

# Skyper's blog

Personal blog about IT,  
Electronics, InfoSec,  
Hacking, Bug Hunting...



Home

Publications

About me

Contact

Support me

## Detect whether you are inside a container or not

27 Jul 2023

Container technologies ([chroot](#), [LXC](#), ...) are very common these days, especially since the massive adoption of [Docker](#).

One of the use cases of container technologies is to isolate services from each others and from the host system. As a result, in case of an intrusion, the attacker would be in theory trapped inside a container. From the attacker's perspective, it is important to be able to detect if a compromised service lives in a restricted environment such as a Docker container or if it runs directly on the host operating system.

One way to do so is to have a look at the [inode](#) of the `/` mount point (`ls -ld /`). On the host system it will be

# Skyper's blog

Personal blog about IT,  
Electronics, InfoSec,  
Hacking, Bug Hunting...



Home

Publications

About me

Contact

Support me

very low (generally 1 or 2) whereas in a container it will generally be quite high (4851522 in the asciicast):

```
[skyper@desktop0:~]$ ls -id /
2 /
[skyper@desktop0:~]$ docker run --rm -it ubuntu bash
root@539b2bf40416:/# ls -id /
4851522 /
root@539b2bf40416:/#
```

Recorded with **asciinema**

On Linux, one of the underlying mechanisms commonly used to create a container is **cgroups**. The `/proc/1/cgroup` virtual file will give you the control groups of the `init` process which are generally `/` for the majority of the controllers by default. However, if you have a look at `/proc/1/cgroup` from the inside of a container, the result is likely to be different as you can see:

# Skyper's blog

Personal blog about IT,  
Electronics, InfoSec,  
Hacking, Bug Hunting...



[Home](#)

[Publications](#)

[About me](#)

[Contact](#)

[Support me](#)

```
[skyper@desktop0:~]$ cat /proc/1/cgroup
11:rdma:/
10:cpu,cpuacct:/
9:pids:/
8:freezer:/
7:memory:/
6:net_cls,net_prio:/
5:devices:/
4:perf_event:/
3:cpuset:/
```

When containers are created by a Docker Engine, this last one adds a `/.dockerenv` file into them. The presence of this file is even [used to this date by some underlying components of the Moby project](#) for the exact same purpose, knowing if they run inside a container:

```
docker run --rm -it alpine:3
/ # ls -al /
total 64
drwxr-xr-x  1 root  root    4096 Apr
drwxr-xr-x  1 root  root    4096 Apr
-rwxr-xr-x  1 root  root      0 Apr
drwxr-xr-x  2 root  root    4096 Mar
drwxr-xr-x  5 root  root    340 Apr
```

📁 [Security](#) · [SysAdmin](#)

📁 [Container](#) · [Docker](#)