Introduction

LXC containers can be of two kinds:

* **Privileged containers**
* **Unprivileged containers**

The former can be thought as old-style containers, they're not safe at all and should only be used  
in environments where **unprivileged containers** aren't available and where you would trust  
your container's user with root access to the host.

The latter has been introduced back in LXC 1.0 (February 2014) and requires a reasonably recent  
kernel (3.13 or higher). The upside being that we do consider those containers to be root-safe and so,  
as long as you keep on top of kernel **security issues**, those containers are safe.

As **privileged containers** are considered unsafe, we typically will not consider new container escape  
exploits to be **security issues** worthy of a CVE and quick fix. We will however try to mitigate those  
issues so that accidental damage to the host is prevented.

Privileged containers

**Privileged containers** are defined as any container where the container uid 0 is mapped to the host's uid 0.  
In such containers, protection of the host and prevention of escape is entirely done through  
Mandatory Access Control (apparmor, selinux), seccomp filters, dropping of capabilities and namespaces.

Those technologies combined will typically prevent any accidental damage of the host,  
where damage is defined as things like reconfiguring host hardware,  
reconfiguring the host kernel or accessing the host filesystem.

LXC upstream's position is that those containers aren't and cannot be root-safe.

They are still valuable in an environment where you are running trusted workloads  
or where no untrusted task is running as root in the container.

We are aware of a number of exploits which will let you escape such containers and get full root privileges on the host.  
Some of those exploits can be trivially blocked and so we do update our different policies once made aware of them.  
Some others aren't blockable as they would require blocking so many core features that the average container would become completely unusable.

Unprivileged containers

**Unprivileged containers** are safe by design. The container uid 0 is mapped to an unprivileged user outside of the container  
and only has extra rights on resources that it owns itself.

With such container, the use of SELinux, AppArmor, Seccomp and capabilities isn't necessary for security.  
LXC will still use those to add an extra layer of security which may be handy in the event  
of a kernel security issue but the security model isn't enforced by them.

To make **unprivileged containers** work, LXC interacts with 3 pieces of setuid code:

* lxc-user-nic (setuid helper to create a veth pair and bridge it on the host)
* newuidmap (from the shadow package, sets up a uid map)
* newgidmap (from the shadow package, sets up a gid map)

Everything else is run as your own user or as a uid which your user owns.

As a result, most **security issues** (container escape, resource abuse, ...) in those containers will apply just as well  
to a random unprivileged user and so would be a generic kernel security bug rather than a LXC issue.

LXC upstream is happy to help track such security issue and get in touch with the Linux kernel community  
to have them resolved as quickly as possible.

Reporting security issues

To ensure **security issues** can be fixed as quickly as possible and simultaneously  
in all Linux distributions, issues should be reported either:

* By e-mail to both serge.hallyn (at) ubuntu (dot) com AND stgraber (at) ubuntu (dot) com
* By opening a private security bug at <https://launchpad.net/ubuntu/+source/lxc/+filebug>

We will then confirm the security issue, come up with fixes against all supported releases,  
provide you those patches for testing and then get a CVE assigned as well as a  
coordinated release date for you and the Linux distribution community.