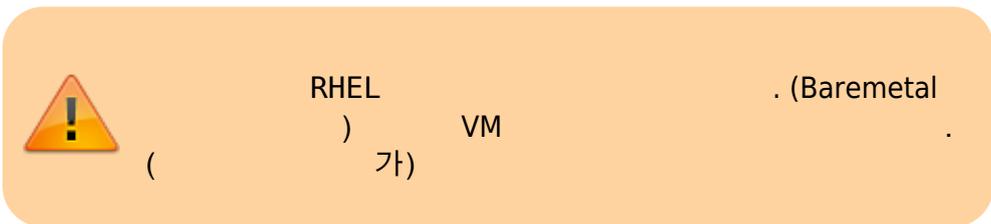# Host USB Passthrough to RHEV Guests

RHEV 3.6

3.6   .

- https://access.redhat.com/documentation/en-us/red_hat_enterprise_virtualization/3.6/html/virtual_machine_management_guide/sect-host_devices

RHEV 3.2

⚠️　　　　　　　　　　RHEL　　　　　　　　　　　　　. (Baremetal
　　　　　　　　　)　　　VM　　　　　　　　　　　　　　　.
　(　　　　　　　　　　　　　)

- RHEL　　　　　　rhev　　　　　　USB　　　　　　　　　　　　　　　　　　　　.

USB　　　RHEL　　　　　　lsusb　　　　　　　vendor:product id　　　　.

```
# lsusb
Bus 002 Device 008: ID 0781:5567 SanDisk Corp. Cruzer Blade
```

[hostusb](https://atl.kr/dokuwiki/)　　　　　rhel　　　before_vm_start.py
/usr/libexec/vdsm/hooks/before_vm_start/　　60_hostusb　　755
　　.

[before_vm_start.py](https://atl.kr/dokuwiki/)

```python
#!/usr/bin/python

import re
import os
import sys
import grp
import pwd
import traceback

import hooking

'''
host usb hook
=============
```

```
          !!! Disclaimer !!!
*****************************************
The host side usb support wasn't thoroughly
tests in kvm!
*****************************************


add hosts usb device/s to VM:

<hostdev mode='subsystem' type='usb'>
    <source>
        <vendor id='0x1234'/>
        <product id='0xbeef'/>
    </source>
</hostdev>

syntax:
    hostusb=0x1234:0xbeef&0x2222:0xabaa
    i.e.
    hostusb=VendorId:ProductId (can add more then one with '&' separator)

Note:
    The VM must be pinned to host and this hook will
    fail any migration attempt.
'''


HOOK_HOSTUSB_PATH = '/var/run/vdsm/hooks/hostusb-permissions'

def log_dev_owner(devpath, user, group):
    entry = devpath + ":" + str(user) + ":" + str(group)

    if not os.path.isdir(os.path.dirname(HOOK_HOSTUSB_PATH)):
        os.mkdir(os.path.dirname(HOOK_HOSTUSB_PATH))

    if os.path.isfile(HOOK_HOSTUSB_PATH):
        f = file(HOOK_HOSTUSB_PATH, 'r')
        for line in f:
            if entry == line:
                f.close()
                return

    f = file(HOOK_HOSTUSB_PATH, 'a')
    f.writelines(entry)
    f.close()


#!TODO:
# merge chown with after_vm_destroy.py
# maybe put it in hooks.py?
def chown(vendorid, productid):

    # remove the 0x from the vendor and product id
```

```python
    devid = vendorid[2:] + ':' + productid[2:]
    command = ['lsusb', '-d', devid]
    retcode, out, err = hooking.execCmd(command, sudo=False, raw=True)
    if retcode != 0:
        sys.stderr.write('hostusb: cannot find usb device: %s\n' % devid)
        sys.exit(2)

    # find the device path:
    # /dev/bus/usb/xxx/xxx
    devpath = '/dev/bus/usb/' + out[4:7] + '/' + out[15:18]
    stat = os.stat(devpath)

    group = grp.getgrnam('qemu')
    gid = group.gr_gid
    user = pwd.getpwnam('qemu')
    uid = user.pw_uid

    # we don't use os.chown because we need sudo
    owner = str(uid) + ':' + str(gid)
    command = ['/bin/chown', owner, devpath]
    retcode, out, err = hooking.execCmd(command, sudo=True, raw=True)
    if retcode != 0:
        sys.stderr.write('hostusb: error chown %s to %s, err = %s\n' %
(devpath, owner, err))
        sys.exit(2)

    log_dev_owner(devpath, stat.st_uid, stat.st_gid)

def create_usb_device(domxml, vendorid, productid):
    hostdev = domxml.createElement('hostdev')
    hostdev.setAttribute('mode', 'subsystem')
    hostdev.setAttribute('type', 'usb')

    source = domxml.createElement('source')
    hostdev.appendChild(source)

    vendor = domxml.createElement('vendor')
    vendor.setAttribute('id', vendorid)
    source.appendChild(vendor)

    product = domxml.createElement('product')
    product.setAttribute('id', productid)
    source.appendChild(product)

    return hostdev

if os.environ.has_key('hostusb'):
    try:
        regex = re.compile('^0x[\d,A-F,a-f]{4}$')
        domxml = hooking.read_domxml()
        devices = domxml.getElementsByTagName('devices')[0]
```

```
        for usb in os.environ['hostusb'].split('&'):
            vendorid, productid = usb.split(':')
            if len(regex.findall(vendorid)) != 1 or
len(regex.findall(productid)) != 1:
                sys.stderr.write('hostusb: bad input, expected 0x0000 format
for vendor and product id, input: %s:%s\n' % (vendorid, productid))
                sys.exit(2)

            hostdev = create_usb_device(domxml, vendorid, productid)
            devices.appendChild(hostdev)
            chown(vendorid, productid)

        hooking.write_domxml(domxml)
    except:
        sys.stderr.write('hostusb: [unexpected error]: %s\n' %
traceback.format_exc())
        sys.exit(2)
```

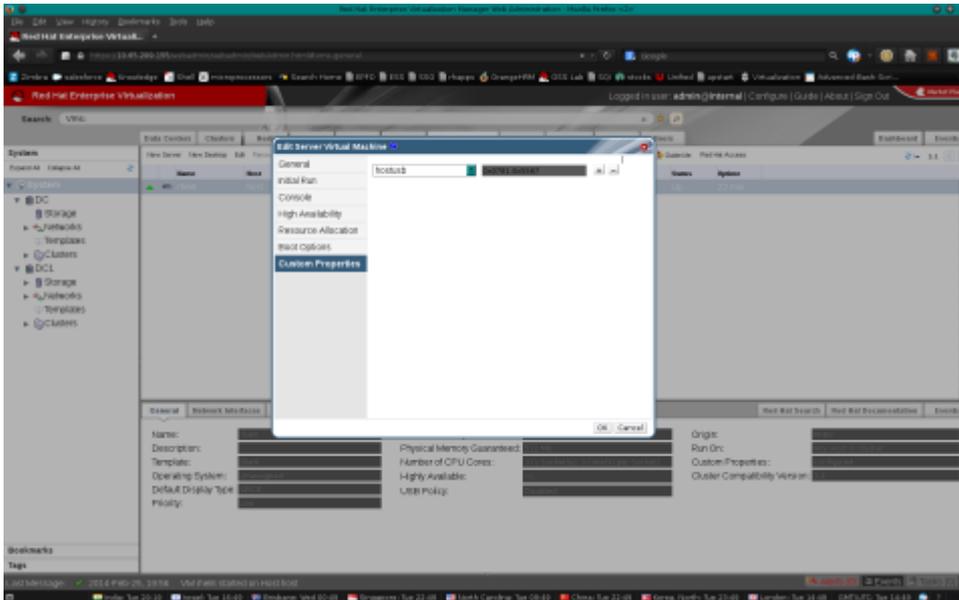vdsm   [VDSM_Hooks]   .

  - rhev   .

                    hostusb   .

```
# rhevm-config -s UserDefinedVMProperties='hostusb=[\w:&]+' --cver=3.2
# /etc/init.d/ovirt-engine restart
```

                                            .

```
# rhevm-config -g UserDefinedVMProperties
```

            rhevm GUI   '   '   Custom Properties
                              .

  1. hostusb
  2.   0x0781 : 0x5567   .

:   ID(0781),   ID(5567)   `lsusb`   .

    VM   USB   .

:   `rhevm-config -s`   `hostusb`
        .   `hostusb`   .

```
# rhevm-config -g UserDefinedVMProperties
UserDefinedVMProperties: copypaste=^(yes|no)$ version: 3.2

# rhevm-config -s
UserDefinedVMProperties='copypaste=^(yes|no)$;hostusb=[\w:&]+' --cver=3.2

# rhevm-config -g UserDefinedVMProperties
UserDefinedVMProperties: copypaste=^(yes|no)$;hostusb=[\w:&]+ version: 3.2
```

- https://access.redhat.com/solutions/146793
- https://access.redhat.com/solutions/474373
- https://access.redhat.com/solutions/734323

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/host_usb_passthrough_to_rhev_guests

Last update: **2020/02/20 06:14**