Recommended Settings

Sometimes people ask the Kernel Self Protection Project what a secure set of build CONFIGs and runtime settings are. This is a brain-dump of the various options for a particularly paranoid system.

Another place to find recommended kernel hardening settings is via the “kernel-hardening-checker” tool maintained by Alexander Popov.

CONFIGs

```
# Report BUG() conditions and kill the offending process.
CONFIG_BUG=y

# Make sure kernel page tables have safe permissions.
CONFIG_DEBUG_KERNEL=y (prior to v4.11, needed to select CONF
CONFIG_DEBUG_RODATA=y (prior to v4.11)
CONFIG_STRICT_KERNEL_RWX=y (since v4.11)

# Report any dangerous memory permissions (not available on
CONFIG_DEBUG_WX=y

# Use -fstack-protector-strong (gcc 4.9+) for best stack car
# Prior to v4.18, these are:
#  CONFIG_CC_STACKPROTECTOR=y
#  CONFIG_CC_STACKPROTECTOR_STRONG=y
CONFIG_STACKPROTECTOR=y
CONFIG_STACKPROTECTOR_STRONG=y

# Do not allow direct physical memory access (but if you mus
# CONFIG_DEVMEM is not set
CONFIG_STRICT_DEVMEM=y
CONFIG_IO_STRICT_DEVMEM=y

# Provides some protections against SYN flooding.
CONFIG_SYN_COOKIES=y

# Perform additional validation of various commonly targetec
CONFIG_LIST_HARDENED=y
CONFIG_DEBUG_CREDENTIALS=y
CONFIG_DEBUG_NOTIFIERS=y
CONFIG_DEBUG_LIST=y
CONFIG_DEBUG_SG=y
CONFIG_DEBUG_VIRTUAL=y
CONFIG_BUG_ON_DATA_CORRUPTION=y
CONFIG_SCHED_STACK_END_CHECK=y

# Provide userspace with seccomp BPF API for syscall attack
CONFIG_SECCOMP=y
CONFIG_SECCOMP_FILTER=y

# Make sure line disciplines can't be autoloaded (since v5.1
# CONFIG_LDISC_AUTOLOAD is not set

# Provide userspace with ptrace ancestry protections.
# Make sure that "yama" is also present in the "CONFIG_LSM=y
CONFIG_SECURITY=y
CONFIG_SECURITY_YAMA=y

# Provide userspace with Landlock MAC interface.
# Make sure that "landlock" is also present in the "CONFIG_L
CONFIG_SECURITY_LANDLOCK=y

# Make sure SELinux cannot be disabled trivially.
# CONFIG_SECURITY_SELINUX_BOOTPARAM is not set
# CONFIG_SECURITY_SELINUX_DEVELOP is not set
# CONFIG_SECURITY_SELINUX_DEBUG is not set
# CONFIG_SECURITY_WRITABLE_HOOKS is not set

# Enable "lockdown" LSM for bright line between the root use
CONFIG_SECURITY_LOCKDOWN_LSM=y
CONFIG_SECURITY_LOCKDOWN_LSM_EARLY=y
CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY=y

# Perform usercopy bounds checking. (And disable fallback to
CONFIG_HARDENED_USERCOPY=y
```

```
# CONFIG_HARDENED_USERCOPY_FALLBACK is not set
# CONFIG_HARDENED_USERCOPY_PAGESPAN is not set

# Randomize allocator freelists, harden metadata.
CONFIG_SLAB_FREELIST_RANDOM=y
CONFIG_SLAB_FREELIST_HARDENED=y
CONFIG_RANDOM_KMALLOC_CACHES=y

# Make cross-slab heap attacks not as trivial when object si
# CONFIG_SLAB_MERGE_DEFAULT is not set

# Allow for randomization of high-order page allocation free
# the "page_alloc.shuffle=1" command line below).
CONFIG_SHUFFLE_PAGE_ALLOCATOR=y

# Sanity check userspace page table mappings (since v5.17)
CONFIG_PAGE_TABLE_CHECK=y
CONFIG_PAGE_TABLE_CHECK_ENFORCED=y

# Allow allocator validation checking to be enabled (see "sl
CONFIG_SLUB_DEBUG=y

# Wipe higher-level memory allocations when they are freed (
# This kernel feature was removed in v5.11.
# Starting from v5.11 CONFIG_PAGE_POISONING unconditionally 
CONFIG_PAGE_POISONING_ZERO=y

# Wipe slab and page allocations (since v5.3)
# Instead of "slub_debug=P" and "page_poison=1", a single pl
# The init_on_free is only needed if there is concern about 
CONFIG_INIT_ON_ALLOC_DEFAULT_ON=y
CONFIG_INIT_ON_FREE_DEFAULT_ON=y

# Initialize all stack variables on function entry. (Clang a
CONFIG_INIT_STACK_ALL_ZERO=y

# Adds guard pages to kernel stacks (not all architectures s
CONFIG_VMAP_STACK=y

# Perform extensive checks on reference counting.
CONFIG_REFCOUNT_FULL=y

# Check for memory copies that might overflow a structure ir
CONFIG_FORTIFY_SOURCE=y

# Avoid kernel memory address exposures via dmesg (sets sysc
CONFIG_SECURITY_DMESG_RESTRICT=y

# Enable trapping bounds checking of array indexes (since v5
CONFIG_UBSAN=y
CONFIG_UBSAN_TRAP=y
CONFIG_UBSAN_BOUNDS=y
CONFIG_UBSAN_SANITIZE_ALL=y
# CONFIG_UBSAN_SHIFT is not set
# CONFIG_UBSAN_DIV_ZERO is not set
# CONFIG_UBSAN_UNREACHABLE is not set
# CONFIG_UBSAN_SIGNED_WRAP is not set
# CONFIG_UBSAN_BOOL is not set
# CONFIG_UBSAN_ENUM is not set
# CONFIG_UBSAN_ALIGNMENT is not set
# This is only available on Clang builds, and is likely alre
CONFIG_UBSAN_LOCAL_BOUNDS=y

# Enable sampling-based overflow detection (since v5.12). Th
CONFIG_KFENCE=y
CONFIG_KFENCE_SAMPLE_INTERVAL=100

# Randomize kernel stack offset on syscall entry (since v5.1
CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT=y

# Do not ignore compile-time warnings (since v5.15)
CONFIG_WERROR=y

# Disable DMA between EFI hand-off and the kernel's IOMMU se
CONFIG_EFI_DISABLE_PCI_DMA=y

# Force IOMMU TLB invalidation so devices will never be able
CONFIG_IOMMU_SUPPORT=y
CONFIG_IOMMU_DEFAULT_DMA_STRICT=y
# CONFIG_IOMMU_DEFAULT_PASSTHROUGH is not set

# Enable feeding RNG entropy from TPM, if available.
CONFIG_HW_RANDOM_TPM=y

# Get as much entropy as possible from external sources. The
# malicious sources should not cause problems.
CONFIG_RANDOM_TRUST_BOOTLOADER=y
CONFIG_RANDOM_TRUST_CPU=y

# Randomize the layout of system structures. This may have c
# use with caution. If using GCC, you can check if using CON
CONFIG_RANDSTRUCT_FULL=y
```

```
# Make scheduler aware of SMT Cores. Program needs to opt-ir
CONFIG_SCHED_CORE=y

# Wipe all caller-used registers on exit from the function (
# minimizes stale data in registers). (Since v5.15)
CONFIG_ZERO_CALL_USED_REGS=y

# Wipe RAM at reboot via EFI.
# For more details, see:
# https://trustedcomputinggroup.org/resource/pc-client-work-
# https://bugzilla.redhat.com/show_bug.cgi?id=1532058
CONFIG_RESET_ATTACK_MITIGATION=y

# This needs userspace support, and will break "regular" dis
CONFIG_STATIC_USERMODEHELPER=y

# Dangerous; enabling this allows direct physical memory wri
# CONFIG_ACPI_CUSTOM_METHOD is not set

# Dangerous; enabling this disables brk ASLR.
# CONFIG_COMPAT_BRK is not set

# Dangerous; enabling this allows direct kernel memory writi
# CONFIG_DEVKMEM is not set

# Dangerous; exposes kernel text image layout.
# CONFIG_PROC_KCORE is not set

# Dangerous; enabling this allows replacement of running ker
# CONFIG_KEXEC is not set

# Dangerous; enabling this allows replacement of running ker
# CONFIG_HIBERNATION is not set

# Prior to v4.1, assists heap memory attacks; best to keep i
# CONFIG_INET_DIAG is not set

# Easily confused by misconfigured userspace, keep off.
# CONFIG_BINFMT_MISC is not set

# Use the modern PTY interface (devpts) only.
# CONFIG_LEGACY_PTYS is not set

# Block TTY stuffing attacks (this will break screen readers
# CONFIG_LEGACY_TIOCSTI is not set
# If SELinux can be disabled at runtime, the LSM structures
# CONFIG_SECURITY_SELINUX_DISABLE is not set

# Reboot devices immediately if kernel experiences an Oops.
CONFIG_PANIC_ON_OOPS=y
CONFIG_PANIC_TIMEOUT=-1

# Limit sysrq to sync,unmount,reboot. For more details see t
# https://docs.kernel.org/admin-guide/sysrq.html
CONFIG_MAGIC_SYSRQ_DEFAULT_ENABLE=176

# Keep root from altering kernel memory via loadable modules
# CONFIG_MODULES is not set

# But if CONFIG_MODULE=y is needed, at least they must be si
# See also kernel.modules_disabled sysctl below.
CONFIG_DEBUG_SET_MODULE_RONX=y (prior to v4.11)
CONFIG_STRICT_MODULE_RWX=y (since v4.11)
CONFIG_MODULE_SIG=y
CONFIG_MODULE_SIG_FORCE=y
CONFIG_MODULE_SIG_ALL=y
CONFIG_MODULE_SIG_SHA512=y
CONFIG_MODULE_SIG_HASH="sha512"
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
# CONFIG_MODULE_FORCE_LOAD is not set
```

GCC plugins

```
# Enable GCC Plugins
CONFIG_GCC_PLUGINS=y

# Gather additional entropy at boot time for systems that ma
CONFIG_GCC_PLUGIN_LATENT_ENTROPY=y

# Force all structures to be initialized before they are pas
# When building with GCC:
CONFIG_GCC_PLUGIN_STRUCTLEAK=y
CONFIG_GCC_PLUGIN_STRUCTLEAK_BYREF_ALL=y

# Wipe stack contents on syscall exit (reduces stale data li
CONFIG_GCC_PLUGIN_STACKLEAK=y
# CONFIG_STACKLEAK_METRICS is not set
# CONFIG_STACKLEAK_RUNTIME_DISABLE is not set
```

x86_64

```
# Full 64-bit means PAE and NX bit.
CONFIG_X86_64=y

# Disallow allocating the first 64k of memory.
CONFIG_DEFAULT_MMAP_MIN_ADDR=65536

# Disable Model-Specific Register writes.
# CONFIG_X86_MSR is not set

# Randomize position of kernel and memory.
CONFIG_RANDOMIZE_BASE=y
CONFIG_RANDOMIZE_MEMORY=y

# Modern libc no longer needs a fixed-position mapping in us
# CONFIG_X86_VSYSCALL_EMULATION is not set
CONFIG_LEGACY_VSYSCALL_NONE=y

# Enable Kernel Page Table Isolation to remove an entire cla
CONFIG_MITIGATION_PAGE_TABLE_ISOLATION=y

# Enforce CET Indirect Branch Tracking in the kernel. (Since
CONFIG_X86_KERNEL_IBT=y

# Support userspace CET Shadow Stack
CONFIG_X86_USER_SHADOW_STACK=y

# Remove additional (32-bit) attack surface, unless you real
# CONFIG_COMPAT is not set
# CONFIG_IA32_EMULATION is not set
# CONFIG_X86_X32 is not set
# CONFIG_X86_X32_ABI is not set
# CONFIG_MODIFY_LDT_SYSCALL is not set

# Enable chip-specific IOMMU support.
CONFIG_INTEL_IOMMU=y
CONFIG_INTEL_IOMMU_DEFAULT_ON=y
CONFIG_INTEL_IOMMU_SVM=y
CONFIG_AMD_IOMMU=y
CONFIG_AMD_IOMMU_V2=y

# Straight-Line-Speculation
CONFIG_MITIGATION_SLS=y
```

```
# Enable Control Flow Integrity (since v6.1).
CONFIG_CFI_CLANG=y
# CONFIG_CFI_PERMISSIVE is not set

# Dangerous; enabling this disables vDSO ASLR on X86_64 and
# On ARM64 this option has different meaning.
# CONFIG_COMPAT_VDSO is not set
```

arm64

```
# Disallow allocating the first 32k of memory (cannot be 64k
CONFIG_DEFAULT_MMAP_MIN_ADDR=32768

# Randomize position of kernel (requires UEFI RNG or bootloa
CONFIG_RANDOMIZE_BASE=y

# Remove arm32 support to reduce syscall attack surface.
# CONFIG_COMPAT is not set

# Make sure PAN emulation is enabled.
CONFIG_ARM64_SW_TTBR0_PAN=y

# Enable Kernel Page Table Isolation to remove an entire cla
CONFIG_UNMAP_KERNEL_AT_EL0=y

# Enable Software Shadow Stack when hardware Pointer Authent
CONFIG_SHADOW_CALL_STACK=y
CONFIG_UNWIND_PATCH_PAC_INTO_SCS=y

# Pointer authentication (ARMv8.3 and later). If hardware ac
# turn off CONFIG_STACKPROTECTOR_STRONG with this enabled.
CONFIG_ARM64_PTR_AUTH=y
CONFIG_ARM64_PTR_AUTH_KERNEL=y

# Available in ARMv8.5 and later.
CONFIG_ARM64_BTI=y
CONFIG_ARM64_BTI_KERNEL=y
CONFIG_ARM64_MTE=y
CONFIG_KASAN_HW_TAGS=y
CONFIG_ARM64_E0PD=y

# Available in ARMv8.7 and later.
CONFIG_ARM64_EPAN=y

# Enable Control Flow Integrity
```

```
CONFIG_CFI_CLANG=y
# CONFIG_CFI_PERMISSIVE is not set
```

x86_32

```
# On 32-bit kernels, require PAE for NX bit support.
# CONFIG_M486 is not set
# CONFIG_HIGHMEM4G is not set
CONFIG_HIGHMEM64G=y
CONFIG_X86_PAE=y

# Disallow allocating the first 64k of memory.
CONFIG_DEFAULT_MMAP_MIN_ADDR=65536

# Disable Model-Specific Register writes.
# CONFIG_X86_MSR is not set

# Randomize position of kernel.
CONFIG_RANDOMIZE_BASE=y

# Enable Kernel Page Table Isolation to remove an entire cla
CONFIG_MITIGATION_PAGE_TABLE_ISOLATION=y

# Enable chip-specific IOMMU support.
CONFIG_INTEL_IOMMU=y
CONFIG_INTEL_IOMMU_DEFAULT_ON=y

# Don't allow for 16-bit program emulation and associated LD
# CONFIG_MODIFY_LDT_SYSCALL is not set

# Dangerous; enabling this disables vDSO ASLR on X86_64 and
# On ARM64 this option has different meaning.
# CONFIG_COMPAT_VDSO is not set
```

arm

```
# Disallow allocating the first 32k of memory (cannot be 64k
CONFIG_DEFAULT_MMAP_MIN_ADDR=32768

# For maximal userspace memory area (and maximum ASLR).
CONFIG_VMSPLIT_3G=y

# If building an old out-of-tree Qualcomm kernel, this is si
```

```
CONFIG_STRICT_MEMORY_RWX=y

# Make sure PXN/PAN emulation is enabled.
CONFIG_CPU_SW_DOMAIN_PAN=y

# Dangerous; old interfaces and needless additional attack s
# CONFIG_OABI_COMPAT is not set
```

kernel command line options

```
# Make sure CONFIG_HARDENED_USERCOPY stays enabled.
hardened_usercopy=1

# Wipe slab and page allocations (Since v5.3; supersedes "sl
# See CONFIG_INIT_ON_ALLOC_DEFAULT_ON=y and CONFIG_INIT_ON_F
init_on_alloc=1
init_on_free=1

# Randomize kernel stack offset on syscall entry (since v5.1
# See CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT above.
randomize_kstack_offset=on

# Randomize page allocator (needs CONFIG_SHUFFLE_PAGE_ALLOCA
page_alloc.shuffle=1

# Disable slab merging to make some heap overflow attacks mo
slab_nomerge

# Always enable Kernel Page Table Isolation, even if the CPU
pti=on

# To prevent against L1TF, at the cost of losing hyper threa
nosmt

# Enable SLUB redzoning and sanity checking (slow; requires 
slub_debug=ZF

# (Before v5.3 without "init_on_free=1") Enable slub/slab al
slub_debug=P

# (Before v5.3 without "init_on_free=1") Enable buddy alloca
page_poison=1

# Force IOMMU TLB invalidation so devices will never be able
iommu.passthrough=0 iommu.strict=1
```

```
# Mitigates all known CPU vulnerabilities, disabling SMT *if
mitigations=auto,nosmt

# Another way to enable KFENCE (see CONFIG_KFENCE_SAMPLE_INT
kfence.sample_interval=100
```

x86_64

```
# Remove vsyscall entirely to avoid it being a fixed-positic
# (Same as CONFIG_LEGACY_VSYSCALL_NONE=y above.)
vsyscall=none

# Make sure COMPAT_VDSO stays disabled
vdso32=0

# Disable FineIBT since it is weaker than pure KCFI.
cfi=kcfi
```

sysctls

```
# Try to keep kernel address exposures out of various /proc
# There is no CONFIG for the changing the initial value:
# https://lore.kernel.org/lkml/20101217164431.08f3e730.akpm@
# If root absolutely needs values from /proc, use value "1".
kernel.kptr_restrict = 2

# Avoid kernel memory address exposures via dmesg (this valu
kernel.dmesg_restrict = 1

# Disable module loading. For example, this can be set after
# https://outflux.net/blog/archives/2009/07/31/blocking-modu
kernel.modules_disabled = 1

# Block non-uid-0 profiling (needs distro patch https://patc
# Otherwise this is the same as "= 2".
kernel.perf_event_paranoid = 3

# Turn off kexec, even if it's built in.
kernel.kexec_load_disabled = 1

# Enable all available Address Space Randomization (ASLR) fc
kernel.randomize_va_space = 2
```

```
# Block all PTRACE_ATTACH. If you need ptrace to work, then
kernel.yama.ptrace_scope = 3

# Disable User Namespaces, as it opens up a large attack sur
user.max_user_namespaces = 0

# Disable tty line discipline autoloading (see CONFIG_LDISC_
dev.tty.ldisc_autoload = 0

# Disable TIOCSTI which is used to inject keypresses. (This
dev.tty.legacy_tiocsti = 0

# Turn off unprivileged eBPF access.
kernel.unprivileged_bpf_disabled = 1

# Reboot after even 1 WARN or BUG/Oops. Adjust for your tole
# If you want to set oops_limit greater than one, you will r
kernel.warn_limit = 1
kernel.oops_limit = 1

# Turn on BPF JIT hardening, if the JIT is enabled.
net.core.bpf_jit_harden = 2

# Disable dangerous userfaultfd usage.
vm.unprivileged_userfaultfd = 0

# Disable POSIX symlink and hardlink corner cases that lead
fs.protected_symlinks = 1
fs.protected_hardlinks = 1

# Disable POSIX corner cases with creating files and fifos u
fs.protected_fifos = 2
fs.protected_regular = 2

# Make sure the default process dumpability is set (processe
fs.suid_dumpable = 0
```

Improve this page