Project – Second part, code check of an open-source project

Ondrej Mosnáček, Lenka Kuníková and Ľubomír Obrátil

16. decembra 2015

The project

- **ZeroTier One** (https://github.com/zerotier/ZeroTierOne) – a secure network virtualization project.
- ∼55 000 lines of code

CppCheck

Statistics

Errors:	4
Warnings:	80
Style warnings:	218
Portability warnings:	1
Performance warnings:	3
Information messages:	130

Errors

- `controller/SqliteNetworkController.cpp:1012` – Exception thrown in function declared not to throw exceptions.
- `osdep/UPNPClient.cpp:100` – Exception thrown in function declared not to throw exceptions.
- `osdep/WindowsEthernetTap.cpp:876` – Exception thrown in function declared not to throw exceptions.
- `service/OneService.cpp` – Used file that is not opened.

Warnings

- Member variable '…' is not initialized in the constructor.
- `operator=` should check for assignment to self to ensure that each block of dynamically allocated memory is owned and managed by only one instance of the class.
- Member variable `_Bucket::next` is not assigned a value in `_Bucket::operator=`.
- `%d` in format string (no. 2) requires `int` but the argument type is `unsigned int`.
- Repositioning operation performed on a file opened in append mode has no effect.
- Either the condition `p && intClient` is redundant or there is possible null pointer dereference: `intClient`.
- The return value of `std::unique()` is ignored. This function returns an iterator to the end of the range containing those elements that should be kept. Elements past new end remain valid but with unspecified values. Use the erase method of the container to delete them.
- `%llx` in format string (no. 1) requires `unsigned long long` but the argument type is `unsigned int`.

Portability warnings

- Using `memset()` on union which contains a floating point number. This is not portable because `memset()` sets each byte of a block of memory to a specific value and the actual representation of a floating-point value is implementation defined. Note: In case of an IEEE754-1985 compatible implementation setting all bits to zero results in the value 0.0.

Performance warnings

- When an object of a class is created, the constructors of all member variables are called consecutively in the order the variables are declared, even if you don't explicitly write them to the initialization list. You could avoid assigning '...' a value by passing the value to the constructor in the initialization list.

Style warnings

- The scope of the variable '...' can be reduced. Warning: Be careful when fixing this message, especially when there are inner loops.
- Variable '...' is assigned a value that is never used.
- Class '...' has a constructor with 1 argument that is not explicit. Such constructors should in general be explicit for type safety reasons. Using the explicit keyword in the constructor means some mistakes when using the class can be avoided.
- The unsigned variable '...' will never be negative so it is either pointless or an error to check if it is.
- struct member '...' is never used.
- Suspicious calculation. Please use parentheses to clarify the code. The code `a & b ? c : d` should be written as either `(a & b) ? c : d` or `a & (b ? c : d)`.
- Unused variable: ...
- C-style pointer casting detected. C++ offers four different kinds of casts as replacements: `static_cast`, `const_cast`, `dynamic_cast` and `reinterpret_cast`. A C-style cast could evaluate to any of those automatically, thus it is considered safer if the programmer explicitly states which kind of cast is expected. See also: https://www.securecoding.cert.org/confluence/display/cplusplus/EXP05-CPP.+Do+not+use+C-style+casts.
- class `Cluster` does not have a copy constructor which is recommended since the class contains a pointer to allocated memory.
- Defensive programming: The variable `k` is used as an array index before it is checked that it is within limits. This can mean that the array might be accessed out of bounds. Reorder conditions such as `(a[i] && i < 10)` to `(i < 10 && a[i])`. That way the array will not be accessed if the index is out of limits.
- Consecutive return, break, continue, goto or throw statements are unnecessary. The second statement can never be executed, and so should be removed.
- The exception is caught by value. It could be caught as a (const) reference which is usually recommended in C++.

MSVC

Statistics

Errors: 1
Warnings: 154
(All only in external libraries.)

Errors

- `ext/miniupnpc/minissdpc.c:535` – error C4996: `inet_addr`: Use `inet_pton()` or `InetPton()` instead or define [...] to disable deprecated API warnings

Warnings

- C4245 - conversion from `int` to `unsigned int` in return statement
- C4204 - nonstandard extension used: non-constant aggregate initializer
- C4132 - const object should be initialized
- C4100 - unreferenced function parameter
- C4244 - conversion from `unsigned int` to `char` (possible loss of information)
- C4706 - assignment within conditional expression

PREfast

Detected nothing.