

## APPENDIX

### ARTIFACT DESCRIPTION

We present in this section the reproducibility artifact of our implementation and experimental validation, following the template currently in development proposed by the IEEE Transactions on Parallel and Distributed Systems journal.

#### A. How delivered

We prototyped Expelliarmus in Python with an easy command line interface released on GitHub<sup>1</sup>.

#### B. Hardware dependencies

We implemented and tested Expelliarmus on an Intel core i7-7600U CPU@2.80GHz quadcore machine with 16 GB of memory and 1 TB of SSD storage. The software, however, is portable and has no such hardware dependencies.

#### C. Software dependencies

We tested Expelliarmus on an Ubuntu 16.04 LTS system, however, it can be deployed on any Linux-based environment. The software dependencies include libguestfs, qemu, and sqlite.

#### D. Data sets

We created each VMI using virt-builder, an efficient tool to build a variety of images for local and Cloud use. The minimal script to create a VMI using virt-builder is listed below:

```
#!/bin/bash
# path to libguestfs
../../libguestfs-1.36.13/run virt-builder
→ ubuntu-16.04 \
-o Image.qcow2 \
--size 4G \
--format qcow2 \
--root-password password:123 \
--edit '/etc/default/grub:s/
→ ^GRUB_CMDLINE_LINUX_DEFAULT=
→ .*/GRUB_CMDLINE_LINUX_DEFAULT=
→ "console=tty0
→ console=ttyS0,115200n8"/' \
--run-command update-grub \
--install "dpkg-repack, fakeroot, dpkg-dev"
→ \
--update \
--write /etc/default/locale:
→ $"LANG=\"en_US.UTF-8\"" \
--append /etc/default/locale:
→ $"LANGUAGE=\"en_US\"" \
--firstboot-command 'userdel -r builder &&
→ useradd -m -p "" builder ; chage -d 0
→ builder'
```

We further shrink the created VMIs to remove the allocated space not used for storage. Initially, we execute following commands on the guest OS:

```
$ dd if=/dev/zero of=/mytempfile
$ rm -f /mytempfile
```

Furthermore, to shrink the VMI disk file without compression, we use:

```
$ mv Image.qcow2 Image.qcow2_backup
$ qemu-img convert -O qcow2
→ Image.qcow2_backup Image.qcow2
```

### EXPELLIARMUS INSTALLATION

We tested the following installation steps on Ubuntu 16.04 LTS, however, Expelliarmus can be deployed on any Linux environment with manual build for libguestfs.

#### E. Requirements

- 1) Python version greater than 2.7;
- 2) Python newtorkx module;
- 3) libguestfs-tools module with a version greater than 1.36.x;
- 4) python-guestfs module.

#### F. Installation steps

```
$ mkdir Expelliarmus
$ cd Expelliarmus
$ git clone https://github.com/
→ ExpelliarmusSuperComp/Expelliarmus
$ cd ../../
$ apt install libguestfs-tools
$ apt install libguestfs-dev
$ apt install python-guestfs
$ wget "download.libguestfs.org/1.36
→ -stable/libguestfs-1.36.13.tar.gz"
$ tar -xf libguestfs-1.36.13.tar.gz
$ cd libguestfs-1.36.13
$ apt-get build-dep libguestfs
$ apt install autoconf automake
→ libtool-bin gettext
$ ./configure
$ make
$ cd ../Expelliarmus/Expelliarmus/
$ python main.py
$ Please provide path to libguestfs:
$ Expelliarmus ready to use:
```

#### G. Troubleshooting

- 1) Error: "libguestfs: error: tar\_in"  
\$ Echo dash >/user/lib  
→ /x86\_64-linux-gnu  
→ /guestfs/supermin.d  
→ /zz-dash-packages
- 2) Error: "suprmin exited with status 1"  
\$ chmod 0644 /boot/vmlinuz\*

More details on building libguestfs are available at <http://libguestfs.org/guestfs-building.1.html>.

<sup>1</sup><https://github.com/ExpelliarmusSuperComp/Expelliarmus>