[OSSA-2020-006] Soft reboot after live-migration reverts instance to original source domain XML (CVE-2020-17376)

Bug #1890501 reported by Lee Yarwood on 2020-08-05

This bug affects 1 person

268

Affects	Status	Importance	Assigned to	Milestone
OpenStack Compute (nova)	Fix Released	Critical	Lee Yarwood	
Pike	Fix Released	Critical	Lee Yarwood	
Queens	Fix Released	Critical	Lee Yarwood	
Rocky	Fix Released	Critical	Lee Yarwood	
Stein	Fix Released	Critical	Lee Yarwood	
Train	Fix Released	Critical	Lee Yarwood	
Ussuri	Fix Released	Critical	Lee Yarwood	
Victoria	Fix Released	Critical	Lee Yarwood	
OpenStack Security Advisory	Fix Released	High	Jeremy Stanley	

Bug Description

```
Description
===========

When live migrating instances with attached volumes Nova will first ensure
that the volumes are connected on the destination before updating the
underlying domain XML to be used on the destination to correctly map to
these volumes.

At present in the case of volumes connected over iSCSI or FC this ensures
that the instance points to the correct host block devices as these may
differ from the source.

However if a user requests a soft reboot of an instance after a successful
live migration the underlying libvirt domain will rollback to the XML
definition used on the source. In the case of volumes provided over iSCSI
or FC etc this can potentially lead to the wrong
 volume being attached to the instance on the destination leading to
possible data exfiltration or corruption.

It appears that this is due to Nova not providing VIR_MIGRATE_
PARAM_PERSIST_XML during the migration resulting in the original source
domains persistent configuration being used instead:

/**
      * VIR_MIGRATE_PARAM_DEST_XML:
      *
      * virDomainMigrate* params field: the new configuration to be used
for the
      * domain on the destination host as VIR_TYPED_PARAM_STRING. The
configuration
      * must include an identical set of virtual devices, to ensure a
stable guest
      * ABI across migration. Only parameters related to host side
configuration
      * can be changed in the XML. Hypervisors which support this field
will forbid
      * migration if the provided XML would cause a change in the guest
ABI. This
      * field cannot be used to rename the domain during migration (use
      * VIR_MIGRATE_PARAM_DEST_NAME field for that purpose). Domain name
in the
      * destination XML must match the original domain name.
      *
      * Omitting this parameter keeps the original domain configuration.
Using this
      * field with hypervisors that do not support changing domain
configuration
      * during migration will result in a failure.
      */
     # define VIR_MIGRATE_PARAM_DEST_XML "destination_xml"

     /**
      * VIR_MIGRATE_PARAM_PERSIST_XML:
      *
      * virDomainMigrate* params field: the new persistent configuration
to be used
      * for the domain on the destination host as VIR_TYPED_PARAM_STRING.
      * This field cannot be used to rename the domain during migration
(use
      * VIR_MIGRATE_PARAM_DEST_NAME field for that purpose). Domain name
in the
      * destination XML must match the original domain name.
      *
      * Omitting this parameter keeps the original domain persistent
configuration.
      * Using this field with hypervisors that do not support changing
domain
      * configuration during migration will result in a failure.
      */
     # define VIR_MIGRATE_PARAM_PERSIST_XML "persistent_xml"

Steps to reproduce
==================

   0) Deploy overcloud with multipath and iscsi/LVM cinder backend.
   1) Delete all instances and check no device path remained on both host1
and host2.
   2) Boot instances, VM1 on host1 and VM2 on host2.
      $ cinder create --name cirros1 --volume-type lvm --image cirros 1
```

Report a bug

This report contains **Public Security** information

Everyone can see this security related information.

You are not directly subscribed to this bug's notifications.

Edit bug mail

Other bug subscribers

Subscribe someone else

Notified of all changes

Lee Yarwood
Mark DeVerter
Mohammed Naser
Nova Core securit...
Robert Frohl
Stephen Finucane
melanie witt

May be notified

-
ANish
Ahmed
Ahmed Ezzat
Aishwarya
Alex Baretto
Alex Ermolov
Alex Meade
Alex Xu
Alfred Shen
Alfredzo Nash
Ali hussnain
Amir Sadoughi
Andrea Frittoli
Andrea Rosa
Andy Southgate
Anna
Anthony Young
Antony Francis Ma...
April Wang
Arpita Rathi
Aruna Kushwaha
Asghar Riahi
Ashish Kumar Singh
Augustina Ragwitz
Aynur
Barki Mustapha
Bartlomiej Plotka
Belmiro Moreira
Bill Dymek
Branko Vukmirovic
Branko Vukmirovic
Brian Wang
Brin Zhang
Bruce Basil Mathews
Bruce Martins
C Sasi Kanth
Calub Viem
Cara O'Brien
Chason Chan
Chinmay Naik
Chris Samson
Christian Berendt
Christoph Fiehe
Craig Miller
David
David Lapsley
David M. Zendzian
David Pravec
David Seelbach
Deepak Nair
DengBO
Derek Ragona
Devdeep Singh
Donghoon Kim
Dongwon Cho
Douglas Mendizábal
Dustin Lundquist
Ed Villalovoz

```
      $ cinder create --name cirros2 --volume-type lvm --image cirros 1
      $ nova boot --block-device-mapping vda=$cirrosvol1 ... --host
host1.localdomain testvm1
      $ nova boot --block-device-mapping vda=$cirrosvol2 ... --host
host2.localdomain testvm2
      $ openstack server add floating ip testvm1 xx.xx.xx.xx
      $ openstack server add floating ip testvm2 yy.yy.yy.yy
   3) Soft reboot each instances and check no problem has occured.
      $ nova reboot testvm1
      $ nova reboot testvm2
   4) Execute live-migration VM1 to host2, check VMs for the device path
setting in
      each XML.
      $ nova live-migration testvm1 host2.localdomain
   5) Execute soft reboot VM1, check VMs for the device path setting in
each XML.
      $ nova reboot testvm1
   6) Login to each VMs and check syslogs.

Expected result
===============

After live-migration and soft reboot instance, device paths indicated by
virsh dumpxml --inactive and qemu XML file are changed to new value fit to
destination host.

Actual result
=============

After live-migration and soft reboot instance, device paths indicated by
virsh dumpxml --inactive and qemu XML file are the value of source host
before migration.

Environment
===========
1. Exact version of OpenStack you are running. See the following
  list for all releases: http://docs.openstack.org/releases/

   Reported downstream against stable/train and libvirt 5.6.0-10.

2. Which hypervisor did you use?
   (For example: Libvirt + KVM, Libvirt + XEN, Hyper-V, PowerKVM, ...)
   What's the version of that?

   libvirt + KVM

2. Which storage type did you use?
   (For example: Ceph, LVM, GPFS, ...)
   What's the version of that?

   LVM/iSCSI with multipath enabled but any host block based storage
backend will do.

3. Which networking type did you use?
   (For example: nova-network, Neutron with OpenVSwitch, ...)

   N/A

Logs & Configs
==============

The following test env logs are copied verbatim from a private downstream
security bug:

https://bugzilla.redhat.com/show_bug.cgi?id=1862353

   * Device paths initial state
                                     host1 host2
     =================================================================
==================================
     VM1 multipath -ll 360014053825c172898b4ba4a5353515c dm-0 ---
         virsh dumpxml <source dev='/dev/dm-0' index='1'/> ---
         virsh dumpxml --inactive <source dev='/dev/dm-0'/> ---
         qemu xml file <source dev='/dev/dm-0'/> ---
     ---------------------------------------------------------------------
------------------------------
     VM2 multipath -ll --- 36001405fc681536d0124af2a9fd99c10 dm-0
         virsh dumpxml --- <source dev='/dev/dm-0' index='1'/>
         virsh dumpxml --inactive --- <source dev='/dev/dm-0'/>
         qemu xml file --- <source dev='/dev/dm-0'/>

   * Device paths after VM1 live-migration to host2
                                     host1 host2
     =================================================================
==================================
     VM1 multipath -ll --- 360014053825c172898b4ba4a5353515c dm-2
         virsh dumpxml --- <source dev='/dev/dm-2' index='1'/>
         virsh dumpxml --inactive --- <source dev='/dev/dm-0'/> <== not
dm-2
         qemu xml file --- <source dev='/dev/dm-0'/> <== not dm-2
     ---------------------------------------------------------------------
------------------------------
     VM2 multipath -ll --- 36001405fc681536d0124af2a9fd99c10 dm-0
         virsh dumpxml --- <source dev='/dev/dm-0' index='1'/>
         virsh dumpxml --inactive --- <source dev='/dev/dm-0'/>
         qemu xml file --- <source dev='/dev/dm-0'/>

   * Device paths after soft reboot VM1 on host2
                                     host1 host2
     =================================================================
==================================
     VM1 multipath -ll --- 360014053825c172898b4ba4a5353515c dm-2
         virsh dumpxml --- <source dev='/dev/dm-0' index='1'/> <== changed
to dm-0
         virsh dumpxml --inactive --- <source dev='/dev/dm-0'/>
         qemu xml file --- <source dev='/dev/dm-0'/>
     ---------------------------------------------------------------------
------------------------------
     VM2 multipath -ll --- 36001405fc681536d0124af2a9fd99c10 dm-0
         virsh dumpxml --- <source dev='/dev/dm-0' index='1'/>
         virsh dumpxml --inactive --- <source dev='/dev/dm-0'/>
         qemu xml file --- <source dev='/dev/dm-0'/>
```

- Eric Kwon
- Eric Xie
- Fontenay Tony
- Gage Hugo
- Gaurav Singh
- Gavin B
- Greg Althaus
- Guangya Liu (Jay ...
- Haobo Liu
- Haochen Zhang
- Harikrishna S
- Hohyun, Jeon
- Hohyun, Jeon
- Honorarac.Org
- Hosam Al Ali
- Hugo Kou
- Hui Cheng
- Ian Y. Choi
- Ilya Alekseyev
- Ivan Groenewald
- Jamal Mitchell
- Jared R Greene
- Jay Janardhan
- Jeff Ward
- Jeremy Stanley
- Jia Dong
- Jie Li
- Jiyong Zhang
- Joel wineland
- John
- John Herndon
- John Lenihan
- John Masciantoni
- Jordan Rinke
- Joshua Padman
- Juergen Leopold
- Jung hyunjin
- Kausal Malladi
- Kausum Kumar
- Kei Masumoto
- Ken'ichi Ohmichi
- Kenji Motohashi
- Kent Liu
- Kunal.Yadav
- LIU Yulong
- Lawrnecy Meng
- Le Tian Ren
- Lei Zhang
- Lewis Denny
- Li Xipeng
- Lisathomes
- Louis Fourie
- Lukas Koenen
- Madhu CR
- Mamta Jha
- Manikantha Sriniv...
- Manoj Raju
- Marcus Santos
- Marcus Vinicius G...
- Mario Carvalho
- Mark McLoughlin
- Marta Sdvoijspa
- Matthew Thode
- Meera Belur
- Michael Rowland H...
- Mika Kohonen
- Mike Evenosky
- Milind Barve
- Mohankumar
- Nanda Kishore
- Naved Ali
- Naved Ali Shah
- Nayna Patel
- Normen Scholtke
- OpenStack Vulnera...
- Pankaj Mishra
- Paul Voccio
- Pavani_addanki
- Perry Waldner
- Piet Delaney
- Piyanai Saowaratt...
- Pradeep Roy Kandru
- Pranali Deore
- Prateek
- Prithiv Mohan
- Prosunjit Biswas
- Rafi Khardalian
- Rajesh Battala
- Raju Alluri
- Ranjit Ray

```
    * VM1 syslog file before live-migration
          $ cat /var/log/messages
          ...
          Jul 28 05:28:38 cirrostestvm1 kern.info kernel: [ 0.780031] usb
1-1: new full-speed USB device number 2 using uhci_hcd
          Jul 28 05:28:39 cirrostestvm1 kern.info kernel: [ 1.272305]
Refined TSC clocksource calibration: 2099.976 MHz.
          Jul 28 05:28:40 cirrostestvm1 authpriv.info dropbear[260]:
Running in background
          Jul 28 05:28:40 cirrostestvm1 daemon.info init: reloading
/etc/inittab
          Jul 28 05:28:40 cirrostestvm1 daemon.info init: starting pid 1,
tty '/dev/ttyS0': '/sbin/getty -L 115200 ttyS0 vt100 '
          Jul 28 05:28:40 cirrostestvm1 daemon.info init: starting pid 1,
tty '/dev/tty1': '/sbin/getty 115200 tty1'
          Jul 28 05:28:48 cirrostestvm1 kern.debug kernel: [ 10.992106]
eth0: no IPv6 routers present
          Jul 28 05:29:45 cirrostestvm1 authpriv.info dropbear[301]: Child
connection from **.**.**.**:33648
          Jul 28 05:29:48 cirrostestvm1 authpriv.notice dropbear[301]:
Password auth succeeded for 'cirros' from **.**.**.**:33648
          $

    * VM1 syslog file after soft reboot on host2
        hostname command return correct value, but VM1 syslog is recorded
by VM2.
        (in some cases, VM1 and VM2 syslog files are destroyed and cannot
be read as text file)
          $ hostname
          cirrostestvm1
          $ cat /var/log/messages | tail
          Jul 28 06:03:01 cirrostestvm2 authpriv.info dropbear[325]: Child
connection from 172.31.151.1:35894
          Jul 28 06:03:05 cirrostestvm2 authpriv.notice dropbear[325]:
Password auth succeeded for 'cirros' from **.**.**.**:35894
          Jul 28 06:03:05 cirrostestvm2 authpriv.info dropbear[325]: Exit
(cirros): Disconnect received
          Jul 28 06:03:30 cirrostestvm2 authpriv.info dropbear[328]: Child
connection from **.**.**.**:36352
          Jul 28 06:03:34 cirrostestvm2 authpriv.notice dropbear[328]:
Password auth succeeded for 'cirros' from **.**.**.**:36352
          Jul 28 06:03:34 cirrostestvm2 authpriv.info dropbear[328]: Exit
(cirros): Disconnect received
          Jul 28 06:03:39 cirrostestvm2 authpriv.info dropbear[331]: Child
connection from **.**.**.**:36484
          Jul 28 06:03:41 cirrostestvm2 authpriv.info dropbear[331]: Exit
before auth (user 'cirros', 0 fails): Exited normally
          Jul 28 06:03:45 cirrostestvm2 authpriv.info dropbear[332]: Child
connection from **.**.**.**:36588
          Jul 28 06:03:49 cirrostestvm2 authpriv.notice dropbear[332]:
Password auth succeeded for 'cirros' from **.**.**.**:36588
```

See original description

Tags: libvirt live-migration security in-stable-ussuri

CVE References

2020-17376

Jeremy Stanley (fungi) wrote on 2020-08-05: #1

Since this report concerns a possible security risk, an incomplete security advisory task has been added while the core security reviewers for the affected project or projects confirm the bug and discuss the scope of any vulnerability along with potential solutions.

description:updated
description:updated
Changed in ossa:
    status:New → Incomplete

Jeremy Stanley (fungi) wrote on 2020-08-05: #2

It's not immediately clear to me from the bug description what security risk this poses. I would appreciate it if someone could clarify that, ideally with an example exploit scenario for how an attacker might leverage the defect to gain unintended access/control.

Lee Yarwood (lyarwood) wrote on 2020-08-05: #3

Apologies, I've added the following to the initial description, let me know if you would like any more details.

"""
When live migrating instances with attached volumes Nova will first ensure that the volumes are connected on the destination before updating the underlying domain XML to be used on the destination to correctly map to these volumes.

At present in the case of volumes connected over iSCSI or FC this ensures that the instance points to the correct host block devices as these may differ from the source.

However if a user requests a soft reboot of an instance after a successful live migration the underlying libvirt domain will rollback to the XML definition used on the source. In the case of volumes provided over iSCSI or FC etc this can potentially lead to the wrong
 volume being attached to the instance on the destination leading to possible data exfiltration or corruption.
"""

description:updated
description:updated

Ratnaker
RaviM Singh
Ray Trejo
Richa
Rick Melick
Robert Carr
Rochelle Grober
Rohini Diwakar
Ron Cannella
Rongze Zhu
Ryo Shi
Salvatore Orlando
Sanjay Tripathi
Sateesh
Satya Sanjibani R...
Satyanarayana Pat...
Scott Sanchez
Sebastian Luna-Va...
Shawn Hartsock
Shruthi Chari
Sid Sun
Simon
Songhee Kang
Soo Choi
Spencer Yu
Sridhar Gaddam
Steve Sloka
Steven Pavlon
Steven Relf
Stuart Hart
Summer Long
Surya N
Surya Seetharaman
Sushma Korati
Swami Reddy
Swaroop Jayanthi
Takashi Kajinami
Tao Zhou
Taurus Cheung
Tayaa Med Amine
Thongth
Tiago Everton Fer...
Tushar Patil
Uma
Venkata Siva Vija...
Vidhisha Nair
Vilobh Meshram
Vinu Pillai
Vish Ishaya
Vladik Romanovsky
Wayne A. Walls
William Wolf
WuBing
Xiang Hui
Xin Zhong
Yahoo! Engineerin...
Yalu Bai
Yongqiang Yang
You, Ji
Yusuf Güngör
Zahid Hasan
ZhangNi
Ziv
aginwala
ammarun
armyman420
avinashsau
brightson
bugtracker@devshe...
chaiwat wannaposop
chitu
congge
david
dk647
droom
fei Yang
fhbeak60161
galeido
gsccc
hougangliu
iopenstack
james kang
jay.xu
jeff wang
kalim khuang
kgrvamsi
lanpi
laoyi
liaonanhai
lica

Jeremy Stanley (fungi) wrote on 2020-08-05: #4

Okay, so *if* someone has access to an instance which happens to have been live-migrated within a vulnerable deployment then they can gain read+write access to some random allocation on the array which might contain another tenant's data?

I'm open to input, but the risk here doesn't seem severe enough to warrant keeping this report secret until a fix is developed and reviewed. We'll still likely want to publish an advisory for this once a fix is available, however.

Lee Yarwood (lyarwood) wrote on 2020-08-05: #5

Correct, any user with access to an instance that has been live migrated (an admin only op) can soft reboot the instance and may end up with RW access to a volume owned by another user.

I'm not entirely convinced that we want to open this up so quickly as this could easily provide a bad actor with access to the root disk of another instance, access to keys and other sensitive data etc. Making such a trivial exploit public before the fix is in the gate doesn't seem right.

Jeremy Stanley (fungi) wrote on 2020-08-06: #6

I'm probably missing some nuance, but this doesn't sound like it would be especially hard for a user to stumble across accidentally anyway (and then get very confused). It also doesn't seem like even a determined attacker could take advantage of this for a particularly focused attack due to the need for an admin to live migrate the instance first, the random nature of the resources they might get access to, and the fact that it can only be exploited once per instance (so somewhat expensive exercise to repeat at a massive enough scale for effective dragnetting).

What versions of Nova does this affect, does anyone know off hand?

Lee Yarwood (lyarwood) wrote on 2020-08-06: #7

VIR_MIGRATE_PARAM_PERSIST_XML was introduced into libvirt by b028e9d7c2f0f7713ba102d01aece13ee72136a1 and first included in the v1.3.4 release that came out in May of 2016.

In terms of which versions of Nova are impacted we've never provided VIR_MIGRATE_PARAM_PERSIST_XML so *any* version running alongside libvirt >= v1.3.4 would be as we don't cap the max supported version of libvirt for each release.

Dan Smith (danms) wrote on 2020-08-06: #8

If this is really is happening, then I agree, it seems like this would have been noticed and reported a LOT since 2016. That makes it hard for me to believe it's really happening. In a lot of cases we'll use something like /dev/disk/by-id/$id in the XML which makes it hard for a collision to expose another user's data, but would rather just break the reboot or something. However, in discussing privately with Lee it sounds like os-brick will tell us to use generic devices like /dev/dm-X for cases like multipath, which would definitely make it a lot easier to get access to a device we shouldn't have.

I think Jeremy's point is that this isn't exploitable directly by a user because live-migration is admin-only, and thus the attack route would be to spin up an instance with a volume, wait a year until a maintenance window has passed, and then try soft-reboot to see if you got anything. Not knowing the backend (i.e. if they're using multipath or ceph or whatever) makes the intermediate expense pretty high for a very rare payoff.

Jeremy, are you prescribing some other handling for this because of the difficulty of exploitation? If so, what is that? I'd also point out that live-migration is admin only by default, but could be exposed to users (although you'd be pretty crazy to do so in a public cloud, it's not uncommon in private clouds). Further, if we did hit this, exposing someone else's data volume to the wrong use is pretty much the worst sin we can commit.

Lee, normally we attach a patch here for review first. I'm guessing this is as simple to fix as just setting that flag on the migrate call, right? Do we need to care about local/remote libvirt version mismatches? If not and the patch is a one-liner, I say we just handle this with care out of caution.

John Garbutt (johngarbutt) wrote on 2020-08-06: #9

We only recently have the min version of libvirt high enough for us to use > v1.3.4. So I guess its pluasable.

+1 Dan's comment on live-migration permissions, many users have access to it, although that is not the default.

+1 on Dan's comment around the data leak being one of the worst possible failure modes here.

I guess the patch is tricker for when min_libvirt is < v1.3.4.

Does this not also affect pinned CPU cores as well? Because we might pick a different set of CPUs on the desitnation hypervisor (train onwards)? With all the speclative execution stuff, that is also a possible data leak. Certainly leads to performance oddness.

Do we have an understanding of what backends use this operation mode? I remember discussing this with Cinder around multi-attach time frame, and it sounded like very few backends (if any upstream?) actually use these host based connections.

Jeremy Stanley (fungi) wrote on 2020-08-06: #10

Dan: my main point on difficulty to exploit was that it supports handling this report in public, since discussing and reviewing fixes in the open is much easier and less error-prone for everyone. Designing fixes in secret

lin han
liuwei
liuzhuangzhuang
lololmarwa255
lpmqtt
maestropandy
manish
mershard frierson
mimul
miralaunchpad
mohit.048
nawawit kes
phalgun sirga
raja
robin
sangbaobao
satyanarayana pat...
satyanarayana pat...
shadyabhi
sivagnanam C
soumiyajit
sunilcn
tangfeixiong
victorye81
vishwa
vivek.ys
vks1
wanghuagong
wangqiang.sheng
woody89
xianliangchi
xiaoningli
xreuze
ya.wang
yangbo
yangkai
yangzhenyu
yanxubin
yilong
yongxiangwang
yysimida
zhengyue
zhu zhu
zhuangkai.zong

Patches

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-master.patch

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-ussuri.patch

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-train.patch

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-stein.patch

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-rocky.patch

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-queens.patch

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-pike.patch

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-newton.patch

Add patch

under embargo should be reserved for only the most risky of defects. I just want to be sure that when we choose to handle fixes in private we're conscious of the cost compared to following our normal community development processes.

Lee Yarwood (lyarwood) wrote on 2020-08-06: #11

Yeah MIN_LIBVIRT_VERSION only went above v1.3.4 in Stein but we've definitely been using it downstream for much much longer than that as again there's no max version constraint in the code. It should be trivial to workaround however with a simple available version check.

Yes AFAICT this will also impact pinned CPUs, NUMA etc basically anything that we update below that differs between the source and destination hosts:

https://github.com/openstack/nova/blob/9ecefeb836964c52a5a2969b15c82b11c51d32ab/nova/virt/libvirt/migration.py#L56-L70

Lee Yarwood (lyarwood) wrote on 2020-08-06: #12

Download full text (7.4 KiB)

Anyway I've reproduced against devstack now and verified that the attached patch works for me:

I've created two bfv instances using the LVM/iSCSI c-vol backend that os-brick is presenting as raw /dev/sd* devices to Nova. b8acff7f-7430-40f8-b67f-5f51dcf07299 running on controller and 45302dcc-906f-4d47-b774-45165a867fca running on subnode.

```
stack@controller $ sudo virsh domblklist b8acff7f-7430-40f8-b67f-
5f51dcf07299
 Target Source
--------------------
 vda /dev/sdb

stack@subnode $ sudo virsh domblklist 45302dcc-906f-4d47-b774-45165a867fca
 Target Source
--------------------
 vda /dev/sdb

stack@controller $ openstack server migrate --os-compute-api-version 2.30
--live-migration \
--host controller.example.com 45302dcc-906f-4d47-b774-45165a867fca

stack@controller $ sudo virsh domblklist 45302dcc-906f-4d47-b774-
45165a867fca
 Target Source
--------------------
 vda /dev/sdc

stack@controller $ sudo virsh dumpxml 45302dcc-906f-4d47-b774-45165a867fca
> original.xml
stack@controller $ openstack server reboot --soft 45302dcc-906f-4d47-b774-
45165a867fca
stack@controller $ sudo virsh dumpxml 45302dcc-906f-4d47-b774-45165a867fca
> soft.xml
stack@controller $ sudo virsh domblklist 45302dcc-906f-4d47-b774-
45165a867fca
 Target Source
--------------------
 vda /dev/sdb

stack@controller $ diff -u original.xml soft.xml

$ diff -u original.xml soft.xml
--- original.xml 2020-08-06 11:30:36.611368640 -0400
+++ soft.xml 2020-08-06 11:30:57.531787186 -0400
@@ -1,23 +1,23 @@
-<domain type='kvm' id='6'>
+<domain type='kvm' id='7'>
   <name>instance-00000004</name>
   <uuid>45302dcc-906f-4d47-b774-45165a867fca</uuid>
   <metadata>
     <nova:instance xmlns:nova="http://openstack.org/xmlns/libvirt/nova/1.
0">
- <nova:package version="21.1.0"/>
- <nova:name>test</nova:name>
- <nova:creationTime>2020-08-06 15:29:32</nova:creationTime>
- <nova:flavor name="m1.tiny">
- <nova:memory>512</nova:memory>
- <nova:disk>1</nova:disk>
- <nova:swap>0</nova:swap>
- <nova:ephemeral>0</nova:ephemeral>
- <nova:vcpus>1</nova:vcpus>
- </nova:flavor>
- <nova:owner>
- <nova:user uuid="c7bfad6fb6cc45778d2eb63642eb10d5">admin</nova:user>
- <nova:project uuid="6b4564ddd49242ecad343e41e6bf134f">admin<
/nova:project>
- </nova:owner>
- </nova:instance>
+ <nova:package version="21.1.0"/>
+ <nova:name>test</nova:name>
+ <nova:creationTime>2020-08-06 15:29:32</nova:creationTime>
+ <nova:flavor name="m1.tiny">
+ <nova:memory>512</nova:memory>
+ <nova:disk>1</nova:disk>
+ <nova:swap>0</nova:swap>
+ <nova:ephemeral>0</nova:ephemeral>
+ <nova:vcpus>1</nova:vcpus>
+ </nova:flavor>
+ <nova:owner>
+ <nova:user uuid="c7bfad6fb6cc45778d2eb63642eb10d5">admin</nova:user>
+ <nova:project uuid="6b4564ddd49242ecad343e41e6bf134f">admin<
/nova:project>
+ </nova:owner>
+ </nova:instance>
   </metadata>
   <memory unit='KiB'>524288</memory>
   <currentMemory unit='KiB'>524288</currentMemory>
@@ -59,7 +59,7 @@
```

```
    <emulator>/usr/bin/qemu-system-x86_64</emulator>
    <disk type='bloc...
```

Read more...

1 comments hidden | view all 101 comments

Lee Yarwood (lyarwood) wrote on 2020-08-06: #14

FWIW I've asked the current Nova PTL gibi to review this in the morning and confirm if he thinks it's okay for us to open this up.

Dan Smith (danms) wrote on 2020-08-06: #15

Lee, apologies if I missed it, but is this something we can just do on one side of an upgrade where the libvirt version is different? Meaning, we're just making a call to *our* libvirt, does it handle the case where the old libvirt doesn't need/handle the new param?

If there's no upgrade concern here, reviewing that patch seems pretty trivial, especially if Lee has tested it.

Balazs Gibizer (balazs-gibizer) wrote on 2020-08-06: #16

After chatting it through with Lee I'm OK with the attached patch. Regarding the publicity of the issue, I'm on Dan side that it is better to keep this private just to be on the safe side due to the size of the impact.

I' will be on PTO in the next two weeks but I have full trust in the already involved nova folks to handle this properly.

Jeremy Stanley (fungi) on 2020-08-06

Changed in ossa:
status: Incomplete → Confirmed

Jeremy Stanley (fungi) wrote on 2020-08-06: #17

I'm still a little fuzzy on the details so please suggest corrections/improvements, but this is an initial draft of the impact description we'd use to request a CVE assignment, and which will eventually form the basis for any public advisory...

Title: Live migration fails to update source domain XML
Reporter: Lee Yarwood (Red Hat)
Products: Nova
Affects: <19.3.1, >=20.0.0 <20.3.1, ==21.0.0

Description:
Lee Yarwood (Red Hat) reported a vulnerability in Nova live migration. By performing a soft reboot of an instance which has previously undergone live migration, a user may gain access to the virtual machine's original block devices resulting in possible access to data for another tenant to whom those devices have since been reallocated. Only deployments allowing host-based connections for instance root and ephemeral devices are affected.

Lee Yarwood (lyarwood) wrote on 2020-08-06: #18

Dan, yeah that's a good point.

The attached patch only works when MIN_LIBVIRT_VERSION is >= v1.3.4. So that's for master, stable/ussuri, stable/train and stable/stein. I can post these backport patches tomorrow in the bug.

Prior to that on stable/rocky and stable/queens we will need to ensure the local libvirt version is >= v1.3.4 before adding the VIR_MIGRATE_PARAM_PERSIST_XML param.

If it isn't then the < v1.3.4 version of libvirt should retain its original behaviour of persisting VIR_MIGRATE_PARAM_DEST_XML when the VIR_MIGRATE_PERSIST_DEST flag is provided. Even when talking to an upgraded >= v1.3.4 libvirt on the dest host.

Lee Yarwood (lyarwood) wrote on 2020-08-06: #19

> Title: Live migration fails to update source domain XML

Live migration fails to update the persistent domain XML on the destination host

> Reporter: Lee Yarwood (Red Hat)

This was initially reported downstream by Tadayoshi Hosoya <email address hidden>, I'm not sure if we can credit him somehow here?

> Affects: <19.3.1, >=20.0.0 <20.3.1, ==21.0.0

I assume this just means all supported releases?

> a user may gain access to the virtual machine's original
> block devices resulting in possible access to data for
> another tenant to whom those devices have since been
> reallocated. Only deployments allowing host-based
> connections for instance root and ephemeral devices are
> affected.

a user may gain access to destination host devices that share the same paths as host devices previously referenced by the virtual machine on the source. This can include block devices that map to different Cinder volumes on the destination to the source.

Jeremy Stanley (fungi) wrote on 2020-08-06: #20

Since stable/rocky is already under extended maintenance there won't be any new point releases and any security fixes we do feel like backporting are provided on a best-effort basis as a convenience anyway, so I'd mostly worry about stable/stein and later as those are our officially supported stable branches right now. We can always add backports for extended maintenance branches after a public advisory.

Lee: Thanks for the impact description edits. I'd like to have a shorter title if possible, since this makes it into E-mail subject lines and the like. Would just "Live migration fails to update persistent domain XML" work? The idea is mainly to be able to distinguish it from any other similar (past or future) Nova vulnerabilities. As for the original reporter would "Tadayoshi Hosoya (NEC)" be accurate? I can credit you both, no problem. And yes, the affects line is all currently supported releases, excluding the next possible releases (consider this from the point of view of someone looking at the advisory or CVE a year from now and trying to work out whether they're patched sufficiently to solve the problem). As for the prose, I'll update it with your text. Here's my next take...

Title: Live migration fails to update persistent domain XML
Reporter: Tadayoshi Hosoya (NEC) and Lee Yarwood (Red Hat)
Products: Nova
Affects: <19.3.1, >=20.0.0 <20.3.1, ==21.0.0

Description:
Tadayoshi Hosoya (NEC) and Lee Yarwood (Red Hat) reported a vulnerability in Nova live migration. By performing a soft reboot of an instance which has previously undergone live migration, a user may gain access to destination host devices that share the same paths as host devices previously referenced by the virtual machine on the source. This can include block devices that map to different Cinder volumes on the destination to the source. Only deployments allowing host-based connections for instance root and ephemeral devices are affected.

Lee Yarwood (lyarwood) wrote on 2020-08-06: #21

ACK thanks Jeremy the text LGTM now.

FWIW with my downstream hat on I will be fixing this back to stable/queens anyway so I'll do my best to have things posted at the time of disclosure upstream as well.

Jeremy Stanley (fungi) on 2020-08-06

Changed in ossa:
status:Confirmed → Triaged

Jeremy Stanley (fungi) wrote on 2020-08-06: #22

A request for CVE assignment has been submitted to MITRE based on the proposed impact description from comment #20, but please feel free to continue suggesting edits if needed.

Lee Yarwood (lyarwood) wrote on 2020-08-07: #23

Dumping some additional context in here after talking to danpb (libvirt/QEMU) about the underlying libvirt migrateToURI3 behaviour. It looks like v1.2.20 initial introduced the libvirt behaviour of copying the source persistent domain definition across to the destination in order to ensure something is persisted when VIR_MIGRATE_PARAM_DEST_XML wasn't provided but the VIR_MIGRATE_PERSIST_DEST flag was. Later v1.3.4 then introduced VIR_MIGRATE_PARAM_PERSIST_XML to overwrite the persistent domain on the destionation.

We also found that the reason Nova is rolling back to the persistent domain during a soft reboot is due to our use of virDomainShutdown [1] and virDomainCreate [2] within _soft_reboot [3]. virDomainReboot [4] wouldn't actually cause this as libvirt doesn't allow QEMU to exit. It would also drop our requirement for transient domains entirely from Nova so is definitely something we should look into as a follow up to this.

[1] https://libvirt.org/html/libvirt-libvirt-domain.html#virDomainShutdown
[2] https://libvirt.org/html/libvirt-libvirt-domain.html#virDomainLaunch
[3] https://github.com/openstack/nova/blob/9ecefeb836964c52a5a2969b15c82b11c51d32ab/nova/virt/libvirt/driver.py#L3157-L3203
[4] https://libvirt.org/html/libvirt-libvirt-domain.html#virDomainReboot

Jeremy Stanley (fungi) wrote on 2020-08-07: #24

MITRE has assigned CVE-2020-17376 for tracking this.

```
summary:Soft reboot after live-migration reverts instance to original source
        - domain XML
        + domain XML (CVE-2020-17376)
```

Lee Yarwood (lyarwood) wrote on 2020-08-07: **Re: Soft reboot after live-migration reverts instance to original source domain XML (CVE-2020-17376)** #25

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-master.patch (5.8 KiB, text/plain)

Lee Yarwood (lyarwood) wrote on 2020-08-07: #26

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-ussuri.patch (5.9 KiB, text/plain)

1 comments hidden view all 101 comments

Lee Yarwood (lyarwood) wrote on 2020-08-07: #28

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-train.patch (6.0 KiB, text/plain)

Lee Yarwood (lyarwood) wrote on 2020-08-07: #29

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-stein.patch (6.0 KiB, text/plain)

Lee Yarwood (lyarwood) wrote on 2020-08-07: #30

I've attached the revised patches for master, ussuri, train and stein. I've successfully ran unit, functional and pep8 against each locally, happy to attach the results if it would help.

The rocky and queens patches are slightly more involved due to MIN_LIBVIRT_VERSION lower than v1.3.4 but I should post them shortly for review as well.

2 comments hidden view all 101 comments

John Garbutt (johngarbutt) wrote on 2020-08-10: #33

For those last two, did you really want: "if persistent_xml_param and destination_xml:" to match the conditional we have on master?

I haven't heavily reviewed the test patches, but that looks good to me.

Lee Yarwood (lyarwood) wrote on 2020-08-10: #34

John, yeah great point, I'll respin and attach later today. Would it help if we added a few additional Nova cores to this bug to help with these reviews?

Jeremy, what is the timeline for public disclosure or is it too early to say?

Jeremy Stanley (fungi) wrote on 2020-08-10: #35

If we can get tentative pre-approval for your patches I'd like to supply them to our downstream stakeholders and the private linux-distros ML as early as tomorrow, with advisory publication to follow a week later (so Tuesday, August 18th ideally). Our policy is to have at least three but no more than five "business" days between advance notification to downstream consumers under embargo and final publication: https://security.openstack.org/vmt-process.html#embargoed-disclosure

Jeremy Stanley (fungi) wrote on 2020-08-10: #36

Also, yes as far as I'm concerned please directly subscribe any other reviewers who can help to confirm these patches expediently. I would prefer they be as finalized as possible (in the absence of public review and CI) before providing copies to anyone, lest we end up needing to send revised copies later in the embargo period and risk causing unwarranted confusion.

Dan Smith (danms) wrote on 2020-08-10: #37

In the past, we needed to make sure the patches solved the problem, even if they weren't cosmetically perfect before allowing them to go public, after which the regular review process could proceed. I think we've also said in the past we only *need* patches for master in order to do that.

Regardless, I'm fine with the master and recent patches, and I trust that Lee's revision for the farther-back ones will get resolved. Especially given what we *do* care about in terms of supported versions, I'm fine moving forward once Lee posts his revisions. I'd much rather get this disclosed and the patches into gerrit sooner than later, as reviewing them here (especially for test and cosmetic reasons) is harder for everyone.

Jeremy Stanley (fungi) wrote on 2020-08-10: #38

If we're doing coordinated disclosure under embargo, we're basically telling Linux distros and public cloud providers to prepare production packages/container images/whatever with these patches applied, with the expectation that these are at least very close to being what will merge to stable branches, so we want them to be as correct as we can reasonably make them while reviewing in private.

Lee Yarwood (lyarwood) wrote on 2020-08-10: #39

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-rocky.patch (14.5 KiB, text/plain)

Lee Yarwood (lyarwood) wrote on 2020-08-10: #40

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-queens.patch (10.1 KiB, text/plain)

21 comments hidden view all 101 comments

Jeremy Stanley (fungi) wrote on 2020-08-17: #62

It's also the case that we've relied (by extension of MITRE's expectations) on the title/subject of vulnerabilities as a means of quickly differentiating multiple defects which can have the same symptoms and expose similar risks. The title line is generally a very brief description of the problem to be fixed (often the same as the corresponding bug report's title), not a risk assessment or exploit scenario, nor a news headline.

Jeremy Stanley (fungi) wrote on 2020-08-17: #63

Discussions of the advisory title aside, is there consensus on Lee's patches attached to comments #25-29 and #39-40? If so, I'll send the downstream notification tomorrow and propose a disclosure date of Tuesday, August 25 (one week from tomorrow). Are there any objections to this plan?

Nick Tait (nickthetait) wrote on 2020-08-17: #64

That date seems fine to me.

John Garbutt (johngarbutt) wrote on 2020-08-18: #65

Those patches look good to my eyes. Thank you Lee.

The date sounds sensible, I am unsure on the usual timeframe, but that sounds like some warning combined with getting this information to our users as soon as we can.

I think that description looks OK. I do wonder if we want to say the VM reverts to using the libvirt XML it used on the source host after a soft reboot. I guess the patches make that very clear.

In terms of mitigations, could you ask users to hard reboot instances that have been live-migrated via the API/horizon. I think that would also reset the persistent libvirt XML? Is that correct, or is it worse than that? I think operators could look at the actions list for each instance to determine if it has been affected by a live-migration followed by a soft reboot, and target those instances for a hard reboot?

Maybe that is too much detail, especially for something we would need to test to be sure it helps?

Lee Yarwood (lyarwood) wrote on 2020-08-18: #66

> In terms of mitigations, could you ask users to hard reboot
> instances that have been live-migrated via the API/horizon.
> I think that would also reset the persistent libvirt XML?
> Is that correct, or is it worse than that? I think operators
> could look at the actions list for each instance to
> determine if it has been affected by a live-migration
> followed by a soft reboot, and target those instances for
> a hard reboot?

Yes hard reboots will correct any instances that have already live migrated but I don't think we can ask users to do this as they can't know by default if their instances have been migrated.

Having operators review the event list for each instance and hard reboot any that have recently live migrated however seems like something we should document.

I'd also like to document a mitigation where admins disable soft reboots through policy until their env is patched. Forcing users to hard reboot and thus correct the persistent configuration.

Jeremy Stanley (fungi) wrote on 2020-08-18: #67

I'd like to get the pre-OSSA sent to downstream stakeholders (including the private linux-distros ML) today if possible. I could however add a sentence to the end of the impact description like this, if folks think it will help:

This only impacts deployments where users are allowed to perform soft reboots of server instances; it is recommended to disable soft reboots in policy (only allowing hard reboots) until the fix can be applied.

Lee Yarwood (lyarwood) wrote on 2020-08-18: #68

ACK sounds good to me Jeremy.

John Garbutt (johngarbutt) wrote on 2020-08-18: #69

+1 that sounds good.

melanie witt (melwitt) wrote on 2020-08-18: #70

The patches also LGTM.

Jeremy Stanley (fungi) wrote on 2020-08-19: #71

Pre-OSSA downstream stakeholder notification with patches for all 6 branches has been sent, with a preliminary disclosure date and time of Tuesday, August 25 at 15:00 UTC.

Lee Yarwood (lyarwood) wrote on 2020-08-19: #72

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-pike.patch (9.4 KiB, text/plain)

I was asked privately if I could provide a stable/pike version of this patch, I've posted a simple cherry-pick of the stable/queens change without testing it as I did with stable/{queens,rocky}.

Nick Tait (nickthetait) wrote on 2020-08-20: #73

If an attacker (Alice) used this flaw to gain access to a user's (Bob) drive. However, Bob uses full disk encryption and Alice doesn't know the decryption key, then the problem is largely averted right?

Jeremy Stanley (fungi) wrote on 2020-08-20: #74

It was suggested that Alice may also be able to write to the device, at a minimum wiping or corrupting the same copy of the volume Bob's system relies on and causing data loss or a denial of service.

Lee Yarwood (lyarwood) wrote on 2020-08-21: #75

Yes correct Jeremy, additionally Alice may end up with access to other underlying host resources that Bob is using such as pass-through PCI devices (nics, GPUs etc).

Mohammed Naser (mnaser) wrote on 2020-08-21: #76

I feel like this might come up pretty often but I think we can mention that deployment with RBD is not affected unless it's deployed with rbd_volume_local_attach=True ?

Lee Yarwood (lyarwood) wrote on 2020-08-25: #77

Mohammed, we could call out RBD but we might end up down a rabbit hole if we start talking about specific storage backends.

Jeremy, what is the schedule for the public disclosure today? When can I post patches to gerrit?

Lee Yarwood (lyarwood) wrote on 2020-08-25: #78

Apologies I missed that you had highlighted 15:00 UTC today in c#71.

Balazs Gibizer (balazs-gibizer) wrote on 2020-08-25: #79

The attached patches looks good to me.

Jeremy Stanley (fungi) wrote on 2020-08-25: #80

Lee, yep thanks, a few minutes prior to 15:00 UTC I'll switch this bug to Public Security state and then you can start pushing changes to Gerrit which mention it (that way they hopefully get recorded in here automatically). Once they've all been pushed I'll finalize the advisory, since we include the change URLs in it.

Jeremy Stanley (fungi) wrote on 2020-08-25: #81

Lee: Please go ahead and push the changes to review for master, stable/ussuri, stable/train, stable/stein, stable/rocky, stable/queens, and stable/pike at your earliest convenience. Thanks!

description:updated
information type:Private Security → Public Security
Changed in ossa:
status:Triaged → In Progress

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Fix proposed to nova (master)** #82

Fix proposed to branch: master
Review: https://review.opendev.org/747969

Changed in nova:
assignee:nobody → Lee Yarwood (lyarwood)
status:New → In Progress

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Fix proposed to nova (stable/ussuri)** #83

Fix proposed to branch: stable/ussuri
Review: https://review.opendev.org/747972

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Fix proposed to nova (stable/train)** #84

Fix proposed to branch: stable/train
Review: https://review.opendev.org/747973

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Fix proposed to nova (stable/stein)** #85

Fix proposed to branch: stable/stein
Review: https://review.opendev.org/747974

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Fix proposed to nova (stable/rocky)** #86

Fix proposed to branch: stable/rocky
Review: https://review.opendev.org/747975

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Fix proposed to nova (stable/queens)** #87

Fix proposed to branch: stable/queens
Review: https://review.opendev.org/747976

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Fix proposed to nova (stable/pike)** #88

Fix proposed to branch: stable/pike
Review: https://review.opendev.org/747978

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Related fix proposed to ossa (master)** #89

Related fix proposed to branch: master
Review: https://review.opendev.org/747980

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Related fix merged to ossa (master)** #90

Reviewed: https://review.opendev.org/747980
Committed: https://git.openstack.org/cgit/openstack/ossa/commit/?id=2cdc6ae08730ba6693700664dd1a233bcffc1e96
Submitter: Zuul
Branch: master

```
commit 2cdc6ae08730ba6693700664dd1a233bcffc1e96
Author: Jeremy Stanley <email address hidden>
Date: Tue Aug 25 14:45:19 2020 +0000

    Add OSSA-2020-006 (CVE-2020-17376)

    Change-Id: I4bb95e74551dc02664074a006f462683967f50f3
    Related-Bug: #1890501
```

OpenStack Infra (hudson-openstack) wrote on 2020-08-25: **Fix merged to nova (master)** #91

```
Reviewed: https://review.opendev.org/747969
Committed: https://git.openstack.org/cgit/openstack/nova/commit/?
id=1bb8ee95d4c3ddc3f607ac57526b75af1b7fbcff
Submitter: Zuul
Branch: master

commit 1bb8ee95d4c3ddc3f607ac57526b75af1b7fbcff
Author: Lee Yarwood <email address hidden>
Date: Wed Aug 5 23:00:06 2020 +0100

    libvirt: Provide VIR_MIGRATE_PARAM_PERSIST_XML during live migration

    The VIR_MIGRATE_PARAM_PERSIST_XML parameter was introduced in libvirt
    v1.3.4 and is used to provide the new persistent configuration for the
    destination during a live migration:

    https://libvirt.org/html/libvirt-libvirt-domain.html#VIR_MIGRATE_
PARAM_PERSIST_XML

    Without this parameter the persistent configuration on the destination
    will be the same as the original persistent configuration on the
source
    when the VIR_MIGRATE_PERSIST_DEST flag is provided.

    As Nova does not currently provide the VIR_MIGRATE_PARAM_PERSIST_XML
    param but does provide the VIR_MIGRATE_PERSIST_DEST flag this means
that
    a soft reboot by Nova of the instance after a live migration can
revert
    the domain back to the original persistent configuration from the
    source.

    Note that this is only possible in Nova as a soft reboot actually
    results in the virDomainShutdown and virDomainLaunch libvirt APIs
being
    called that recreate the domain using the persistent configuration.
    virDomainReboot does not result in this but is not called at this
time.

    The impact of this on the instance after the soft reboot is pretty
    severe, host devices referenced in the original persistent
configuration
    on the source may not exist or could even be used by other users on
the
    destination. CPU and NUMA affinity could also differ drastically
between
    the two hosts resulting in the instance being unable to start etc.

    As MIN_LIBVIRT_VERSION is now > v1.3.4 this change simply includes the
    VIR_MIGRATE_PARAM_PERSIST_XML param using the same updated XML for the
    destination as is already provided to VIR_MIGRATE_PARAM_DEST_XML.

    Co-authored-by: Tadayoshi Hosoya <email address hidden>
    Closes-Bug: #1890501
    Change-Id: Ia3f1d8e83cbc574ce5cb440032e12bbcb1e10e98
```

Changed in nova:
status:In Progress → Fix Released

OpenStack Infra (hudson-openstack) wrote on 2020-08-26: **Fix merged to nova (stable/ussuri)** #92

```
Reviewed: https://review.opendev.org/747972
Committed: https://git.openstack.org/cgit/openstack/nova/commit/?
id=bbf9d1de06e9991acd968fceee899a8df3776d60
Submitter: Zuul
Branch: stable/ussuri

commit bbf9d1de06e9991acd968fceee899a8df3776d60
Author: Lee Yarwood <email address hidden>
Date: Wed Aug 5 23:00:06 2020 +0100

    libvirt: Provide VIR_MIGRATE_PARAM_PERSIST_XML during live migration

    The VIR_MIGRATE_PARAM_PERSIST_XML parameter was introduced in libvirt
    v1.3.4 and is used to provide the new persistent configuration for the
    destination during a live migration:

    https://libvirt.org/html/libvirt-libvirt-domain.html#VIR_MIGRATE_
PARAM_PERSIST_XML

    Without this parameter the persistent configuration on the destination
    will be the same as the original persistent configuration on the
source
    when the VIR_MIGRATE_PERSIST_DEST flag is provided.

    As Nova does not currently provide the VIR_MIGRATE_PARAM_PERSIST_XML
    param but does provide the VIR_MIGRATE_PERSIST_DEST flag this means
that
    a soft reboot by Nova of the instance after a live migration can
revert
    the domain back to the original persistent configuration from the
    source.

    Note that this is only possible in Nova as a soft reboot actually
    results in the virDomainShutdown and virDomainLaunch libvirt APIs
being
    called that recreate the domain using the persistent configuration.
    virDomainReboot does not result in this but is not called at this
time.

    The impact of this on the instance after the soft reboot is pretty
    severe, host devices referenced in the original persistent
configuration
    on the source may not exist or could even be used by other users on
the
```

```
    destination. CPU and NUMA affinity could also differ drastically
between
    the two hosts resulting in the instance being unable to start etc.

    As MIN_LIBVIRT_VERSION is now > v1.3.4 this change simply includes the
    VIR_MIGRATE_PARAM_PERSIST_XML param using the same updated XML for the
    destination as is already provided to VIR_MIGRATE_PARAM_DEST_XML.

    NOTE(lyarwood): A simple change to test_migrate_v3_unicode is included
    as Iccce0ab50eee515e533ab36c8e7adc10cb3f7019 had removed this from
    master.

    Co-authored-by: Tadayoshi Hosoya <email address hidden>
    Closes-Bug: #1890501
    Change-Id: Ia3f1d8e83cbc574ce5cb440032e12bbcb1e10e98
    (cherry picked from commit 1bb8ee95d4c3ddc3f607ac57526b75af1b7fbcff)
```

tags:added: in-stable-ussuri

Lee Yarwood (lyarwood) wrote on 2020-08-26: **Re: Soft reboot after live-migration reverts instance to original source domain XML (CVE-2020-17376)** #93

0001-libvirt-Provide-VIR_MIGRATE_PARAM_PERSIST_XML-newton.patch (9.5 KiB, text/plain)

```
I've been asked to provide a version of the patch against the newton-eol
tag.

Please find this attached, however I've not executed tox -e {pep8,unit,
functional} tests against it and it is just provided as a guide for how
this can be resolved for Newton.
```

OpenStack Infra (hudson-openstack) wrote on 2020-08-26: **Fix merged to nova (stable/train)** #94

```
Reviewed: https://review.opendev.org/747973
Committed: https://git.openstack.org/cgit/openstack/nova/commit/?
id=6a07edb4b29d8bfb5c86ed14263f7cd7525958c1
Submitter: Zuul
Branch: stable/train

commit 6a07edb4b29d8bfb5c86ed14263f7cd7525958c1
Author: Lee Yarwood <email address hidden>
Date: Wed Aug 5 23:00:06 2020 +0100

    libvirt: Provide VIR_MIGRATE_PARAM_PERSIST_XML during live migration

    The VIR_MIGRATE_PARAM_PERSIST_XML parameter was introduced in libvirt
    v1.3.4 and is used to provide the new persistent configuration for the
    destination during a live migration:

    https://libvirt.org/html/libvirt-libvirt-domain.html#VIR_MIGRATE_
PARAM_PERSIST_XML

    Without this parameter the persistent configuration on the destination
    will be the same as the original persistent configuration on the
source
    when the VIR_MIGRATE_PERSIST_DEST flag is provided.

    As Nova does not currently provide the VIR_MIGRATE_PARAM_PERSIST_XML
    param but does provide the VIR_MIGRATE_PERSIST_DEST flag this means
that
    a soft reboot by Nova of the instance after a live migration can
revert
    the domain back to the original persistent configuration from the
    source.

    Note that this is only possible in Nova as a soft reboot actually
    results in the virDomainShutdown and virDomainLaunch libvirt APIs
being
    called that recreate the domain using the persistent configuration.
    virDomainReboot does not result in this but is not called at this
time.

    The impact of this on the instance after the soft reboot is pretty
    severe, host devices referenced in the original persistent
configuration
    on the source may not exist or could even be used by other users on
the
    destination. CPU and NUMA affinity could also differ drastically
between
    the two hosts resulting in the instance being unable to start etc.

    As MIN_LIBVIRT_VERSION is now > v1.3.4 this change simply includes the
    VIR_MIGRATE_PARAM_PERSIST_XML param using the same updated XML for the
    destination as is already provided to VIR_MIGRATE_PARAM_DEST_XML.

    Co-authored-by: Tadayoshi Hosoya <email address hidden>
    Closes-Bug: #1890501
    Change-Id: Ia3f1d8e83cbc574ce5cb440032e12bbcb1e10e98
    (cherry picked from commit 1bb8ee95d4c3ddc3f607ac57526b75af1b7fbcff)
    (cherry picked from commit bbf9d1de06e9991acd968fceee899a8df3776d60)
```

OpenStack Infra (hudson-openstack) wrote on 2020-08-28: **Fix merged to nova (stable/stein)** #95

```
Reviewed: https://review.opendev.org/747974
Committed: https://git.openstack.org/cgit/openstack/nova/commit/?
id=b9ea91d17703f5b324a50727b6503ace0f4e95eb
Submitter: Zuul
Branch: stable/stein

commit b9ea91d17703f5b324a50727b6503ace0f4e95eb
Author: Lee Yarwood <email address hidden>
Date: Wed Aug 5 23:00:06 2020 +0100

    libvirt: Provide VIR_MIGRATE_PARAM_PERSIST_XML during live migration

    The VIR_MIGRATE_PARAM_PERSIST_XML parameter was introduced in libvirt
    v1.3.4 and is used to provide the new persistent configuration for the
    destination during a live migration:

    https://libvirt.org/html/libvirt-libvirt-domain.html#VIR_MIGRATE_
PARAM_PERSIST_XML

    Without this parameter the persistent configuration on the destination
    will be the same as the original persistent configuration on the
```

```
source
    when the VIR_MIGRATE_PERSIST_DEST flag is provided.

    As Nova does not currently provide the VIR_MIGRATE_PARAM_PERSIST_XML
    param but does provide the VIR_MIGRATE_PERSIST_DEST flag this means
that
    a soft reboot by Nova of the instance after a live migration can
revert
    the domain back to the original persistent configuration from the
    source.

    Note that this is only possible in Nova as a soft reboot actually
    results in the virDomainShutdown and virDomainLaunch libvirt APIs
being
    called that recreate the domain using the persistent configuration.
    virDomainReboot does not result in this but is not called at this
time.

    The impact of this on the instance after the soft reboot is pretty
    severe, host devices referenced in the original persistent
configuration
    on the source may not exist or could even be used by other users on
the
    destination. CPU and NUMA affinity could also differ drastically
between
    the two hosts resulting in the instance being unable to start etc.

    As MIN_LIBVIRT_VERSION is now > v1.3.4 this change simply includes the
    VIR_MIGRATE_PARAM_PERSIST_XML param using the same updated XML for the
    destination as is already provided to VIR_MIGRATE_PARAM_DEST_XML.

    Co-authored-by: Tadayoshi Hosoya <email address hidden>
    Closes-Bug: #1890501
    Change-Id: Ia3f1d8e83cbc574ce5cb440032e12bbcb1e10e98
    (cherry picked from commit 1bb8ee95d4c3ddc3f607ac57526b75af1b7fbcff)
    (cherry picked from commit bbf9d1de06e9991acd968fceee899a8df3776d60)
    (cherry picked from commit 6a07edb4b29d8bfb5c86ed14263f7cd7525958c1)
```

Jeremy Stanley (fungi) on 2020-08-30

```
Changed in ossa:
  assignee:nobody → Jeremy Stanley (fungi)
importance:Undecided → High
    status:In Progress → Fix Released
   summary:- Soft reboot after live-migration reverts instance to original source
          - domain XML (CVE-2020-17376)
          + [OSSA-2020-006] Soft reboot after live-migration reverts instance to
          + original source domain XML (CVE-2020-17376)
```

OpenStack Infra (hudson-openstack) wrote on 2020-09-03: **Fix merged to nova (stable/rocky)** #96

```
Reviewed: https://review.opendev.org/747975
Committed: https://git.openstack.org/cgit/openstack/nova/commit/?
id=c438fd9a0eb1903306a53ab44e3ae80660d8a429
Submitter: Zuul
Branch: stable/rocky

commit c438fd9a0eb1903306a53ab44e3ae80660d8a429
Author: Lee Yarwood <email address hidden>
Date: Wed Aug 5 23:00:06 2020 +0100

    libvirt: Provide VIR_MIGRATE_PARAM_PERSIST_XML during live migration

    The VIR_MIGRATE_PARAM_PERSIST_XML parameter was introduced in libvirt
    v1.3.4 and is used to provide the new persistent configuration for the
    destination during a live migration:

    https://libvirt.org/html/libvirt-libvirt-domain.html#VIR_MIGRATE_
PARAM_PERSIST_XML

    Without this parameter the persistent configuration on the destination
    will be the same as the original persistent configuration on the
source
    when the VIR_MIGRATE_PERSIST_DEST flag is provided.

    As Nova does not currently provide the VIR_MIGRATE_PARAM_PERSIST_XML
    param but does provide the VIR_MIGRATE_PERSIST_DEST flag this means
that
    a soft reboot by Nova of the instance after a live migration can
revert
    the domain back to the original persistent configuration from the
    source.

    Note that this is only possible in Nova as a soft reboot actually
    results in the virDomainShutdown and virDomainLaunch libvirt APIs
being
    called that recreate the domain using the persistent configuration.
    virDomainReboot does not result in this but is not called at this
time.

    The impact of this on the instance after the soft reboot is pretty
    severe, host devices referenced in the original persistent
configuration
    on the source may not exist or could even be used by other users on
the
    destination. CPU and NUMA affinity could also differ drastically
between
    the two hosts resulting in the instance being unable to start etc.

    As MIN_LIBVIRT_VERSION is now > v1.3.4 this change simply includes the
    VIR_MIGRATE_PARAM_PERSIST_XML param using the same updated XML for the
    destination as is already provided to VIR_MIGRATE_PARAM_DEST_XML.

    NOTE(lyarwood): As this is no longer the case from stable/rocky the
    change is slightly more involved introducing a persistent_xml_param
    kwarg that is used from _live_migration_operation within the driver
    based on the availability of libvirt v1.3.4 on the source host.

    Co-authored-by: Tadayoshi Hosoya <email address hidden>
    Closes-Bug: #1890501
    Change-Id: Ia3f1d8e83cbc574ce5cb440032e12bbcb1e10e98
    (cherry picked from commit 1bb8ee95d4c3ddc3f607ac57526b75af1b7fbcff)
    (cherry picked from commit bbf9d1de06e9991acd968fceee899a8df3776d60)
    (cherry picked from commit 6a07edb4b29d8bfb5c86ed14263f7cd7525958c1)
    (cherry picked from commit b9ea91d17703f5b324a50727b6503ace0f4e95eb)
```

OpenStack Infra (hudson-openstack) wrote on 2020-09-19: **Fix merged to nova (stable/queens)** #97

Download full text (3.2 KiB)

```
Reviewed: https://review.opendev.org/747976
Committed: https://git.openstack.org/cgit/openstack/nova/commit/?
id=a721ca5f510ce3c8ef24f22dac9e475b3d7651db
Submitter: Zuul
Branch: stable/queens

commit a721ca5f510ce3c8ef24f22dac9e475b3d7651db
Author: Lee Yarwood <email address hidden>
Date: Wed Aug 5 23:00:06 2020 +0100

    libvirt: Provide VIR_MIGRATE_PARAM_PERSIST_XML during live migration

    The VIR_MIGRATE_PARAM_PERSIST_XML parameter was introduced in libvirt
    v1.3.4 and is used to provide the new persistent configuration for the
    destination during a live migration:

    https://libvirt.org/html/libvirt-libvirt-domain.html#VIR_MIGRATE_
PARAM_PERSIST_XML

    Without this parameter the persistent configuration on the destination
    will be the same as the original persistent configuration on the
source
    when the VIR_MIGRATE_PERSIST_DEST flag is provided.

    As Nova does not currently provide the VIR_MIGRATE_PARAM_PERSIST_XML
    param but does provide the VIR_MIGRATE_PERSIST_DEST flag this means
that
    a soft reboot by Nova of the instance after a live migration can
revert
    the domain back to the original persistent configuration from the
    source.

    Note that this is only possible in Nova as a soft reboot actually
    results in the virDomainShutdown and virDomainLaunch libvirt APIs
being
    called that recreate the domain using the persistent configuration.
    virDomainReboot does not result in this but is not called at this
time.

    The impact of this on the instance after the soft reboot is pretty
    severe, host devices referenced in the original persistent
configuration
    on the source may not exist or could even be used by other users on
the
    destination. CPU and NUMA affinity could also differ drastically
between
    the two hosts resulting in the instance being unable to start etc.

    As MIN_LIBVIRT_VERSION is now > v1.3.4 this change simply includes the
    VIR_MIGRATE_PARAM_PERSIST_XML param using the same updated XML for the
    destination as is already provided to VIR_MIGRATE_PARAM_DEST_XML.

    Conflicts:
        nova/tests/unit/virt/libvirt/test_driver.py
        nova/tests/unit/virt/test_virt_drivers.py
        nova/virt/libvirt/driver.py
        nova/virt/libvirt/guest.py

    NOTE(lyarwood): Conflicts as If0a091a7441f2c3269148e40ececc3696d69684c
    (libvirt: Bump MIN_{LIBVIRT,QEMU}_VERSION for "Rocky"),
    Id9ee1feeadf612fa79c3d280cee3a614a74a00a7 (libvirt: Remove usage of
    migrateToURI{2} APIs) and I3af68f745ffb23ef2b5407ccec0bebf4b2645734
    (Remove mox in test_virt_drivers.py) are not present on stable/queens.
    As a result we can now add the parameter directly in
    _live_migration_operation before calling down into guest.migrate.

    Co-authored-by: Tadayoshi Hosoya <email address hidden>
    Closes-Bug: #1890501
    Change-Id: Ia3f1d8e83cbc574ce5cb440032e12bbcb1e10e98
    (cherry picked from commit 1bb8ee95d4c3ddc3f607ac57526b75af1b7fbcff)
    (cherry picked from commit bbf9d1de06e9991acd968fceee899a8df3776d60)
    (cherry picked from commit 6a07edb4b29d8bfb5c86ed14263f7cd7525958c1)
    (cherry picked from commit b9ea91d17...
```

Read more...

OpenStack Infra (hudson-openstack) wrote on 2020-09-25: **Fix merged to nova (stable/pike)** #98

Download full text (3.6 KiB)

```
Reviewed: https://review.opendev.org/747978
Committed: https://git.openstack.org/cgit/openstack/nova/commit/?
id=2faf17995dd9daa6f0b91e44be43264e447c678d
Submitter: Zuul
Branch: stable/pike

commit 2faf17995dd9daa6f0b91e44be43264e447c678d
Author: Lee Yarwood <email address hidden>
Date: Wed Aug 5 23:00:06 2020 +0100

    libvirt: Provide VIR_MIGRATE_PARAM_PERSIST_XML during live migration

    The VIR_MIGRATE_PARAM_PERSIST_XML parameter was introduced in libvirt
    v1.3.4 and is used to provide the new persistent configuration for the
    destination during a live migration:

    https://libvirt.org/html/libvirt-libvirt-domain.html#VIR_MIGRATE_
PARAM_PERSIST_XML

    Without this parameter the persistent configuration on the destination
    will be the same as the original persistent configuration on the
source
    when the VIR_MIGRATE_PERSIST_DEST flag is provided.

    As Nova does not currently provide the VIR_MIGRATE_PARAM_PERSIST_XML
    param but does provide the VIR_MIGRATE_PERSIST_DEST flag this means
that
    a soft reboot by Nova of the instance after a live migration can
revert
    the domain back to the original persistent configuration from the
    source.
    Note that this is only possible in Nova as a soft reboot actually
    results in the virDomainShutdown and virDomainLaunch libvirt APIs
being
    called that recreate the domain using the persistent configuration.
    virDomainReboot does not result in this but is not called at this
time.

    The impact of this on the instance after the soft reboot is pretty
    severe, host devices referenced in the original persistent
configuration
    on the source may not exist or could even be used by other users on
the
    destination. CPU and NUMA affinity could also differ drastically
between
    the two hosts resulting in the instance being unable to start etc.

    As MIN_LIBVIRT_VERSION is now > v1.3.4 this change simply includes the
    VIR_MIGRATE_PARAM_PERSIST_XML param using the same updated XML for the
    destination as is already provided to VIR_MIGRATE_PARAM_DEST_XML.

    Conflicts:
        nova/tests/unit/virt/libvirt/test_driver.py
        nova/virt/libvirt/driver.py

    NOTE(melwitt): Conflicts in driver.py are because changes:

      I6ac601e633ab2b0a67b4802ff880865255188a93
        (libvirt: Provide VGPU inventory for a single GPU type)
      I947bf0ad34a48e9182a3dc016f47f0c9f71c9d7b
        ([libvirt] Allow multiple volume attachments)
      Ibfa64f18bbd2fb70db7791330ed1a64fe61c1355
        (libvirt: QEMU native LUKS decryption for encrypted volumes)
      If2035cac931c42c440d61ba97ebc7e9e92141a28
        (libvirt: Rework 'EBUSY' (SIGKILL) error handling code path)
      Ibf210dd27972fed2651d6c9bd73a0bcf352c8bab
        (libvirt: create vGPU for instance)

    are not in Pike. Conflict in test_driver.py is because the Pike
    backport of change I9b545ca8aa6dd7b41ddea2d333190c9fbed19bc1
explicitly
    asserts byte string destination_xml in
    _test_live_migration_block_migration_flags and the change is not in
    Queens where this is being backported from.

    Co-authored-by: Tadayoshi Hosoya <<email address hidden>...
```

Read more...

OpenStack Infra (hudson-openstack) wrote on 2022-08-01: **Fix included in openstack/nova pike-eol** #99

```
This issue was fixed in the openstack/nova pike-eol release.
```

OpenStack Infra (hudson-openstack) wrote on 2022-11-11: **Fix included in openstack/nova queens-eol** #100

```
This issue was fixed in the openstack/nova queens-eol release.
```

OpenStack Infra (hudson-openstack) wrote on 2022-11-11: **Fix included in openstack/nova rocky-eol** #101

```
This issue was fixed in the openstack/nova rocky-eol release.
```

See full activity log

Displaying first 40 and last 40 comments. View all 101 comments or add a comment.