
Subject: Possibly memory leak SSGM

Posted by [Neijwiert](#) on Thu, 01 May 2014 17:47:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

So I was going through some code and I found my way to the file gmlog.cpp

The logger keeps track of the current socket connections it has in a SimpleDynVecClass list. However this class does not use the delete operator on its contents when you call ::Delete.

So I found these 2 leaks:

- in void SSGMGameLog::Think()

where you gracefully close the sockets upon error or something else you call Connections.Delete(Connections[i]); this does not however delete the resources to Connections[i].

- in void SSGMGameLog::Shutdown()

you loop through all connections and close the sockets, however no memory is released through this process.

All objects inside this list are created with the new operator, however no subsequent delete operators are called.

I might be completely wrong at this, but anyway it's worth reporting I suppose...

EDIT:

Further explanation:

```
bool Delete(int index,bool allow_shrink = true)
{
    if (index < ActiveCount-1)
    {
        memmove(&(Vector[index]),&(Vector[index+1]),(ActiveCount - index - 1) * sizeof(T));
    }
    ActiveCount--;
    if (allow_shrink)
    {
        Shrink();
    }
    return true;
}
bool Delete(T const & object,bool allow_shrink = true)
{
    int id = Find_Index(object);
    if (id != -1)
```

```
{
  return Delete(id,allow_shrink);
}
return false;
}
```

these are the 2 overloaded functions inside SimpleDynVecClass. As you can see it only uses memmove which doesn't guarantee that the resources are overwritten. The code in gmlog.cpp calls the bottom Delete first, which then calls the top one if there's an index found.
