groups.google.com /g/qubes-devel/c/QwL5PjqPs-4

Disposable VM versus local forensics?

13-17 minutes

adre...@gmail.com

unread,

Sep 29, 2013, 6:02:11 PM9/29/13

to qubes...@googlegroups.com

Disposable VMs run entirely in RAM?

Let's assume for the sake of argument, that we're only up against local forensics. Not against network snooping.

When a Disposable VM is closed, is it guaranteed, that there are no traces on the local harddrive? (What about swap?)

Did anyone verify that claim using forensics? (One could image an installation of QubesOS, use a browser in a Disposable VM, image QubesOS again and then compare the images.)

If DispVMs defeat local forensics, I'd say this is a big advantage of QubesOS / DispVMs and should be advertised as such in documentation.

Joanna Rutkowska

unread,

Nov 8, 2013, 1:48:58 PM11/8/13

to qubes...@googlegroups.com

It does not.

Each DispVM uses volatile.img file that is used to create R/W illusion
for its root fs via COW (same mechanism as for any other AppVM based on
a template). This file is located in
/var/lib/qubes/appvms/<name-of-dispvm-template/volatile.img.

(BTW, an interesting peculiarity, which you, like myself, might be
wondering about: when another DispVM is started this volatile.img, used
by the previous DispVM is unlinked, and so the new DispVM uses a new
copy of the file, inode-wise).

Theoretically, one could modify the conf file used for dispvm creation,
specifically this line (e.g. in
/var/lib/qubes/appvms/fedora-18-x64-dvm/dvm.conf):

'script:file:/var/lib/qubes/appvms/fedora-18-x64-dvm/volatile.img,xvdc,w',

... and change the path to point to /dev/shm/qubes, where the DispVM
savefile is kept, and which is stored in RAM entirely (the whole
/dev/shm is a ram-based fs).

But, as Marek pointed out, a DispVM which might got to the dark side of
the force, might then generate write to its whole root fs and make this
vaoltile.img file to swallow up to 10GB in size, making the whole system

unusable if we kept it in /dev/shm.

Unless we could setup some quote on /dev/shm, can we?

Anyway, regarding your forensics concerns -- a rather easy and
convenient way to remove any records of a particular AppVM's activities
is to simply shred its private and volatile images:

```
shred private.img volatile.img
```

... and then remove it normally:

```
qvm-remove <appvm>
```

joanna.

Axon

unread,

Nov 8, 2013, 9:00:28 PM11/8/13

to qubes...@googlegroups.com

On 11/08/13 05:48, Joanna Rutkowska wrote:
> On 09/29/13 20:02, adre...@gmail.com wrote:
>> Disposable VMs run entirely in RAM?
>>
>> Let's assume for the sake of argument, that we're only up against local
>> forensics. Not against network snooping.
>>
>> When a Disposable VM is closed, is it guaranteed, that there are no traces
>> on the local harddrive? (What about swap?)
>>
>> Did anyone verify that claim using forensics? (One could image an
>> installation of QubesOS, use a browser in a Disposable VM, image QubesOS
>> again and then compare the images.)
>>
>> If DispVMs defeat local forensics, I'd say this is a big advantage of
>> QubesOS / DispVMs and should be advertised as such in documentation.
>>
>
> It does not.
>
> Each DispVM uses volatile.img file that is used to create R/W illusion
> for its root fs via COW (same mechanism as for any other AppVM based on
> a template). This file is located in
> /var/lib/qubes/appvms/<name-of-dispvm-template/volatile.img.
>

So, what does the "Keep dispVM in memory" option do? (Qubes VM Manager
--> System --> Global Settings)

> (BTW, an interesting peculiarity, which you, like myself, might be
> wondering about: when another DispVM is started this volatile.img, used
> by the previous DispVM is unlinked, and so the new DispVM uses a new
> copy of the file, inode-wise).
>
> Theoretically, one could modify the conf file used for dispvm creation,
> specifically this line (e.g. in
> /var/lib/qubes/appvms/fedora-18-x64-dvm/dvm.conf):
>

> 'script:file:/var/lib/qubes/appvms/fedora-18-x64-dvm/volatile.img,xvdc,w',
>
> ... and change the path to point to /dev/shm/qubes, where the DispVM
> savefile is kept, and which is stored in RAM entirely (the whole
> /dev/shm is a ram-based fs).
>
> But, as Marek pointed out, a DispVM which might got to the dark side of
> the force, might then generate write to its whole root fs and make this
> vaoltile.img file to swallow up to 10GB in size, making the whole system
> unusable if we kept it in /dev/shm.
>
> Unless we could setup some quote on /dev/shm, can we?
>
> Anyway, regarding your forensics concerns -- a rather easy and
> convenient way to remove any records of a particular AppVM's activities
> is to simply shred its private and volatile images:
>
> shred private.img volatile.img

But this is probably ineffective on certain file systems (see man page)
or on SSDs (due to wear leveling), correct?

>
> ... and then remove it normally:
>
> qvm-remove <appvm>
>
> joanna.
>

It sounds like the DispVM page
(http://qubes-os.org/trac/wiki/DisposableVms) should probably be updated
to reflect this information, since I think most Qubes users have assumed
that DispVMs run entirely in RAM. If you agree, I can update the page
(though I still do not completely understand how DispVMs actually work).

adrelanos

unread,

Nov 8, 2013, 10:13:19 PM11/8/13

to qubes...@googlegroups.com

Joanna Rutkowska:

> Unless we could setup some quote on /dev/shm, can we?

Probably. I found some suggestions. All untested.

a)
Create a file and mount it as a folder.

```
dd if=/dev/zero of=/userdisk.fs bs=1024 count= 100000
mkfs.ext3 /userdisk.fs
mount -t ext3 -o loop /userdisk.fs /home/user
```

b)
Fedora documentation says, "Disk quotas can be configured for individual
users as well as user groups." [1] -> Have a group "dispvm", dynamically
create a folder in /dev/shm, assign ownership to "dispvm" and assign xx%
of available RAM (configurable) (by setting quota for that group).

All dispvm would share the same quota then and one could compromised
dispvm could needlessly fill up the RAM. Well, that could maybe be
solved as well dynamically creating a new group per dispvm and assigning
it xx% of available RAM.

c)
fs_setquota
"Sets the quota for [...] directory."

It's in Ubuntu repositories but my little Fedora knowledge doesn't let
me find out if it or a similar tool is available in Fedora as well.

[1]
https://docs.fedoraproject.org/en-US/Fedora/14/html/Storage_Administration_Guide/ch-disk-quotas.html

Joanna Rutkowska

unread,

Nov 8, 2013, 10:53:00 PM11/8/13

to qubes...@googlegroups.com

On 11/08/13 22:00, Axon wrote:
> On 11/08/13 05:48, Joanna Rutkowska wrote:
>> On 09/29/13 20:02, adre...@gmail.com wrote:
>>> Disposable VMs run entirely in RAM?
>>>
>>> Let's assume for the sake of argument, that we're only up against local
>>> forensics. Not against network snooping.
>>>
>>> When a Disposable VM is closed, is it guaranteed, that there are no traces
>>> on the local harddrive? (What about swap?)
>>>
>>> Did anyone verify that claim using forensics? (One could image an
>>> installation of QubesOS, use a browser in a Disposable VM, image QubesOS
>>> again and then compare the images.)
>>>
>>> If DispVMs defeat local forensics, I'd say this is a big advantage of
>>> QubesOS / DispVMs and should be advertised as such in documentation.
>>>
>>
>> It does not.
>>
>> Each DispVM uses volatile.img file that is used to create R/W illusion
>> for its root fs via COW (same mechanism as for any other AppVM based on
>> a template). This file is located in
>> /var/lib/qubes/appvms/<name-of-dispvm-template/volatile.img.
>>
>
> So, what does the "Keep dispVM in memory" option do? (Qubes VM Manager
> --> System --> Global Settings)
>

The intention was for this to control whether the *savefile* is kept in
RAM (to speedup DispVM startup) or not (not to consume precious RAM).

But, from a quick grepping throught the sources, it appears to me that
it does nothing. I.e. it only touches
/var/lib/qubes/dvmdata/dont_use_shm, but the latter is never used by any
of the core code.

So, I think we should make this option grayed out in the manager, and
also change the label to something like: "Keep the DispVM savefile in
RAM to speed up DispVM startup" to make it clear what the intention is.
In the future we might add another switch that would be forcing the
volatile.img to be kept on /dev/shm as well, as discussed in the
previous message.

Marek, am I missing something here?

>> (BTW, an interesting peculiarity, which you, like myself, might be
>> wondering about: when another DispVM is started this volatile.img, used
>> by the previous DispVM is unlinked, and so the new DispVM uses a new
>> copy of the file, inode-wise).
>>
>> Theoretically, one could modify the conf file used for dispvm creation,
>> specifically this line (e.g. in
>> /var/lib/qubes/appvms/fedora-18-x64-dvm/dvm.conf):
>>
>> 'script:file:/var/lib/qubes/appvms/fedora-18-x64-dvm/volatile.img,xvdc,w',
>>
>> ... and change the path to point to /dev/shm/qubes, where the DispVM
>> savefile is kept, and which is stored in RAM entirely (the whole
>> /dev/shm is a ram-based fs).
>>
>> But, as Marek pointed out, a DispVM which might got to the dark side of
>> the force, might then generate write to its whole root fs and make this
>> vaoltile.img file to swallow up to 10GB in size, making the whole system
>> unusable if we kept it in /dev/shm.
>>
>> Unless we could setup some quote on /dev/shm, can we?
>>
>> Anyway, regarding your forensics concerns -- a rather easy and
>> convenient way to remove any records of a particular AppVM's activities
>> is to simply shred its private and volatile images:
>>
>> shred private.img volatile.img
>
> But this is probably ineffective on certain file systems (see man page)

In practice journals only backup metadata, not actual blocks, at least
this is the case for the fs we have in Dom0.

> or on SSDs (due to wear leveling), correct?
>

Never investigated this from a practical persepctive, I tend to believe
the risk is just theoretical.

>>
>> ... and then remove it normally:
>>
>> qvm-remove <appvm>
>>
>> joanna.
>>
>
> It sounds like the DispVM page
> (http://qubes-os.org/trac/wiki/DisposableVms) should probably be updated
> to reflect this information, since I think most Qubes users have assumed
> that DispVMs run entirely in RAM. If you agree, I can update the page
> (though I still do not completely understand how DispVMs actually work).
>

Sure.

joanna.

Joanna Rutkowska

unread,

Nov 8, 2013, 10:54:49 PM11/8/13

to qubes...@googlegroups.com

On 11/08/13 23:13, adrelanos wrote:
> Joanna Rutkowska:
>> Unless we could setup some quote on /dev/shm, can we?
>
> Probably. I found some suggestions. All untested.
>

:)

I'm sure you will understand that we really don't have resources to work
on this at the moment. But we would gladly accept a patch for this
(including the check box for manager to enable/disable this :).

joanna.

adrelanos

unread,

Nov 8, 2013, 11:16:27 PM11/8/13

to qubes...@googlegroups.com

Joanna Rutkowska:

> On 11/08/13 23:13, adrelanos wrote:
>> Joanna Rutkowska:
>>> Unless we could setup some quote on /dev/shm, can we?
>>
>> Probably. I found some suggestions. All untested.
>>
> :)
>
> I'm sure you will understand that we really don't have resources to work
> on this at the moment.

Sure. I know the feeling being surrounded by great ideas and suggestions
and not having resources to implement them due to my work on Whonix.
Just couldn't leave your (eventually rhetorical) question unanswered.
Always good to have an overview about possibilities. :)

> But we would gladly accept a patch for this
> (including the check box for manager to enable/disable this :).

I am not capable myself (yet), but perhaps someone else gets inspired by
this discussion or perhaps you'll get to it someday. :)

Marek Marczykowski-Górecki

unread,

Nov 8, 2013, 11:24:18 PM11/8/13

to qubes...@googlegroups.com, Joanna Rutkowska

Ough, one place where path wasn't converted to use '-' instead of '_'.
Check /usr/lib/qubes/startup-dvm.sh

--
Best Regards,
Marek Marczykowski-Górecki
Invisible Things Lab
A: Because it messes up the order in which people normally read text.
Q: Why is top-posting such a bad thing?

vit...@gmail.com

unread,

May 7, 2015, 6:34:29 PM5/7/15

to qubes...@googlegroups.com, joa...@invisiblethingslab.com

I am not sure if this is better to post here or to the Github issue (i.e. https://github.com/QubesOS/qubes-issues/issues/904).

I've some scripts that I use for a temporary swapfile and temporary filesystem. They use a random key (from /dev/random). The tmp filesystem uses some configuration for better performance by disabling some features like journaling. (We don't need journaling for filesystem that are expected to be unreadable after reboot…)

My current usage:
1. The swapfile is attached automatically added in the background after 120 seconds in order not to block by reading from /dev/random during the system boot.
2. The largetmp is mounted only when needed, i.e. manually. (Yes, the usage of sudo suggests that…) I usually use tmpfs, but when I need something large, I mount the largetmp. I was thinking about automount of largetmp, but I was unsure about safety and some other potential issues. (Moreover, largetmp lies on rotational HDD, so using it instead of tmpfs could cause much more HDD usage and more power consumption.)

Some security considerations:
1. It is essential to handle safely situation when largetmp is not mounted. This is the reason why I use /tmp/large and not /large/tmp. If I forget to mount it, the worst thing that can happen is writing large amount of data to RAM. If it was in /large/tmp, accidental writing to a less protected partition (i.e. /large or /, which depends on the setup) may happen if the largetmp1 is not mounted. (Which is what I've once accidentally done. It was followed by several days of continuous wiping…)
2. The /dev/random is usually seeded from saved random seed. When some wear-levelling or relocation is used, the random seed might be available to local forensics, which could reduce the effective entropy of the key in the considered case. (Fortunatelly, the Qubes login screen requires some keystrokes, which adds some entrophy.)

There are the scripts and crypttab lines: https://gist.github.com/v6ak/3171313bc2c22efc263d

Regards,
Vít Šesták 'v6ak'