request; _Args = {Request}; _Alloc = std::allocator<Request>; decltype (_S_construct(__a, __p, (forward<_Args>)(std::allocator_traits::construct::__args)...)) = <type error>]' /usr/include/c++/5.1.1/bits/vector.tcc:96:30: required from 'void std::vector<_Tp, _Alloc>::emplace_back(_Args&& ...) [with _Args = {Request}; _Tp = Request; _Alloc = std::allocator<Request>]' /usr/include/c++/5.1.1/bits/stl_vector.h:932:21: required from 'void std::vector<_Tp, _Alloc>::push_back(std::vector<_Tp, _Alloc>::value_type&&) [with _Tp = Request; _Alloc = std::allocator<Request>; std::vector<_Tp, _Alloc>::value_type = Request]' dispatch.cc:223:48: required from here /usr/include/c++/5.1.1/ext/new_allocator.h:120:4: error: invalid initialization of non-const reference of type 'Request&' from an rvalue of type 'Request' { ::new((void *)__p) _Up(std::forward<_Args>(__args)...); } ^ dispatch.cc:50:1: note: initializing argument 1 of 'Request::Request(Request&)' Request::Request(Request &r) : host(r.host), length(r.length), io(r.io) ^ In file included from /usr/include/c++/5.1.1/memory:64:0, from dispatch.h:27, from dispatch.cc:26: /usr/include/c++/5.1.1/bits/stl_construct.h: In instantiation of 'void std::_Construct(_T1*, _Args&& ...) [with _T1 = Request; _Args = {Request}]': /usr/include/c++/5.1.1/bits/stl_uninitialized.h:75:18: required from 'static _ForwardIterator std::__uninitialized_copy<_TrivialValueTypes>::__uninit_copy(_InputIterator, _InputIterator, _ForwardIterator) [with _InputIterator = std::move_iterator<Request*>; _ForwardIterator = Request*; bool _TrivialValueTypes = false]' /usr/include/c++/5.1.1/bits/stl_uninitialized.h:126:15: required from '_ForwardIterator std::uninitialized_copy(_InputIterator, _InputIterator, _ForwardIterator) [with _InputIterator = std::move_iterator<Request*>; _ForwardIterator = Request*]' /usr/include/c++/5.1.1/bits/stl_uninitialized.h:281:37: required from '_ForwardIterator std::__uninitialized_copy_a(_InputIterator, _InputIterator, _ForwardIterator, std::allocator<_Tp>&) [with _InputIterator = std::move_iterator<Request*>; _ForwardIterator = Request*; _Tp = Request]' /usr/include/c++/5.1.1/bits/stl_uninitialized.h:303:2: required from '_ForwardIterator std::__uninitialized_move_if_noexcept_a(_InputIterator, _InputIterator, _ForwardIterator, _Allocator&) [with _InputIterator = Request*; _ForwardIterator = Request*; _Allocator = std::allocator<Request>]' /usr/include/c++/5.1.1/bits/vector.tcc:422:8: required from 'void std::vector<_Tp, _Alloc>::_M_emplace_back_aux(_Args&& ...) [with _Args = {Request}; _Tp = Request; _Alloc = std::allocator<Request>]' /usr/include/c++/5.1.1/bits/vector.tcc:101:23: required from 'void std::vector<_Tp, _Alloc>::emplace_back(_Args&& ...) [with _Args = {Request}; _Tp = Request; _Alloc = std::allocator<Request>]' /usr/include/c++/5.1.1/bits/stl_vector.h:932:21: required from 'void std::vector<_Tp, _Alloc>::push_back(std::vector<_Tp, _Alloc>::value_type&&) [with _Tp = Request; _Alloc = std::allocator<Request>; std::vector<_Tp, _Alloc>::value_type =

Request]' dispatch.cc:223:48: required from here /usr/include/c++/5.1.1/bits/stl_construct.h:75:7: error: invalid initialization of non-const reference of type 'Request&' from an rvalue of type 'Request' { ::new(static_cast<void*>(__p)) _T1(std::forward<_Args>(__args)...); } ^ dispatch.cc:50:1: note: initializing argument 1 of 'Request::Request(Request&)' Request::Request(Request &r) : host(r.host), length(r.length), io(r.io) ^ Makefile:738: recipe for target 'dispatch.lo' failed make[3]: *** [dispatch.lo] Error 1 make[3]: Leaving directory '/home/bcall/dev/apache/trafficserver/plugins/experimental/multiplexer' Makefile:625: recipe for target 'all-recursive' failed make[2]: *** [all-recursive] Error 1 make[2]: Leaving directory '/home/bcall/dev/apache/trafficserver/plugins/experimental' Makefile:615: recipe for target 'all-recursive' failed make[1]: *** [all-recursive] Error 1 make[1]: Leaving directory '/home/bcall/dev/apache/trafficserver/plugins' Makefile:670: recipe for target 'all-recursive' failed make: *** [all-recursive] Error 1 {code}

15. Commit 9f4d1d1e3d13ae3af59b3fb483a419c3ddb4a56d in trafficserver's branch refs/heads/master from [~bcall] [ https://git-wip-us.apache.org/repos/asf?p=trafficserver.git;h=9f4d1d1 ] TS-3984: CID 1328817: Resource leaks (CTOR_DTOR_LEAK) in multiplexer plugin Put back const for the assignment operator
16. Looks like I might have to add const back into the assignment operator because of ubuntu_12_04 and centos_6. Building again and hopefully it will build successfully.