

# Kernel Configuration

Use the following configuration settings as a base for an Android kernel configuration. Settings are organized into **android-base** and **android-recommended** .cfg files:

- **android-base**. These options enable core Android features and should be enabled by all devices.
- **android-recommended**. These options enable advanced Android features and are optional for devices.

Both the android-base.cfg and android-recommended.cfg files are located in the android-common kernel repo at <https://android.googlesource.com/kernel/common/> (<https://android.googlesource.com/kernel/common/>).

In version 4.8 of the upstream Linux kernel, a new location (kernel/configs) was designated for kernel configuration fragments. The android base and recommended config fragments are located in that directory for branches based on 4.8 or later. For kernel branches based on releases prior to 4.8, the config fragments are located in the android/ directory.

For details on controls already undertaken to strengthen the kernel on your devices, see [System and Kernel Security](https://source.android.com/security/overview/kernel-security.html) (<https://source.android.com/security/overview/kernel-security.html>). For details on required settings, see the [Android Compatibility Definition Document \(CDD\)](https://source.android.com/compatibility/cdd.html) (<https://source.android.com/compatibility/cdd.html>).

## Generating kernel config

For devices that have a minimalist defconfig, you can use the following to enable options:

```
ARCH=arch scripts/kconfig/merge_config.sh path/device_defconfig android/configs/android-base.cfg android/configs/andr
```

This generates a .config file you can use to save a new defconfig or compile a new kernel with Android features enabled.

## Enabling USB host mode options

For USB host mode audio, enable the following options:

```
CONFIG_SND_USB=y
CONFIG_SND_USB_AUDIO=y
# CONFIG_USB_AUDIO is for a peripheral mode (gadget) driver
```

For USB host mode MIDI, enable the following option:

```
CONFIG_SND_USB_MIDI=y
```

## Seccomp-BPF with TSYNC

Seccomp-BPF is a kernel security technology that enables the creation of sandboxes to restrict the system calls a process is allowed to make. The TSYNC feature enables the use of Seccomp-BPF from multithreaded programs. This ability is limited to architectures that have seccomp support upstream: ARM, ARM64, x86, and x86\_64.

### Backporting for Kernel 3.10 for ARM-32, X86, X86\_64

Ensure that CONFIG\_SECCOMP\_FILTER=y is enabled in the Kconfig (verified as of the Android 5.0 CTS), then cherry-pick the following changes from the AOSP kernel/common:android-3.10 repository:  
[9499cd23f9d05ba159fac6d55dc35a7f49f9ce76..a9ba4285aa5722a3b4d84888e78ba8adc0046b28](https://android.googlesource.com/kernel/common/+log/9499cd23f9d05ba159fac6d55dc35a7f49f9ce76..a9ba4285aa5722a3b4d84888e78ba8adc0046b28)  
(<https://android.googlesource.com/kernel/common/+log/9499cd23f9d05ba159fac6d55dc35a7f49f9ce76..a9ba4285aa5722a3b4d84888e78ba8adc0046b28>)

- [a03 a242 arch: Introduce smp\\_load\\_acquire\(\), smp\\_store\\_release\(\)](https://android.googlesource.com/kernel/common/+a03a2426ea9f1d9dada33cf4a824f63e8f916c9d)  
(<https://android.googlesource.com/kernel/common/+a03a2426ea9f1d9dada33cf4a824f63e8f916c9d>) by Peter Zijlstra
- [987a0f 1 introduce for\\_each\\_thread\(\) to replace the buggy while\\_each\\_thread\(\)](https://android.googlesource.com/kernel/common/+987a0f1102321853565c4bfecde6a5a58ac6db11)  
(<https://android.googlesource.com/kernel/common/+987a0f1102321853565c4bfecde6a5a58ac6db11>) by Oleg Nesterov
- [2a30a43 seccomp: create internal mode-setting function](#)

- (<https://android.googlesource.com/kernel/common/+ /2a30a4386e4a7e1283157c4cf4cfcc0306b22ac8>) by Kees Cook
- [b8a9cff seccomp: extract check/assign mode helpers](#)  
(<https://android.googlesource.com/kernel/common/+ /b8a9cff6dbe9cfddb4d17e2dea496e523544687>) by Kees Cook
- [8908dde seccomp: split mode setting routines](#)  
(<https://android.googlesource.com/kernel/common/+ /8908dde5a7fdca974374b0dbe6dfb10f69df7216>) by Kees Cook
- [e985fd4 seccomp: add "seccomp" syscall](#)  
(<https://android.googlesource.com/kernel/common/+ /e985fd474debedb269fba27006eda50d0b6f07ef>) by Kees Cook
- [9d0ff69 sched: move no\\_new\\_privs into new atomic flags](#)  
(<https://android.googlesource.com/kernel/common/+ /9d0ff694bc22fb458acb763811a677696c60725b>) by Kees Cook
- [b6a12bf seccomp: split filter prep from check and apply](#)  
(<https://android.googlesource.com/kernel/common/+ /b6a12bf4dd762236c7f637b19cfe10a268304b9b>) by Kees Cook
- [61b6b88 seccomp: introduce writer locking](#)  
(<https://android.googlesource.com/kernel/common/+ /61b6b882a0abfeb627d25a069cfa1d232b84c8eb>) by Kees Cook
- [c852ef7 seccomp: allow mode setting across threads](#)  
(<https://android.googlesource.com/kernel/common/+ /c852ef778224ecf5fe995d74ad96087038778bca>) by Kees Cook
- [f14a5db seccomp: implement SECCOMP\\_FILTER\\_FLAG\\_TSYNC](#)  
(<https://android.googlesource.com/kernel/common/+ /f14a5db2398afed8f416d244e6da6b23940997c6>) by Kees Cook
- [9ac8600 seccomp: Replace BUG\(!spin\\_is\\_locked\(\)\) with assert\\_spin\\_lock](#)  
(<https://android.googlesource.com/kernel/common/+ /9ac860041db860a59bfd6ac82b31d6b6f76ebb52>) by Guenter Roeck
- [900e9fd seccomp: fix syscall numbers for x86 and x86\\_64](#)  
(<https://android.googlesource.com/kernel/common/+ /900e9fd0d5d15c596cacfb89ce007c933cea6e1c>) by Lee Campbell
- [a9ba428 ARM: add seccomp syscall](#) (<https://android.googlesource.com/kernel/common/+ /a9ba4285aa5722a3b4d84888e78ba8adc0046b28>) by Kees Cook

## Backporting for Kernel 3.10 for ARM-64

Ensure `CONFIG_SECCOMP_FILTER=y` is enabled in the Kconfig (verified as of the Android 5.0 CTS), then cherry-pick the following changes from the AOSP kernel/common:android-3.10 repository:

- [cfc7e99e9 arm64: Add \\_\\_NR\\_\\* definitions for compat syscalls](#)  
(<https://android.googlesource.com/kernel/common/+ /cfc7e99e9e3900056028a7d90072e9ea0d886f8d>) by JP Abgrall
- [bf11863 arm64: Add audit support](#) (<https://android.googlesource.com/kernel/common/+ /bf11863d45eb3dac0d0cf1f818ded11ade6e28d3>) by AKASHI Takahiro
- [3e21c0b arm64: audit: Add audit hook in syscall\\_trace\\_enter/exit\(\)](#)  
(<https://android.googlesource.com/kernel/common/+ /3e21c0bb663a23436e0eb3f61860d4fedc233bab>) by JP Abgrall
- [9499cd2 syscall\\_get\\_arch: remove useless function arguments](#)  
(<https://android.googlesource.com/kernel/common/+ /9499cd23f9d05ba159fac6d55dc35a7f49f9ce76>) by Eric Paris
- [2a30a43 seccomp: create internal mode-setting function](#)  
(<https://android.googlesource.com/kernel/common/+ /2a30a4386e4a7e1283157c4cf4cfcc0306b22ac8>) by Kees Cook
- [b8a9 cff seccomp: extract check/assign mode helpers](#)  
(<https://android.googlesource.com/kernel/common/+ /b8a9cff6dbe9cfddb4d17e2dea496e523544687>) by Kees Cook
- [8908dde seccomp: split mode setting routines](#)  
(<https://android.googlesource.com/kernel/common/+ /8908dde5a7fdca974374b0dbe6dfb10f69df7216>) by Kees Cook
- [e985fd4 seccomp: add "seccomp" syscall](#)  
(<https://android.googlesource.com/kernel/common/+ /e985fd474debedb269fba27006eda50d0b6f07ef>) by Kees Cook
- [9d0ff69 sched: move no\\_new\\_privs into new atomic flags](#)  
(<https://android.googlesource.com/kernel/common/+ /9d0ff694bc22fb458acb763811a677696c60725b>) by Kees Cook
- [b6a12bf seccomp: split filter prep from check and apply](#)  
(<https://android.googlesource.com/kernel/common/+ /b6a12bf4dd762236c7f637b19cfe10a268304b9b>) by Kees Cook
- [61b6b88 seccomp: introduce writer locking](#)  
(<https://android.googlesource.com/kernel/common/+ /61b6b882a0abfeb627d25a069cfa1d232b84c8eb>) by Kees Cook
- [c852ef7 seccomp: allow mode setting across threads](#)  
(<https://android.googlesource.com/kernel/common/+ /c852ef778224ecf5fe995d74ad96087038778bca>) by Kees Cook

- [f14a5db seccomp: implement SECCOMP\\_FILTER\\_FLAG\\_TSYNC](#)  
(<https://android.googlesource.com/kernel/common/+f14a5db2398afed8f416d244e6da6b23940997c6>) by Kees Cook
- [9ac8600 seccomp: Replace BUG\(!spin\\_is\\_locked\(\)\) with assert\\_spin\\_lock](#)  
(<https://android.googlesource.com/kernel/common/+9ac860041db860a59bfd6ac82b31d6b6f76ebb52>) by Guenter Roeck
- [900e9fd seccomp: fix syscall numbers for x86 and x86\\_64](#)  
(<https://android.googlesource.com/kernel/common/+900e9fd0d5d15c596cacfb89ce007c933cea6e1c>) by Lee Campbell
- [a9ba428 ARM: add seccomp syscall](#) (<https://android.googlesource.com/kernel/common/+a9ba4285aa5722a3b4d84888e78ba8adc0046b28>) by Kees Cook
- [4190090 ARM: 8087/1: ptrace: reload syscall number after secure\\_computing\(\) check](#)  
(<https://android.googlesource.com/kernel/common/+41900903483eb96602dd72e719a798c208118aad>) by Will Deacon
- [abbfed9 arm64: ptrace: add PTRACE\\_SET\\_SYSCALL](#)  
(<https://android.googlesource.com/kernel/common/+abbfed9ed1a78701ef3db74f5287958feb897035>) by AKASHI Takahiro
- [feb2843 arm64: ptrace: allow tracer to skip a system call](#)  
(<https://android.googlesource.com/kernel/common/+feb28436457d33fef9f264635291432df4b74122>) by AKASHI Takahiro
- [dab1073 asm-generic: add generic seccomp.h for secure computing mode 1](#)  
(<https://android.googlesource.com/kernel/common/+dab10731da65a0deba46402ca9fadf6974676cc8>) by AKASHI Takahiro
- [4f1 2b53 add seccomp syscall for compat task](#)  
(<https://android.googlesource.com/kernel/common/+4f12b53f28a751406a27ef7501a22f9e32a9c30b>) by AKASHI Takahiro
- [7722723 arm64: add SIGSYS signinfo for compat task](#)  
(<https://android.googlesource.com/kernel/common/+77227239d20ac6381fb1aee7b7cc902f0d14cd85>) by AKASHI Takahiro
- [210957c arm64: add seccomp support](#) (<https://android.googlesource.com/kernel/common/+210957c2bb3b4d111963bb296e2c42beb8721929>) by AKASHI Takahiro

---

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated August 8, 2017.