Bugs mitigated:

* [CVE-2013-1956](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1956), [1957](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1957), [1958](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1958), [1959](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1959), [1979](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1979), [CVE-2014-4014](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-4014), [5206](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-5206), [5207](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-5207), [7970](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7970), [7975](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7975), [CVE-2015-2925](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2925), [8543](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-8543), [CVE-2016-3134](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-3134), [3135](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-3135), etc.: The introduction of unprivileged user namespaces lead to a huge increase in the attack surface available to unprivileged users by giving such users legitimate access to previously root-only system calls like mount(). All of these CVEs are examples of security vulnerabilities due to introduction of user namespaces. Docker can use user namespaces to set up containers, but then disallows the process inside the container from creating its own nested namespaces through the default seccomp profile, rendering these vulnerabilities unexploitable.
* [CVE-2014-0181](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0181), [CVE-2015-3339](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3339): These are bugs that require the presence of a setuid binary. Docker disables setuid binaries inside containers via the NO\_NEW\_PRIVS process flag and other mechanisms.
* [CVE-2014-4699](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-4699): A bug in ptrace() could allow privilege escalation. Docker disables ptrace() inside the container using apparmor, seccomp and by dropping CAP\_PTRACE. Three times the layers of protection there!
* [CVE-2014-9529](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-9529): A series of crafted keyctl() calls could cause kernel DoS / memory corruption. Docker disables keyctl() inside containers using seccomp.
* [CVE-2015-3214](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3214), [4036](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4036): These are bugs in common virtualization drivers which could allow a guest OS user to execute code on the host OS. Exploiting them requires access to virtualization devices in the guest. Docker hides direct access to these devices when run without --privileged. Interestingly, these seem to be cases where containers are “more secure” than a VM, going against common wisdom that VMs are “more secure” than containers.
* [CVE-2016-0728](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-0728): Use-after-free caused by crafted keyctl() calls could lead to privilege escalation. Docker disables keyctl() inside containers using the default seccomp profile.
* [CVE-2016-2383](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-2383): A bug in eBPF – the special in-kernel DSL used to express things like seccomp filters – allowed arbitrary reads of kernel memory. The bpf() system call is blocked inside Docker containers using (ironically) seccomp.
* [CVE-2016-3134](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-3134), [4997](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-4997), [4998](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-4998): A bug in setsockopt with IPT\_SO\_SET\_REPLACE, ARPT\_SO\_SET\_REPLACE, and ARPT\_SO\_SET\_REPLACE causing memory corruption / local privilege escalation. These arguments are blocked by CAP\_NET\_ADMIN, which Docker does not allow by default.

Bugs not mitigated:

* [CVE-2015-3290](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3290), [5157](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5157): Bugs in the kernel’s non-maskable interrupt handling allowed privilege escalation. Can be exploited in Docker containers because the modify\_ldt()system call is not currently blocked using seccomp.
* [CVE-2016-5195](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-5195): A race condition was found in the way the Linux kernel’s memory subsystem handled the copy-on-write (COW) breakage of private read-only memory mappings, which allowed unprivileged local users to gain write access to read-only memory. Also known as “dirty COW.” Partial mitigations: on some operating systems this vulnerability is mitigated by the combination of seccomp filtering of ptrace and the fact that /proc/self/mem is read-only.