|  |  |  |  |
| --- | --- | --- | --- |
| My team has been getting pretty excited about using docker because it promises to simplify our deployments and provide a number of other design and operational benefits. We recently started to really get things rolling with it and ran into some issues with the fact that the docker daemon runs as root.  In general, my stance on running server platforms as root is "don't". We recently went through a bunch of battles to get our operators to stop doing this and to even stop running things under accounts that can modify the server deployment. So right off the bat, I have problem here that it seems a little hypocritical to go back to these same people and ask them to set up docker to run as root.  I'm not the first to comment on [risks](http://reventlov.com/advisories/using-the-docker-command-to-root-the-host) posed by the docker daemon running as root. According to [this](https://docs.docker.com/articles/security/)"Eventually, it is expected that the Docker daemon will run restricted privileges, ..." Should we just wait for this to be addressed? I thought docker would improve our security profile but this seems to make it worse. My enthusiasm has been deflated for docker and I'm not sure I am motivated to make a case for using it in this state to our risk team.  EDIT: I should clarify that I'm not specifically concerned here with the issues around users being members of the "docker" group. It's important to know about but that can be managed. I really appreciate all the great answers here (both pro and con). I think I've been conflating the daemon with the containers themselves. I probably need to work on a clarified mental model of the docker architecture. Again lots of great stuff here all around. I'll have to ruminate a bit before accepting an answer.  Am I wrong to think this is a major flaw with docker?  [docker](http://security.stackexchange.com/questions/tagged/docker)   |  |  |  | | --- | --- | --- | | [share](http://security.stackexchange.com/q/102323)[improve this question](http://security.stackexchange.com/posts/102323/edit) | [edited Oct 12 '15 at 19:28](http://security.stackexchange.com/posts/102323/revisions) | asked Oct 9 '15 at 20:18  [[https://www.gravatar.com/avatar/ac23cd4b72c5ecae7a00dcd7d7e12001?s=32&d=identicon&r=PG](http://security.stackexchange.com/users/88460/jimmyjames)](http://security.stackexchange.com/users/88460/jimmyjames)  [JimmyJames](http://security.stackexchange.com/users/88460/jimmyjames)  **388**6 | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | I think that you're overly paranoid. The docker daemon cannot be accessed remotely unless you explicitly enable that. Otherwise it can only be accessed by first rooting the physical machine, at which point containers are at stake anyway (you have bigger things to worry about then). From the point of view of a container, it doesn't matter whether the docker daemon runs as root, first of all they are isolated (as if a different machine), and also the kernel always "runs as root", too. – [Damon](http://security.stackexchange.com/users/39774/damon) [Oct 9 '15 at 22:13](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root#comment177508_102323) | | |  |  | | --- | --- | |  |  | | In addition to what Damon wrote, there should be good defense in depth in practice. If the security of the entire system will crumble just because the Docker daemon is running as root and successfully exploited, then there are more important things to focus on. I don't think there is a yes/no answer to this question but I would suggest putting together a threat model if you haven't already to understand what is the risk of Docker running as root in your environment. Not to sound cliche but if you can achieve a huge win for your business and this risk is low for your environment, roll with it. – [VirtualJJ](http://security.stackexchange.com/users/79331/virtualjj) [Oct 10 '15 at 4:15](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root#comment177528_102323) | | |  |  | | --- | --- | |  |  | | I'm not sure I understand your statement "If the security of the entire system will crumble just because [X] is running as root and successfully exploited." When an application that is running as root is exploited, the chance that exploiter can take control of the system is increased. I don't think this is something to take lightly.– [JimmyJames](http://security.stackexchange.com/users/88460/jimmyjames) [Oct 19 '15 at 15:17](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root#comment179283_102323) |   add a comment |

3 Answers

[active](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root?answertab=active#tab-top)[oldest](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root?answertab=oldest#tab-top)[**votes**](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root?answertab=votes#tab-top)

|  |  |  |  |
| --- | --- | --- | --- |
| up vote5down voteaccepted | The docker daemon does run as root, as it interfaces with the host operating system in a farily fundamental manner, however that's no different than most/any system daemon that makes use of linux capabilities which require that privilege.  This doesn't mean that using docker is insecure, just that you need to be careful with how you use it. Luckily there is already some fairly good security configuration advice available on this from[Docker themselves](https://docs.docker.com/articles/security/) and in the form of a security configuration "best practices" guide from the[Center for Internet Security](https://benchmarks.cisecurity.org/tools2/docker/CIS_Docker_1.6_Benchmark_v1.0.0.pdf)  A couple of practical considerations which you do need to be aware of when running docker.  First up is that if someone is a member of the "docker" group on the host, they're effectively root on the system as it's possible to use docker to escalate privileges in this case. So you need to treat membership of that group on production hosts with care.  Docker uses Linux capabilities to restrict what actions a user inside a container can take, so just being root inside a container doesn't necessarily mean automatic root on the host. That said you do need to be careful with things like volume mounts (so if you mount a system directory from the host into a container for example) as this can allow a root user inside a container to make changes to files on the host.  You can also reduce the capabilities provided to a container fairly easily, which can improve the security of the process further.  Also remember you don't need to run processes in containers as root, you can run as other users with a bit of configuration, and this mitigates the volume mounting risk.  I find that the way to think about containers is as processes on the host. If you compare running things in containers to running them directly on the host, I'd say that the container architecture is likely to add security rather than remove it, as you've got a better defined interface between the process and the host and more control over what the process can do on the host.  Also it's worth nothing that a richer user mapping architecture is coming fairly soon to docker ([planned for 1.9](https://github.com/docker/docker/wiki/Engine-1.9.0)) which will add some more options for restricting actions in a container.   |  |  | | --- | --- | | [share](http://security.stackexchange.com/a/102350)[improve this answer](http://security.stackexchange.com/posts/102350/edit) | answered Oct 10 '15 at 11:35  [[https://www.gravatar.com/avatar/85fb3963936c95b4cb719f10f8e3ce7c?s=32&d=identicon&r=PG](http://security.stackexchange.com/users/37/r%d0%bery-mccune)](http://security.stackexchange.com/users/37/r%d0%bery-mccune)  [Rоry McCune](http://security.stackexchange.com/users/37/r%d0%bery-mccune)  **36.4k**766139 | |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | "I find that the way to think about containers is as processes on the host. If you compare running things in containers to running them directly on the host, I'd say that the container architecture is likely to add security rather than remove it..." That's a really good way of putting it. I might add that while containers might not have quite the security separation from a host as a VM does, one can always run containers inside a VM for scenarios that require stronger security promises. The Docker documentation describes a basic Virtualbox set-up: [docs.docker.com/machine/get-started](https://docs.docker.com/machine/get-started/) – [halfinformed](http://security.stackexchange.com/users/86410/halfinformed) [Oct 10 '15 at 18:57](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root#comment177586_102350) | | |  |  | | --- | --- | | 1 |  | | "*however that's no different than most/any system daemon that makes use of linux capabilities which require that privilege*". It does not *require* it. It is the very same thing I reproach to libvirt and Docker: Linux makes things to nicely meet the least privilege axiom: one does not need root privileges to run a virtual machine or a sandboxed environment. However, for historical reasons and because it makes things easier for developers and end-user by avoiding any privilege issue, these layers requires root privileges on the host system. Convenient? Yes. Secure? No. – [WhiteWinterWolf](http://security.stackexchange.com/users/32746/whitewinterwolf) [Oct 12 '15 at 7:29](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root#comment177757_102350) | | |  |  | | --- | --- | |  |  | | @WhiteWinterWolf to clarify are you thinking that the docker daemon shouldn't need root or that containers that it instantiates shouldn't? My thinking is that the docker daemon does as it'll set up things like new bridges/IPtables rules etc to set-up a container which I would think of as privileged operations. Containers running as root is a different prospect and I'd agree (as would docker as they're implementing user namespacing) – [Rоry McCune](http://security.stackexchange.com/users/37/r%d0%bery-mccune) [Oct 12 '15 at 20:19](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root#comment177917_102350) | | |  |  | | --- | --- | | 1 |  | | I hope for Docker that compatibility with user namespacing will be really available for 1.9 since, as I said in my answer, this feature has been postponed several times since it becomes available in LXC. Qemu, for instance, provides a minimal very small SUID utility (runnable only by Qemu group members) which transparently takes in charge bridges creation/deletion. Apart from that all the rest is running as end-user, which reduces the attack surface compared to a root daemon. Maybe in a future version Docker will be able to completely drop root privileges, but I fear this may not be soon. – [WhiteWinterWolf](http://security.stackexchange.com/users/32746/whitewinterwolf) [Oct 13 '15 at 9:53](http://security.stackexchange.com/questions/102323/risks-posed-by-docker-daemon-running-as-root#comment178045_102350) | | |  |  | | --- | --- | |  |  | | @WhiteWinterWolf it got committed to master over the weekend and as that's before the 1.9 feature freeze date yesterday, it looks probable to have made the cut for this version (absent problems in test I'd guess) (The commit is here [github.com/docker/docker/commit/…](https://github.com/docker/docker/commit/ed9434c5bb64f49db442027e5e748bfbb46d0a0a)) and the pull request is linked from that. – [Rоry McCune](http://security.stackexchange.com/users/37/r%d0%bery-mccune) | |