9 May 2023 - CHEP

Dave Dykstra, Fermilab

Apptainer Without Setuid
A little Apptainer history

• Apptainer is the new name for the Singularity containerization project
• singularity-2 was written in C by Greg Kurtzer at LBNL in 2015, with many community contributions
  — Predated unprivileged user namespaces, but optional support for them was added when they arrived in RHEL 7
  — The High Throughput Computing (HTC) community already started using non setuid-root mode then
• Greg founded Sylabs in 2017, where singularity-3 was rewritten in Go with many new features
• In May 2020, Greg left Sylabs, taking the Singularity project leadership with him
  — Some core Singularity developers came with him to the new company he founded
  — Singularity was hosted in a new community organization, HPCng
• In May 2021, Sylabs left the project and created an open source fork, calling it SingularityCE
• In November 2021, the Singularity project joined the Linux Foundation, which required a name change, and Apptainer was chosen
  — Greg became one of 5 leaders on a Technical Steering Committee, and I am also on that committee
• Apptainer 1.0.0 released in March 2022
• Apptainer 1.1.0 released in September 2022, rootless by default
Rootless by default

• Setuid-root used to be required for mounting anything beyond bind mounts, but recent kernels allow FUSE mounts entirely in unprivileged user namespaces, without the setuid-root assistance of fusermount
  — Apptainer 1.1 added the ability to mount squashfs and ext3 filesystem types and overlayfs using FUSE, completely unprivileged

• Apptainer 1.1 has no setuid-root component in the main apptainer package
  — Can still use setuid-root, with optional apptainer-suid package or by compiling with a --with-suid option

• Apptainer normally uses only a single user id so it has no need for /etc/sub[ug]id mappings
  — Those mappings require an entry for every user, assigning them 64k otherwise unused ids
    • Can be challenging to manage on large clusters
    • Can result in odd behavior of creating files that are only accessible inside a container
  — The mappings require assistance from elevated-privilege programs, so Apptainer is more “rootless” than other container systems that call themselves rootless
Advantages and disadvantages of rootless

**Advantages of rootless:**

- Setuid-root notoriously difficult to secure
- Linux kernel reviewed by far more security experts than a single application
- The kernel can be attacked if unprivileged users can directly write filesystem data that is interpreted by kernel drivers
  - Published vulnerabilities get deemed low or medium severity because normally users can’t modify the raw data, and so the vulnerabilities don’t get patched quickly or at all on older distributions
- Being entirely unprivileged allows full nesting of containers, even though Apptainer uses NO_NEW_PRIVS

**Disadvantages of rootless:**

- FUSE performance is likely worse (but not necessarily – see next slide) than kernel filesystems
- Security fixes to unprivileged user namespaces in the kernel require reboots
  - Almost all published vulnerabilities based on unprivileged user namespaces in the last few years were in combination with network namespaces, so we recommend disabling network namespaces if possible
- Image encryption/decryption not supported in Apptainer 1.1 but is now in main branch and will be in 1.2
- Supplementary groups are not available in an unprivileged user namespace
- Some other little-used features not available without setuid-root
Unprivileged FUSE mounts

• Apptainer uses SIF files which include the container image in squashfs format, and mounts on the node
  — For HPC this is important to avoid high load on network filesystem’s metadata server when many nodes start at once
    • HTC uses sandboxes on the CernVM FileSystem (CVMFS) which also avoids this problem plus has on-demand download
  — Apptainer 1.1 supports mounting the SIF partition with squashfuse_ll or squashfuse
  — Current release of squashfuse is single threaded, but Apptainer packages contain a multi-threaded patched version
  — **Measurements** of a HEP python-based container benchmark run times on a 16-core system (lower is better):
    - sandbox on local disk: 6:21
    - sandbox on lustre: 6:32 (only one node, not parallel launches)
    - kernel squashfs, sif on lustre: 6:33
    - standard squashfuse: 41:33
    - standard squashfuse_ll: 12:48
    - multithreaded squashfuse_ll: 6:29
    - sandbox on CVMFS: 6:50 (warm cache)

• Optional overlays with ext3 filesystems or sandboxes are also supported rootless in Apptainer 1.1
  — Uses kernel overlayfs if the kernel allows it, otherwise `fuse-overlayfs`
  — ext3 filesystems are mounted with `fuse2fs`
  — Apptainer (and Singularity) does not require overlay to make missing bind points, for that it can use “underlay”
Prior to Apptainer 1.1, building containers from definition files required either root or /etc/sub[ug]id-based --fakeroot

- Building a container often requires multiple user ids for package installs, unlike a typical user application

In Apptainer 1.1, the --fakeroot option is used by default when a non-root user tries to build from a definition file and is expanded in these ways:

1. If the user is in /etc/subuid, Apptainer will use that method as before
2. Else if unprivileged user namespaces are available, it will start a root-mapped user namespace
   - Equivalent to unshare -r
3. Additionally, if the fakeroot command is available on the host, it will bind-map the command into the container and run the build commands under that
   - This command is LD_PRELOAD-based and reports success on system calls that don’t really succeed, such as chown(), and tracks the changes that would have been made in case they are requested later
   - Bind-mapping is convenient and leaves the container image without a fakeroot package but can fail if the libc of the host and the image are not compatible. The user could instead run unshare -r apptainer and in the definition file install the fakeroot package and run the rest under the fakeroot command.
   - The fakeroot-sysv command is considerably faster than and is preferred over the fakeroot command
4. If unprivileged user namespaces are not available, but setuid-root and the fakeroot command are, it will run the build only under fakeroot
Since apptainer-1.1 added dependencies of FUSE program packages, it became more of a challenge for unprivileged users to install.

Solution: install-unprivileged.sh script
- Installs apptainer and all its dependencies from rpm package files that it downloads, into path of user’s choice
- Works on RHEL-based systems, Fedora, Debian, Ubuntu, and OpenSUSE
- Works with any architecture supported by the Fedora/EPEL build system
- Required changes to apptainer to enable relocation even when compiled with the --with-suid flag, as long as the setuid-root component isn’t present

This script is now used to install x86, arm, and ppc architectures into CVMFS at /cvmfs/oasis.opensciencegrid.org/mis/apptainer/bin/apptainer
Links

- Apptainer major announcements: https://apptainer.org/news
- Squashfuse performance: https://github.com/apptainer/apptainer/issues/665
- Short paper about the 1.1 release (a draft of what will become the CHEP paper): https://arxiv.org/abs/2208.12106