NAME

guestfs−recipes − libguestfs, guestfish and virt tools recipes

DESCRIPTION

This page contains recipes for and links to things you can do using libguestfs, **guestfish** (1) and the virt tools.

Access a remote disk image using guestfish

If the disk image is on a remote server which is accessible using SSH, HTTP, FTP, NBD, iSCSI, or similar, then you can open it directly. See ''ADDING REMOTE STORAGE'' in **guestfish** (1) for several examples. This requires libguestfs ≥ 1.22 and qemu ≥ 1.5.

Audit a virtual machine for setuid files

See: ''EXAMPLES'' in **virt−ls** (1).

Audit a virtual machine for vulnerabilities and security problems

See:
https://rwmj.wordpress.com/2013/05/16/scanning−offline−guests−using−openscap−and−guestmount/#content

Change the background image in a Windows XP VM

The links below explain how to use **guestfish** (1) to change the background image for a user of a Windows XP VM. Unfortunately the technique appears to be substantially different for each version of Windows.

https://lists.fedoraproject.org/pipermail/virt/2011−May/002655.html
https://lists.fedoraproject.org/pipermail/virt/2011−May/002658.html

Checksum a file or device within a disk image

To checksum a whole device, or a partition, LV etc within a disk image:

```
guestfish −−ro −a disk.img run : checksum−device md5 /dev/sda1
```

Replace `md5` with the type of checksum you want. See ''guestfs_checksum_device'' in **guestfs** (3) for a list of supported types.

/dev/sda1 means ''the first partition''. You could use */dev/sda* to checksum the whole disk image, or the name of a logical volume or RAID device.

To checksum a single file:

```
guestfish −−ro −a disk.img −i checksum sha256 /etc/passwd
```

or for a Windows guest:

```
guestfish −−ro −a disk.img −i \
  checksum sha256 'win:\windows\system32\config\SOFTWARE'
```

Cloning a virtual machine

Use a combination of tools like **cp** (1), **dd** (1), and virt tools like **virt−sysprep** (1), **virt−sparsify** (1) and **virt−resize** (1).

For more details, see: ''COPYING AND CLONING'' in **virt−sysprep** (1).

Convert a CD-ROM / DVD / ISO to a tarball

This converts input *cd.iso* to output *cd.tar.gz*:

```
guestfish −−ro −a cd.iso −m /dev/sda tgz−out / cd.tar.gz
```

To export just a subdirectory, eg. */files*, do:

```
guestfish −−ro −a cd.iso −m /dev/sda tgz−out /files cd.tar.gz
```

Convert from one format/filesystem to another

If you have a data disk in one format / filesystem / partition / volume manager, you can convert it another using this technique.

In this example, we start with a data disk that has a single partition containing a filesystem, and we want to create another disk that contains the same files but on an ext3 filesystem embedded in a logical volume on a sparse raw-format disk.

First create the formatted-but-empty target disk:

```
truncate −s 10G target.img
virt−format −a target.img --partition=mbr --lvm --filesystem=ext3
```

Now, pipe two guestfish instances together to transfer the old data to the new disk:

```
guestfish --ro −a source.img −m /dev/sda1  -- tar−out / − | \
guestfish --rw −a target.img −m /dev/VG/LV -- tar−in − /
```

To browse the final disk image, do:

```
guestfish --ro −a target.img −m /dev/VG/LV
><fs> ll /
```

This technique is quite powerful, allowing you for example to split up source directories over the target filesystems.

Note this won't work (at least, not directly) for bootable virtual machine disks because it doesn't copy over the boot loader.

Convert Windows DVD to bootable USB key

http://rwmj.wordpress.com/2013/05/09/tip−convert−a−windows−dvd−iso−to−a−bootable−usb−key−using−guestfish/#con

Convert Xen-style partitionless image to partitioned disk image

Xen disk images are often partitionless, meaning that the filesystem starts directly at the beginning of the disk with no partition table. You can in fact use these directly in KVM (provided the guest isn't Windows), but some people like to convert them to regular partitioned disk images, and this is required for Windows guests. Here is how to use guestfish to do this:

```
guestfish
><fs> add−ro input.img
><fs> sparse output.img 10G     # adjust the output size
><fs> run
# Create a partition table on the output disk:
><fs> part−init /dev/sdb mbr
><fs> part−add /dev/sdb p 2048 −2048
# Copy the data to the target partition:
><fs> copy−device−to−device /dev/sda /dev/sdb1 sparse:true
# Optionally resize the target filesystem.  Use ntfsresize
# for Windows guests:
><fs> resize2fs /dev/sdb1
```

Such a disk image won't be directly bootable. You may need to boot it with an external kernel and initramfs (see below). Or you can use the guestfish commands`syslinux` or `extlinux` to install a SYSLINUX bootloader.

Create empty disk images

The **virt−format** (1) tool can do this directly.

Use **virt−make−fs** (1) to create a disk image with content. This can also create some standard disk images such as virtual floppy devices (VFDs).

You can also use the **guestfish** (1) *−N* option to create empty disk images. The useful guide below explains the options available.

https://rwmj.wordpress.com/2010/09/08/new−guestfish−n−options−in−1−5−9/#content

virt−builder (1) can create minimal guests.

Delete a file (or other simple file operations)

Use guestfish. To delete a file:

```
guestfish −a disk.img −i rm /file/to/delete
```

To touch a file (bring it up to date or create it):

```
guestfish −a disk.img −i touch /file/to/touch
```

To stat a file. Since this is a read-only operation, we can make it safer by adding the −−*ro* flag.

```
guestfish --ro −a disk.img −i stat /file/to/stat
```

There are dozens of these commands. See **guestfish** (1) or the output of `guestfish −h`

Diff two guests; compare a snapshot to the current version

Since libguestfs ≥ 1.26, use **virt−diff** (1) to look for differences between two guests (for example if they were originally cloned from the same source), or between two snapshots from the same guest. In earlier versions of libguestfs, use **virt−ls** (1).

Disable a systemd service

The following is the equivalent of `systemctl mask ...`. To disable the `cloud−init` service so it doesn't start at next boot:

```
guestfish −a disk.img −i \
    ln−sf /dev/null /etc/systemd/system/cloud−init.service
```

To disable tmp-on-tmpfs:

```
guestfish −a disk.img −i \
    ln−sf /dev/null /etc/systemd/system/tmp.mount
```

One problem with the commands above is there is no feedback if you get the name of the service you are trying to mask wrong. But you can use **virt−ls** (1) to list the available systemd services like this:

```
virt−ls −a /tmp/fedora−19.img −R /lib/systemd/system
```

Drive letters over FUSE

You have a Windows guest, and you want to expose the drive letters as FUSE mountpoints (*/C/...*, */D/...* etc). Instead of **guestmount** (1), use this Perl script:

```
#!/usr/bin/perl −w
use strict;
use Sys::Guestfs;
$| = 1;
die "usage: $0 mountpoint disk.img" if @ARGV < 2;
my $mp = shift @ARGV;
my $g = new Sys::Guestfs;
$g−>add_drive_opts ($_) foreach @ARGV;
$g−>launch;
my @roots = $g−>inspect_os;
die "$0: no operating system found" if @roots != 1;
my $root = $roots[0];
die "$0: not Windows" if $g−>inspect_get_type ($root) ne "windows";
my %map = $g−>inspect_get_drive_mappings ($root);
foreach (keys %map) {
    $g−>mkmountpoint ("/$_");
    eval { $g−>mount ($map{$_}, "/$_") };
    warn "$@ (ignored)\n" if $@;
}
$g−>mount_local ($mp);
print "filesystem ready on $mp\n";
$g−>mount_local_run;
$g−>shutdown;
```

You can use the script like this:

```
$ mkdir /tmp/mnt
$ ./drive−letters.pl /tmp/mnt windows7.img
filesystem ready on /tmp/mnt
```

In another window:

```
$ cd /tmp/mnt
$ ls
C  D
$ cd C
$ ls
Documents and Settings
PerfLogs
ProgramData
Program Files
[etc]
$ cd ../..
$ guestunmount /tmp/mnt
```

Dump raw filesystem content from inside a disk image or VM

You can use the **guestfish** (1) `download` command to extract the raw filesystem content from any filesystem in a disk image or a VM (even one which is encrypted or buried inside an LV or RAID device):

```
guestfish −−ro −a disk.img run : download /dev/sda1 sda1.img

guestfish −−ro −d Guest run : download /dev/vg_guest/lv_root lv.img
```

To download to stdout, replace the filename with a `−` character:

```
guestfish −−ro −a disk.img run : download /dev/sda1 − | gzip > sda1.gz
```

To list the filesystems in a disk image, use **virt−filesystems** (1).

See also ''Uploading raw filesystem content''.

Edit grub configuration in a VM

You can use this to:

- Fix a virtual machine that does not boot.
- Change which kernel is used to boot the VM.
- Change kernel command line options.

Use **virt−edit** (1) to edit the grub configuration:

```
virt−edit −d BrokenGuest /boot/grub2/grub.cfg
```

or for general tinkering inside an unbootable VM use **virt−rescue** (1) like this:

```
virt−rescue −d BrokenGuest
```

Export any directory from a VM

To export */home* from a VM into a local directory use **virt−copy−out** (1):

```
virt−copy−out −d Guest /home .
```

Notes:

- The final dot of the command is not a printing error. It means we want to copy out to the current directory.
- This creates a directory called `home` under the current directory.

If the guest is a Windows guest then you can use drive letters and backslashes, but you must prefix the path with `win:` and quote it to protect it from the shell, like this:

```
virt−copy−out −d WinGuest 'win:c:\windows\system32\config' .
```

To get the output as a compressed tarball, do:

```
virt−tar−out −d Guest /home − | gzip −−best > home.tar.gz
```

Although it sounds tempting, this is usually not a reliable way to get a backup from a running guest. See the entry in the FAQ: http://libguestfs.org/FAQ.html#backup

Export external kernel and initramfs (initrd)

If a Linux guest doesn't have a boot loader or it is broken, then you can usually boot it using an external kernel and initramfs. In this configuration, the hypervisor acts like a bootloader, loading the kernel from the host disk into guest memory and jumping straight into the kernel.

However you may wonder how to get the right kernel corresponding to the disk image you have. Since libguestfs ≥ 1.24 **virt−builder** (1) can get the latest kernel and corresponding initramfs for you:

```
mkdir outputdir
virt−builder −−get−kernel disk.img −o outputdir
ls −lh outputdir
```

Find out which user is using the most space

This simple script examines a Linux guest to find out which user is using the most space in their home directory:

```
#!/bin/sh −

set −e

vm="$1"
dir=/home

eval $(guestfish −−ro −d "$vm" −i −−listen)

for d in $(guestfish −−remote ls "$dir"); do
    echo −n "$dir/$d"
    echo −ne '\t'
    guestfish −−remote du "$dir/$d";
done | sort −nr −k 2

guestfish −−remote exit
```

Get DHCP address from a VM

The link below explains the many different possible techniques for getting the last assigned DHCP address of a virtual machine.

https://rwmj.wordpress.com/2011/03/31/tip−code−for−getting−dhcp−address−from−a−virtual−machine−disk−image/#co

In the libguestfs source examples directory you will find the latest version of the *virt−dhcp−address.c* program.

Get the operating system product name string

Save the following script into a file called *product−name.sh*:

```
#!/bin/sh −
set −e
eval "$(guestfish −−ro −d "$1" −−i −−listen)"
root="$(guestfish −−remote inspect−get−roots)"
guestfish −−remote inspect−get−product−name "$root"
guestfish −−remote exit
```

Make the script executable and run it on a named guest:

```
# product−name.sh RHEL60x64
Red Hat Enterprise Linux Server release 6.0 (Santiago)
```

You can also use an XPath query on the **virt−inspector** (1) XML using the `xpath` command line tool or from your favourite programming language:

```
# virt−inspector RHEL60x64 > xml
# xpath '//product_name' < xml
Found 1 nodes:
−− NODE −−
<product_name>Red Hat Enterprise Linux Server release 6.0 (Santiago)</product_na
```

Get the default boot kernel for a Linux VM

The link below contains a program to print the default boot kernel for a Linux VM.

https://rwmj.wordpress.com/2010/10/30/tip−use−augeas−to−get−the−default−boot−kernel−for−a−vm/#content

It uses Augeas, and the technique is generally applicable for many different tasks, such as:

- listing the user accounts in the guest
- what repositories is it configured to use
- what NTP servers does it connect to
- what were the boot messages last time it booted
- listing who was logged in recently

http://augeas.net/

Hanging guests

There are various ways to use libguestfs to find out why a guest is hanging or unresponsive:

1. Read the log files using virt-cat:

   ```
   virt−cat Guest /var/log/messages | less
   ```

2. Read the Windows Event Log (Windows Vista or later only):

   https://rwmj.wordpress.com/2011/04/17/decoding−the−windows−event−log−using−guestfish/#content

3. Find out which files were last updated in a guest:

   https://rwmj.wordpress.com/2012/02/27/using−libguestfs−to−find−out−why−a−windows−guest−was−hanging/#cont

   This might give you a clue as to what program is running.

Hex-dumping sectors from the guest

Hex-dump the boot partition (Master Boot Record / first sector):

```
guestfish −−ro −a disk.img run : pread−device /dev/sda 0x200 0 |
  hexdump −C
```

(`0x200` = 512 bytes which is the size of traditional PC sectors)

To hexdump the N'th partition, substitute a number for `N` in the following command:

```
guestfish −−ro −a disk.img \
    run : pread−device /dev/sda 0x200 $((N*0x200)) |
  hexdump −C
```

Hex-editing sectors in the guest

Hex-edit the boot partition (Master Boot Record / first sector):

```
guestfish −−rw −a disk.img run : hexedit /dev/sda 0x200
```

Install packages (RPMs, Debian packages) in a guest

Since libguestfs 1.26, **virt−builder** (1), **virt−customize** (1) and **virt−sysprep** (1) have an −−*install* option for installing packages in Linux guests. (Use virt-customize if you have an existing guest, or virt-builder if you want to create a guest from scratch).

For example:

```
virt−builder fedora−20 −−install emacs
```

Install packages from an alternate repository

Since libguestfs 1.26, you can use **virt−builder** (1), **virt−customize** (1) or **virt−sysprep** (1) −−*edit* option to edit repository metadata before installing packages

For example this would install packages from the updates-testing repository in Fedora:

```
virt−builder fedora−20 \
  −−edit '/etc/yum.repos.d/fedora−updates−testing.repo:
            s/enabled=0/enabled=1/' \
  −−install emacs
```

Install SYSLINUX bootloader in a guest

SYSLINUX is a small, easy to configure bootloader for Linux and Windows guests. If your guest is not bootable, you can install the SYSLINUX bootloader using either the guestfish commands `syslinux` (for FAT-based guests) or `extlinux` (for ext2/3/4 and btrfs-based guests).

This guide assumes a Linux guest where */dev/sda1* is */boot*, */boot/vmlinuz* is the guest kernel, and */dev/sda3* is the root partition. For a Windows guest you would need a FAT-formatted boot partition and you would need to use the `syslinux` command instead.

Create a *syslinux.cfg* configuration file. You should check the SYSLINUX documentation at http://www.syslinux.org but it may look something like this:

```
DEFAULT linux
LABEL linux
  SAY Booting the kernel
  KERNEL vmlinuz
  INITRD initrd
  APPEND ro root=/dev/sda3
```

Locate the syslinux master boot record (a file called something like */usr/share/syslinux/mbr.bin*).

```
guestfish −a disk.img −i
# Upload the master boot record and configuration file:
><fs> upload ..../mbr.bin /boot/mbr.bin
><fs> upload ..../syslinux.cfg /boot/syslinux.cfg
# Put the MBR into the boot sector:
><fs> copy−file−to−device /boot/mbr.bin /dev/sda size:440
# Install syslinux on the first partition:
><fs> extlinux /boot
# Set the first partition as bootable:
><fs> part−set−bootable /dev/sda 1 true
```

See also: http://rwmj.wordpress.com/2013/04/04/new−in−libguestfs−use−syslinux−or−extlinux−to−make−bootable−guests/#conten

List applications installed in a VM

Save the following to a file *list−apps.sh*:

```
#!/bin/sh −
set −e
eval "$(guestfish −−ro −d "$1" −−i −−listen)"
root="$(guestfish −−remote inspect−get−roots)"
guestfish −−remote inspect−list−applications "$root"
guestfish −−remote exit
```

Make the file executable and then you can run it on any named virtual machine:

```
# list−apps.sh WinGuest
[0] = {
  app_name: Mozilla Firefox (3.6.12)
  app_display_name: Mozilla Firefox (3.6.12)
  app_epoch: 0
  app_version: 3.6.12 (en−GB)
  app_release:
  app_install_path: C:\Program Files\Mozilla Firefox
  app_trans_path:
  app_publisher: Mozilla
  app_url: http://www.mozilla.com/en−GB/
  app_source_package:
  app_summary:
  app_description: Mozilla Firefox
}
[1] = {
  app_name: VLC media player
  app_display_name: VLC media player 1.1.5
  app_epoch: 0
  app_version: 1.1.5
  app_release:
  app_install_path: C:\Program Files\VideoLAN\VLC
  app_trans_path:
  app_publisher: VideoLAN
  app_url: http://www.videolan.org/
  app_source_package:
  app_summary:
  app_description:
}
```

If you want to run the script on disk images (instead of libvirt virtual machines), change `−d "$1"` to `−a "$1"`. See also**virt−inspector** (1).

List files and directories in a VM

Use **virt−ls** (1).

List services in a Windows VM

The link below contains a script that can be used to list out the services from a Windows VM, and whether those services run at boot time or are loaded on demand.

https://rwmj.wordpress.com/2010/12/10/tip−list−services−in−a−windows−guest/#content

Make a disk image sparse

Use **virt−sparsify** (1).

Monitor disk usage over time

You can use **virt−df** (1) to monitor disk usage of your guests over time. The link below contains a guide.

http://web.archive.org/web/20130214073726/http://virt−tools.org/learning/advanced−virt−df/

Reading the Windows Event Log from Windows Vista (or later)

guestfish (1) plus the tools described in the link below can be used to read out the Windows Event Log from any virtual machine running Windows Vista or a later version.

https://rwmj.wordpress.com/2011/04/17/decoding−the−windows−event−log−using−guestfish/#content

Remove root password (Linux)

Using the **virt−edit** (1) *−e* option you can do simple replacements on files. One use is to remove the root password from a Linux guest:

```
 virt-edit -d domname /etc/passwd -e 's/^root:.*?:/root::/'

 virt-edit -a disk.img /etc/passwd -e 's/^root:.*?:/root::/'
```

Remove Administrator password (Windows)

The link below contains one technique for removing the Administrator password from a Windows VM, or to be more precise, it gives you a command prompt the next time you log in which you can use to bypass any security:

https://mdbooth.wordpress.com/2010/10/18/resetting−a−windows−guests−administrator−password−with−guestfish/

Sysprepping a virtual machine (Windows)

It is possible to do a ‘‘sysprep’’ using libguestfs alone, although not straightforward. Currently there is code in the Aeolus Oz project which does this (using libguestfs). It is likely we will add this to **virt−sysprep** (1) in future.

https://github.com/clalancette/oz
https://www.redhat.com/archives/virt−tools−list/2011−May/msg00019.html

Unpack a live CD

Linux live CDs often contain multiple layers of disk images wrapped like a Russian doll. You can use **guestfish** (1) to look inside these multiple layers, as outlined in the guide below.

https://rwmj.wordpress.com/2009/07/15/unpack−the−russian−doll−of−a−f11−live−cd/#content

Uploading and downloading files

The link below contains general tips on uploading (copying in) and downloading (copying out) files from VMs.

https://rwmj.wordpress.com/2010/12/02/tip−uploading−and−downloading/#content

Uploading raw filesystem content

You can use **guestfish** (1) to upload whole filesystems into a VM, even into a filesystem which is encrypted or buried inside an LV or RAID device:

```
 guestfish --rw -a disk.img run : upload sda1.img /dev/sda1

 guestfish --rw -d Guest run : upload lv.img /dev/vg_guest/lv_root
```

One common problem is that the filesystem isn’t the right size for the target. If it is too large, there’s not much you can do with libguestfs − you have to prepare the filesystem differently. But if the filesystem needs to expand into the target, you can use guestfish to resize it to the right size:

```
 guestfish --rw -d Guest run : \
   upload lv.img /dev/vg_guest/lv_root : \
   resize2fs /dev/vg_guest/lv_root
```

(or use `ntfsresize` if the filesystem is NTFS).

Use libguestfs tools on VMware ESX guests

The link below explains how to use libguestfs, **guestfish** (1) and the virt tools on any VMware ESX guests, by first sharing the VMware VMFS over sshfs.

https://rwmj.wordpress.com/2011/05/10/tip−use−libguestfs−on−vmware−esx−guests/#content

SEE ALSO

guestfs (3), **guestfish** (1), **guestfs−examples** (3), **guestfs−erlang** (3), **guestfs−gobject** (3), **guestfs−golang** (3), **guestfs−java** (3), **guestfs−lua** (3), **guestfs−ocaml** (3), **guestfs−perl** (3), **guestfs−python** (3), **guestfs−ruby** (3), http://libguestfs.org/.

AUTHORS

Richard W.M. Jones (`rjones at redhat dot com`)

COPYRIGHT

LICENSE

BUGS

To get a list of bugs against libguestfs, use this link: https://bugzilla.redhat.com/buglist.cgi?component=libguestfs&product=Virtualization+Tools

To report a new bug against libguestfs, use this link: https://bugzilla.redhat.com/enter_bug.cgi?component=libguestfs&product=Virtualization+Tools

When reporting a bug, please supply:

- The version of libguestfs.
- Where you got libguestfs (eg. which Linux distro, compiled from source, etc)
- Describe the bug accurately and give a way to reproduce it.
- Run **libguestfs−test−tool** (1) and paste the **complete, unedited** output into the bug report.