

Serious Memory Leaks with VDDK 1.1 on ...



Mitch Haile 7 posts since

Nov 27, 2007

It appears in our testing that there exist serious memory leaks in VDDK on Linux 32 bit. Any pointers would be appreciated.

A sample program is at the end of this message. With calls to VixDiskLib_Open() and Close(), no reading of data, each call to Open leaks a good chunk of memory. The below valgrind output is from a single Open/Close pair.

Over 1000 errors are reported, mostly out of libxml2 and libcurl.

```
==715==
==715== ERROR SUMMARY: 39170 errors from 1000 contexts (suppressed: 42 from 1)
==715== malloc/free: in use at exit: 383,967 bytes in 8,033 blocks.
==715== malloc/free: 19,217 allocs, 11,184 frees, 2,394,652 bytes allocated.
==715== For counts of detected errors, rerun with: -v
==715== searching for pointers to 8,033 not-freed blocks.
==715== checked 2,474,068 bytes.
==715==
==715== LEAK SUMMARY:
==715== definitely lost: 366,477 bytes in 7,944 blocks.
==715== possibly lost: 0 bytes in 0 blocks.
==715== still reachable: 17,490 bytes in 89 blocks.
==715== suppressed: 0 bytes in 0 blocks.
==715== Rerun with --leak-check=full to see details of leaked memory.
```

Without the call to VixDiskLib_Open() and Close():

```
==687==
==687== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 25 from 1)
==687== malloc/free: in use at exit: 39,651 bytes in 1,757 blocks.
==687== malloc/free: 1,970 allocs, 213 frees, 137,174 bytes allocated.
==687== For counts of detected errors, rerun with: -v
==687== searching for pointers to 1,757 not-freed blocks.
==687== checked 2,493,324 bytes.
==687==
==687== LEAK SUMMARY:
==687== definitely lost: 0 bytes in 0 blocks.
==687== possibly lost: 0 bytes in 0 blocks.
```

Serious Memory Leaks with VDDK 1.1 on ...

==687== still reachable: 39,651 bytes in 1,757 blocks.

==687== suppressed: 0 bytes in 0 blocks.

==687== Rerun with --leak-check=full to see details of leaked memory.

Here is a simple test case:

```
#include <string.h>
#include <stdio.h>
#include <vixDiskLib.h>

#define VIXDISKLIB_VERSION_MAJOR 1
#define VIXDISKLIB_VERSION_MINOR 0
#define VDDK_DISK_OPEN 1

static void
panic_if_vixerror(VixError vixError, const char *func_name)
{
    if (VIX_FAILED(vixError)) {
        fprintf(stderr, "%s failed: %s\n", func_name, VixDiskLib_GetErrorText(vixError, NULL));
        exit(1);
    }

    fprintf(stderr, "%s: OK\n", func_name);
}

static void
init_connect_params(VixDiskLibConnectParams *cp)
{
    (void) memset(cp, 0, sizeof(*cp));

    cp->vmxSpec = NULL;
    cp->serverName = "192.168.201.106";
    cp->credType = VIXDISKLIB_CRED_UID;
    cp->creds.uid.userName = "root";
    cp->creds.uid.password = "abc123";
    cp->port = 902;
}

int
main(int argc, char **argv)
{
    (void) argv;

    VixError vixError;
```

Serious Memory Leaks with VDDK 1.1 on ...

```
VixDiskLibConnection conn = NULL;
VixDiskLibConnectParams cp;

if (argc <= 1) {
    fprintf(stderr, "Usage: $0 <vmdk path>\n");
    return 1;
}

init_connect_params(&cp);

vixError = VixDiskLib_Init(VIXDISKLIB_VERSION_MAJOR,
VIXDISKLIB_VERSION_MINOR,
NULL, NULL, NULL, // Log, warn, panic
NULL); // libDir
panic_if_vixerror(vixError, "VixDiskLib_Init");

vixError = VixDiskLib_Connect(&cp, &conn);
panic_if_vixerror(vixError, "VixDiskLib_Connect");

#if VDDK_DISK_OPEN
{
    const char *disk_name = argv[1];
    VixDiskLibHandle disk_handle = NULL;

    VixError vixError = VixDiskLib_Open(conn, disk_name, VIXDISKLIB_FLAG_OPEN_READ_ONLY, &disk_handle);
    panic_if_vixerror(vixError, "VixDiskLib_Open");

    vixError = VixDiskLib_Close(disk_handle);
    panic_if_vixerror(vixError, "VixDiskLib_Close");
}
#endif

if (conn != NULL) {
    vixError = VixDiskLib_Disconnect(conn);
    panic_if_vixerror(vixError, "VixDiskLib_Disconnect");
    conn = NULL;
}

VixDiskLib_Exit();

printf("--done--\n");
return 0;
}
```

It doesn't seem to matter if VIXDISKLIB_VERSION_MINOR is 0 or 1.

Serious Memory Leaks with VDDK 1.1 on ...

The code is built with:

```
$ gcc -I/usr/lib/vmware-vix-disklib/include -L/usr/lib/vmware-vix-disklib/lib32 -g -lvixDiskLib -o testvddk test_vddk.c
```

Attachments:

- [test_vddk.c](#) (2.0 K)

Tags: memory, leak, vddk



[sudarsan](#) 263 posts since

Sep 20, 2006 1. **Re: Serious Memory Leaks with VDDK 1.1 on Linux** Jul 5, 2009 10:17 PM

Thanks for the message, let me take a look.

-Sudarsan



[unideskjohn](#) 9 posts since

Mar 3, 2008 2. **Re: Serious Memory Leaks with VDDK 1.1 on Linux** Jul 14, 2009 11:18 AM

👤 in response to: [sudarsan](#) Any update on this? We are seeing some significant memory leaks from the VIX libraries and the libraries it uses.



[Mitch Haile](#) 7 posts since

Nov 27, 2007 3. **Re: Serious Memory Leaks with VDDK 1.1 on Linux** Jul 14, 2009 11:25 AM

👤 in response to: [unideskjohn](#) unideskjohn - Are you seeing memory leaks beyond just the Disk Open? I'd be interested to know what else is leaking so we can be on the lookout.

-Mitch

mitch.haile@gmail.com

408-850-0129



[unideskjohn](#) 9 posts since

Mar 3, 2008 4. **Re: Serious Memory Leaks with VDDK 1.1 on Linux** Jul 15, 2009 7:19 AM

👤 in response to: [Mitch Haile](#) I believe most of the memory leaks we are seeing are a result of the Disk Open call. However in the run we were profiling we were doing read and writes as well. Its a little hard to tell due to the vast number of complaints being spewed by valgrind. After seeing all the complaints I came to the forums to see if anyone else had seen the problem and came across your post. We have not tried to narrow it down farther as you have done.

John



[sudarsan](#) 263 posts since

Sep 20, 2006 5. **Re: Serious Memory Leaks with VDDK 1.1 on Linux** Jul 29, 2009 5:33 PM

👤 in response to: [unideskjohn](#)

I apologize for being so delayed, I have been working against a deadline but should be able to let you know more about this issue this week,


Serious Memory Leaks with VDDK 1.1 on ...

Thanks for your patience,
Sudarsan



UniBill 1 posts since

Aug 4, 2009 6. **Re: Serious Memory Leaks with VDDK 1.1 on Linux** Aug 4, 2009 7:52 AM

 in response to: [sudarsan](#)

Any update on this issue? I'm seeing even more memory leaks on further calls to this library.

==9440== LEAK SUMMARY:

==9440== definitely lost: 17,202 bytes in 36 blocks.

==9440== indirectly lost: 422,229 bytes in 9,384 blocks.

==9440== possibly lost: 0 bytes in 0 blocks.

==9440== still reachable: 82,564 bytes in 90 blocks.

==9440== suppressed: 0 bytes in 0 blocks.

==9440== Reachable blocks (those to which a pointer was found) are not shown.

==9440== To see them, rerun with: --leak-check=full --show-reachable=yes

The biggest difference between my test program and Mitch's above is that I also called `VixDiskLib_GetInfo()` after the `VixDiskLib_Open()` call, and then a corresponding `VixDiskLib_FreeInfo()` call right before the `VixDiskLib_Close()` call.

I haven't run valgrind against IO calls yet, but that's a lot of memory to lose on just an init, connect, getinfo, freeinfo, close, disconnct, exit chain of function calls.