New issue

Jump to bottom

Null pointer caused by UAF in unicorn-1.0.3 #1578

Closed liyansong2018 opened this issue on Apr 1 · 5 comments

Labels question stale

liyansong2018 commented on Apr 1 • edited Contributor

Hello, :)

Unicorn-1.0.3 Python API

- Add `emu_start` to the hook function (uc_hook_mem_read_unmapped)
- Releasing the hook function will cause the null pointer dereference.

PoC is as follows

```
def disasm(cont, addr):
    md = Cs(CS_ARCH_X86, CS_MODE_64)
    for i in md.disasm(cont, addr):
        print("0x%x:\t%s\t%s" % (i.address, i.mnemonic, i.op_str))

def read(name):
    with open(name, "rb") as fp:
        return fp.read()

def hook_unmapped(uc, access, address, size, value, user_data):
    print('>>> Tracing instruction at 0x%x, instruction size = 0x%x' %(address, size))
    print(os.getpid())
    #time.sleep(10)
    ######### poc #########
    uc.emu_start(address + size, 0x400582)
    ######### poc #########

insn_skip_list = [0x4004ef, 0x4004f6, 0x400502, 0x40054f, 0x400560]
def hook_code(mu, address, size, user_data):
    disasm(mu.mem_read(address, size), address)

def main():
    uc = Uc(UC_ARCH_X86, UC_MODE_64)
```

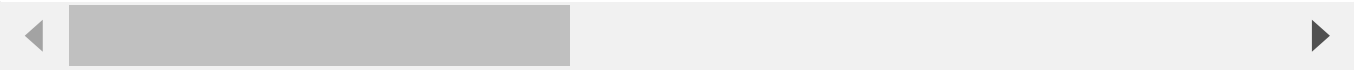
```
    uc.mem_map(0x400000, 1024 * 1024)    # binary
    uc.mem_map(0x100000, 1024 * 1024)        # stack
    uc.mem_write(0x400000, read("./fibonacci"))
    uc.reg_write(UC_X86_REG_RSP, 0x100000 + 1024 * 1024 - 1)

    uc.hook_add(UC_HOOK_CODE, hook_code)
    uc.hook_add(UC_HOOK_MEM_READ_UNMAPPED, hook_unmapped)
    #uc.hook_add(UC_ERR_FETCH_UNMAPPED, unicorn_hook_add)

    try:
        uc.emu_start(0x4004E0, 0x400582)
    except UcError as e:
        print(e)

if __name__ == "__main__":
    main()
```

segmentation fault

```
└─$ python3 poc.py
0x4004e0:       push    rbp
0x4004e1:       push    rbx
0x4004e2:       xor     esi, esi
0x4004e4:       mov     ebp, 0x4007e1
0x4004e9:       xor     ebx, ebx
0x4004eb:       sub     rsp, 0x18
0x4004ef:       mov     rdi, qword ptr [rip + 0x200b42]
>>> Tracing instruction at 0x601038, instruction size = 0x8
17303
zsh: segmentation fault  python3 poc.py
```

gdb debug

```
gef➤  bt
#0  0x0000000000000000 in ?? ()
#1  0x00000000013fdd38 in ?? ()
#2  0x00007f9e1eac5269 in clear_deleted_hooks (uc=0x13fd620) at /home/lys/Documents/unicorn-1.0.3/uc.
#3  uc_emu_start (uc=0x13fd620, begin=<optimized out>, until=<optimized out>, timeout=0x0, count=<opt
#4  0x00007f9e1fa12ccd in ?? () from /lib/x86_64-linux-gnu/libffi.so.7
#5  0x00007f9e1fa1225a in ?? () from /lib/x86_64-linux-gnu/libffi.so.7
#6  0x00000000004f8209 in _PyObject_GenericSetAttrWithDict ()
#7  0x00000000004f7d8a in PyObject_SetAttr ()
#8  0x0000000000505fa5 in _PyEval_EvalFrameDefault ()
#9  0x000000000050f9c0 in _PyObject_FastCallDict ()
#10 0x0000000000526a61 in ?? ()
#11 0x00007f9e1fa1202c in ?? () from /lib/x86_64-linux-gnu/libffi.so.7
#12 0x00007f9e1fa1225a in ?? () from /lib/x86_64-linux-gnu/libffi.so.7
#13 0x00007f9e1fb9960a in _ctypes_callproc () from /usr/lib/python3.8/lib-dynload/_ctypes.cpython-38-
#14 0x00000000008c3bc0 in ?? ()
#15 0x00007f9e1fa8ed70 in ?? ()
#16 0x00007f9e1fb027c0 in ?? ()
#17 0x00007f9e1fa9b680 in ?? ()
#18 0x0000000000000000 in ?? ()
```



aquynh commented on Apr 1 — Member

Please try with "dev" branch

liyansong2018 commented on Apr 1 • edited — Contributor, Author

Unicorn Python package supports up to version 1.0.3

```
[Latest version](https://pypi.org/project/unicorn/)
Released: May 27, 2021
unicorn 1.0.3
```

I also tried to use `libunicorn.so.2` generated by compiling unicorn dev branch replace `libunicorn.so` (1.0.3) installed by PIP command. But unicorn 2 doesn't seem to work well in the latest version of the unicorn python API.

I also modified the `unicorn.py` to try to bypass `unicorn.py` detection of the unicorn version, but there will still be compatibility problems.

```
# unicorn.py
# verify version compatibility with the core before doing anything
(major, minor, _combined) = uc_version()
'''
if major != uc.UC_API_MAJOR or minor != uc.UC_API_MINOR:
    self._uch = None
    # our binding version is different from the core's API version
    raise UcError(uc.UC_ERR_VERSION)
'''
```

liyansong2018 commented on Apr 2 • edited — Contributor, Author

I may know the cause of the problem.

`0x00000000013fdd38 -> uc->hooks_to_del->head .`

When we use `uc_emu_start` for the second time, unicorn is already in an abnormal state (`Invalid memory read (UC_ERR_READ_UNMAPPED)`), so it will continue run to the following code

```
// uc.c  uc_emu_start
if (uc->vm_start(uc)) {
    return UC_ERR_RESOURCE;
}

// emulation is done
uc->emulation_done = true;

// remove hooks to delete
clear_deleted_hooks(uc);
```

remove hooks to delete!

However, our PoC is still in the first `uc_emu_start->vm_start` environment, and QEMU TCG is still trying to execute hook code. In fact, the hook code has been released.

wtdcode commented on Apr 2 Member

> Unicorn Python package supports up to version 1.0.3
>
> ```
> [Latest version](https://pypi.org/project/unicorn/)
> Released: May 27, 2021
> unicorn 1.0.3
> ```
>
> I also tried to use `libunicorn.so.2` generated by compiling unicorn dev branch replace `libunicorn.so` (1.0.3) installed by PIP command. But unicorn 2 doesn't seem to work well in the latest version of the unicorn python API.
>
> I also modified the `unicorn.py` to try to bypass `unicorn.py` detection of the unicorn version, but there will still be compatibility problems.
>
> ```
> # unicorn.py
> # verify version compatibility with the core before doing anything
> (major, minor, _combined) = uc_version()
> '''
> if major != uc.UC_API_MAJOR or minor != uc.UC_API_MINOR:
>     self._uch = None
>     # our binding version is different from the core's API version
>     raise UcError(uc.UC_ERR_VERSION)
> '''
> ```

`pip3 install --pre unicorn` will do the trick.

  **wtdcode** added the question label on Apr 3

github-actions bot commented on Jun 3

This issue is stale because it has been open 60 days with no activity. Remove stale label or comment or this will be closed in 15 days.

github-actions bot added the **stale** label on Jun 3

github-actions bot closed this as completed on Jun 19

Assignees

No one assigned

Labels

question **stale**

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants