

# Безопасность ядра Linux: в теории и на практике

Александр Попов  
Positive Technologies



# Безопасность ядра Linux: в теории и на практике

Александр Попов

Positive Technologies

25.11.2022

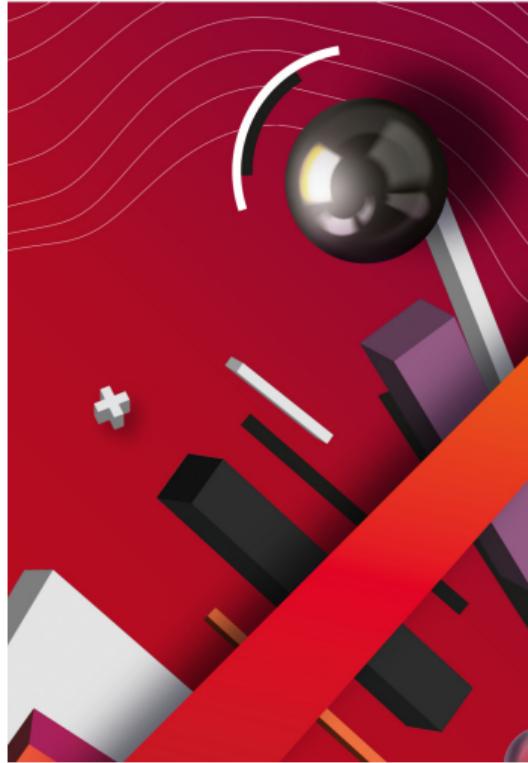


- Александр Попов
- Разработчик ядра Linux с 2012 года
- Исследователь информационной безопасности в
  - **positive technologies**
- Докладчик на конференциях:  
OffensiveCon, Nullcon, Linux Security Summit, Still Hacking Anyway, Zer0Con  
Positive Hack Days, ZeroNights, Open Source Summit, OS DAY, Linux Plumbers  
и других

[a13xp0p0v.github.io/conference\\_talks](https://a13xp0p0v.github.io/conference_talks)

# Цель доклада

- ① Показать вам общую картину безопасности ядра Linux
- ② Рассказать об инструментах, которые помогают:
  - ▶ изучать эту предметную область
  - ▶ управлять безопасностью ядра Linux



# Определимся с терминами

**Операционная система (ОС)** — это программное обеспечение, которое:

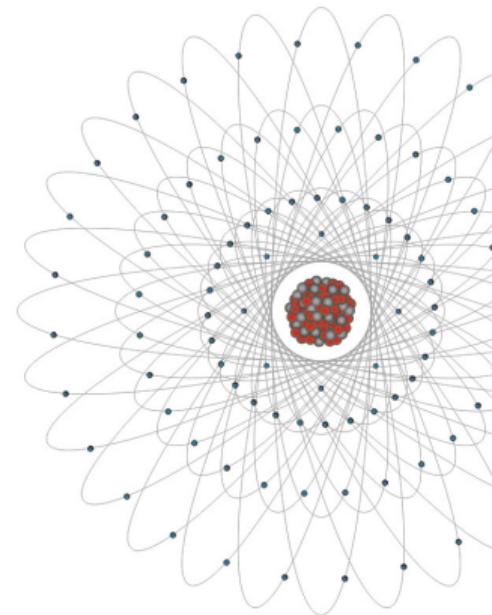
- ① управляет аппаратными и программными ресурсами компьютера
- ② предоставляет сервисы для компьютерных программ

# Определимся с терминами

**Ядро ОС** — основная (*и самая интересная*) часть ОС, управляющая процессами и их доступом к ресурсам вычислительной системы:

- процессорному времени (задача планировщика)
- оперативной памяти
- аппаратным средствам
- механизмам межпроцессного взаимодействия (IPC)

Ядро работает **в привилегированном режиме CPU**

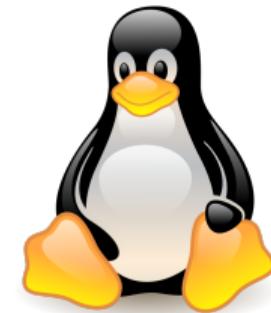


[pediaa.com/difference-between-uranium-and-thorium/](https://pediaa.com/difference-between-uranium-and-thorium/)

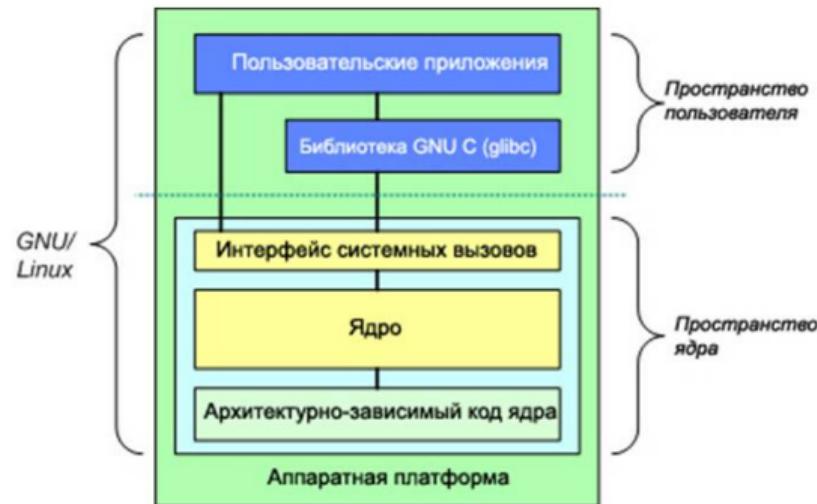
- Ядро Linux — основа семейства операционных систем GNU/Linux и Android
- «One of the most successful collaborative development projects in history»

[linuxfoundation.org/2017-linux-kernel-report-landing-page](https://linuxfoundation.org/2017-linux-kernel-report-landing-page)

- ▶ 90% публичных облачных ресурсов (2017)
- ▶ 62% рынка встраиваемых устройств (2017)
- ▶ 99% рынка суперкомпьютеров (2017)
- ▶ Более 3 млрд активных Android-устройств (2021)  
[theverge.com/2021/5/18/22440813/android-devices-active-number-smartphones-google-2021](https://theverge.com/2021/5/18/22440813/android-devices-active-number-smartphones-google-2021)
- ▶ Разрабатывается под открытой лицензией **GPL 2.0**
- ▶ Более 4000 разработчиков принимают участие ежегодно с 2017 года



# Интерфейсы ядра Linux



[ibm.com/developerworks/linux/library/l-linux-kernel/](http://ibm.com/developerworks/linux/library/l-linux-kernel/)

# Определимся с терминами

- Основная задача информационной безопасности — **целесообразная и сбалансированная** защита конфиденциальности, целостности и доступности данных
- «Безопасность — это управление рисками» (Брюс Шнайер)  
[schneier.com/essays/archives/2007/01/information\\_security\\_1.html](http://schneier.com/essays/archives/2007/01/information_security_1.html)
- Для оценки рисков необходима **модель угроз** информационной системы
- Без модели угроз невозможно выработать хорошую **модель безопасности** (как средства защиты устраниют угрозы и снижают риски)

# Определимся с терминами

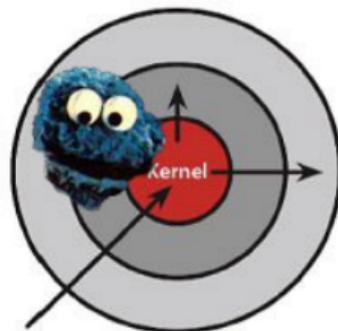
- **Уязвимость** — недостаток в программно-аппаратном обеспечении информационной системы, который может быть использован (проэксплуатирован) для реализации угрозы безопасности (проведения атаки)
- **Экспloit** — программа или последовательность команд, использующая уязвимости в ПО и применяемая для атаки на информационную систему

# Устранение уязвимостей в ядре Linux

- Гигантская кодовая база ядра Linux развивается с огромной скоростью.  
Linux v5.11:
  - ▶ Содержит более **30 млн** строк кода
  - ▶ **8900** строк кода добавляется, **2500** удаляется и **2100** изменяется **каждый день**
  - ▶ Средняя скорость merge — **9,6 патчей в час**
  - ▶ Каждый год в развитии участвуют более **4000** разработчиков (с 2017 года)
- У нас есть санитайзеры, фаззер syzkaller, инструменты статического анализа, но...
- Уязвимости появляются быстрее, чем исправляются  
(«Сказка о тысяче ядерных багов» Дмитрия Вьюкова)

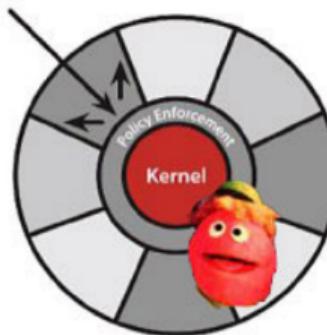
# Уязвимости и механизмы разграничения доступа

- Механизмы разграничения доступа в ядре Linux реализуются с помощью LSM (Linux Security Modules)
- Примеры LSM из «ванильного» ядра: AppArmor, SELinux, Smack
- LSM не защищают систему от эксплуатации ядерных уязвимостей
- Ниже — объяснение от grsecurity



Discretionary Access Control

Once a security exploit gains access to privileged system component, the entire system is compromised.



Mandatory Access Control

Kernel policy defines application rights, firewalling applications from compromising the entire system.



Blackhats with kernel exploits

Basement dwelling 12-year olds armed with kernel exploit released past Tuesday. A SELinux disabling payload in the exploit turns your entire MAC policy into laughing stock. You spend the rest of the weekend removing SSH backdoors.

# Kernel Self Protection Project

- Чтобы повысить безопасность ядра, нужно больше чем исправление ошибок
- Ядро Linux должно безопасно отрабатывать в ошибочной ситуации
- Идеи grsecurity и PaX — во многом источник вдохновения
- **Цель:** устранение классов уязвимостей и методов их эксплуатации
  - ▶ KSPP wiki: [kernsec.org/wiki/index.php/Kernel\\_Self\\_Protection\\_Project](https://kernsec.org/wiki/index.php/Kernel_Self_Protection_Project)
  - ▶ Обзор KSPP (Kees Cook): [outflux.net/slides/2021/lss/kspp.pdf](https://outflux.net/slides/2021/lss/kspp.pdf)



# Безопасность ядра Linux

Безопасность ядра Linux — очень сложная предметная область. Ключевые понятия:

- Классы уязвимостей
- Техники эксплуатации уязвимостей
- Механизмы выявления ошибок
- Технологии защиты
  - ▶ Входящие в mainline
  - ▶ Поставляемые отдельно (в т. ч. коммерческие)
  - ▶ Требующие аппаратной поддержки

Все они имеют сложные взаимосвязи...

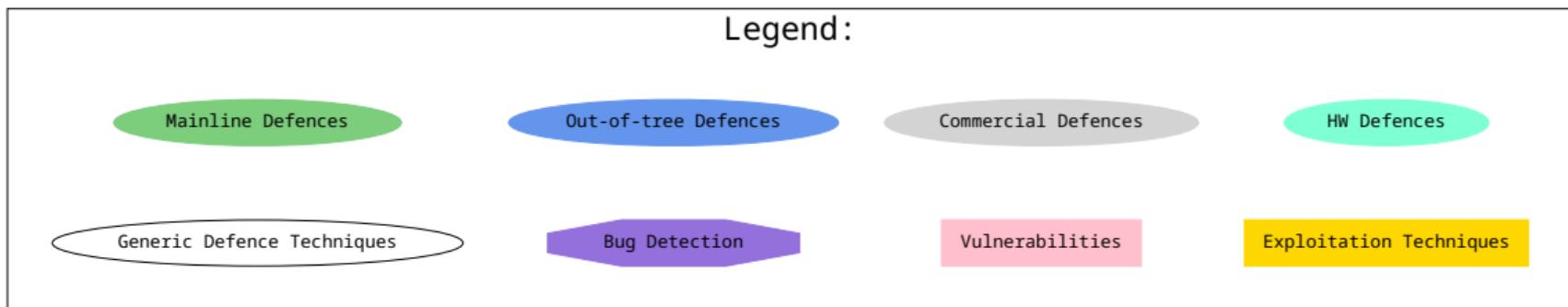
Было бы полезно иметь их графическое представление



Drawn by Daniel Reeve, made by weta

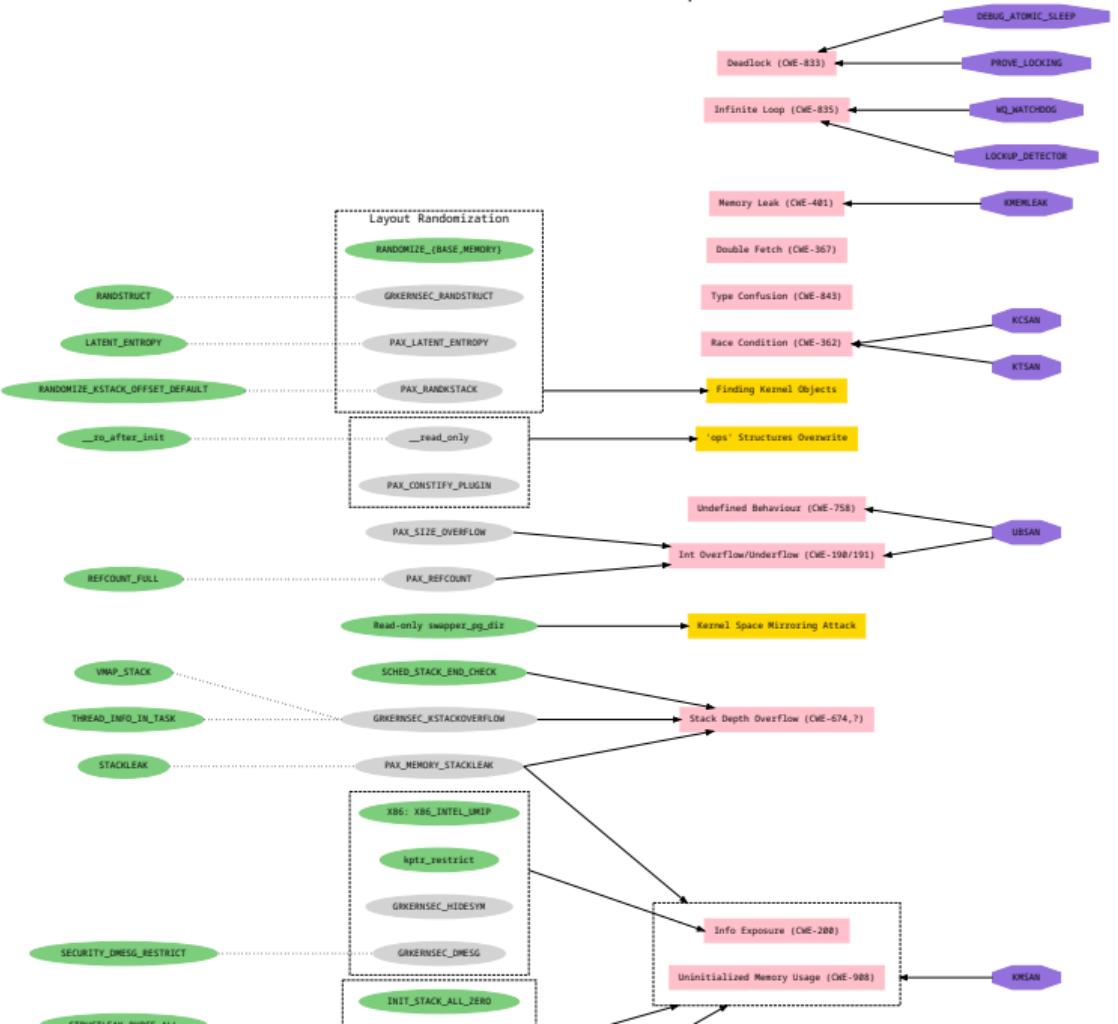
# Linux Kernel Defence Map

- Поэтому я разработал **карту средств защиты ядра Linux**  
[github.com/a13xp0p0v/linux-kernel-defence-map](https://github.com/a13xp0p0v/linux-kernel-defence-map)
- Начал создавать карту в 2018 году, продолжаю улучшать и обновлять ее
- На схеме — ключевые понятия:

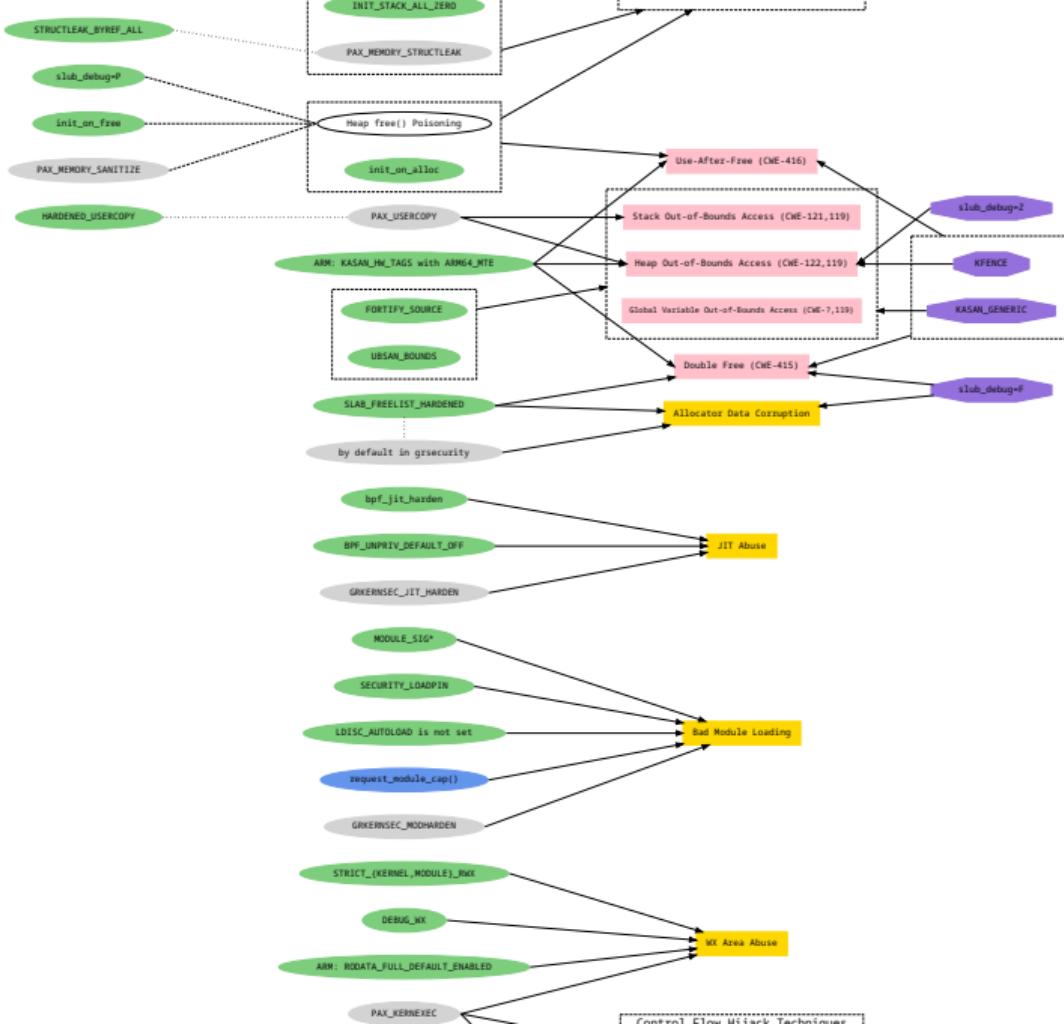


- Каждая линия между объектами на карте обозначает их взаимное влияние
- Суть этого влияния следует выяснить в документации
- [!] Карта не затрагивает способы уменьшения поверхности атаки

# Linux Kernel Defence Map, whole picture (1/5)



## Linux Kernel Defence Map, whole picture (2/5)



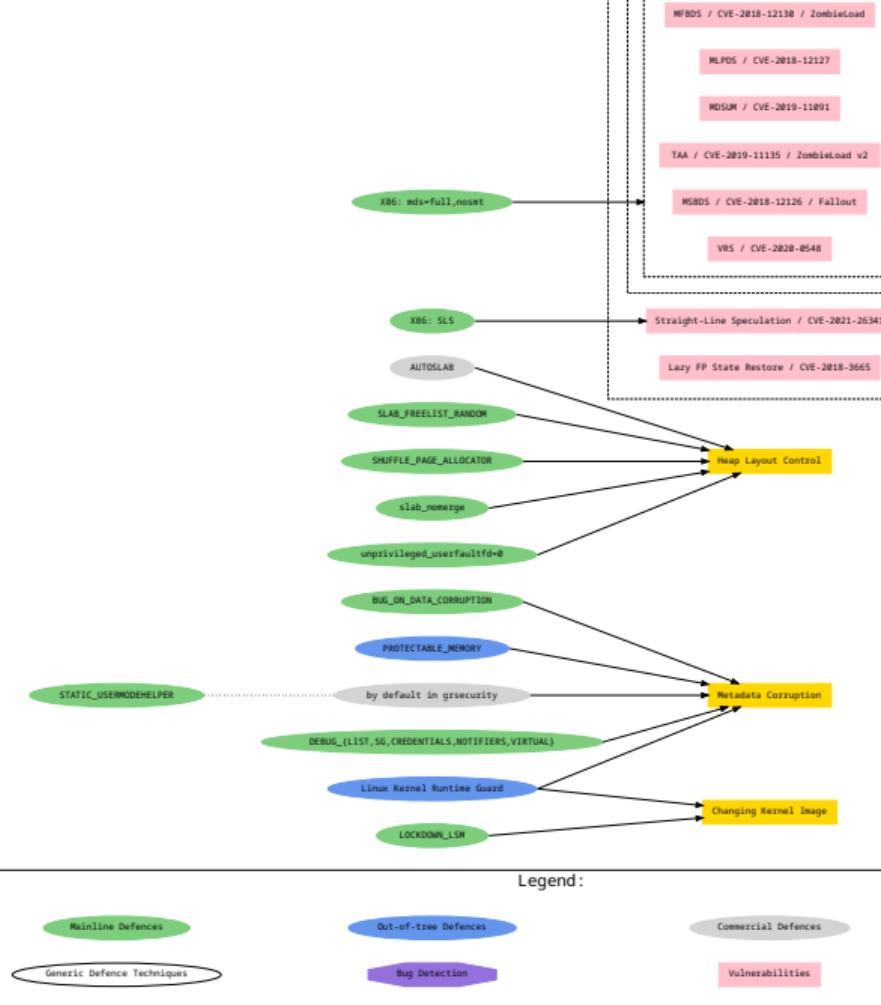
## Linux Kernel Defence Map, whole picture (3/5)



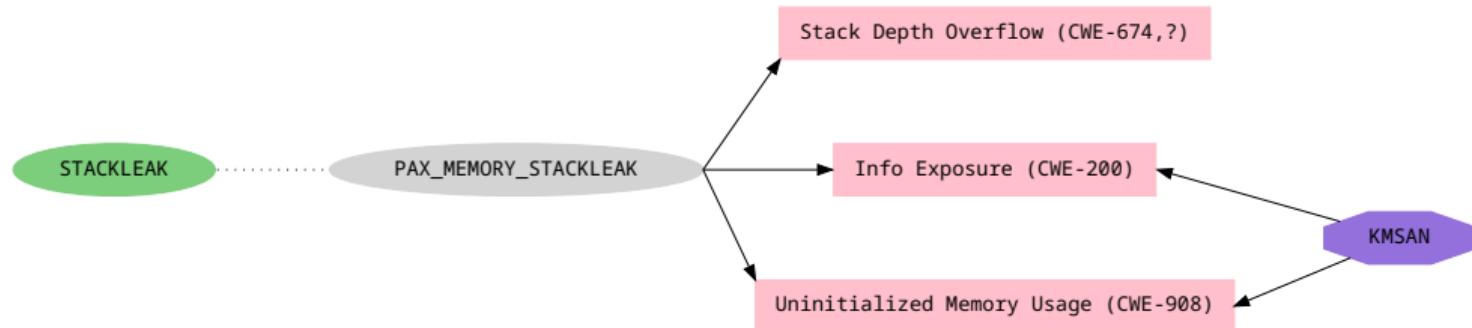
## Linux Kernel Defence Map, whole picture (4/5)



# Linux Kernel Defence Map, whole picture (5/5)



# Примеры из карты: STACKLEAK



Legend:

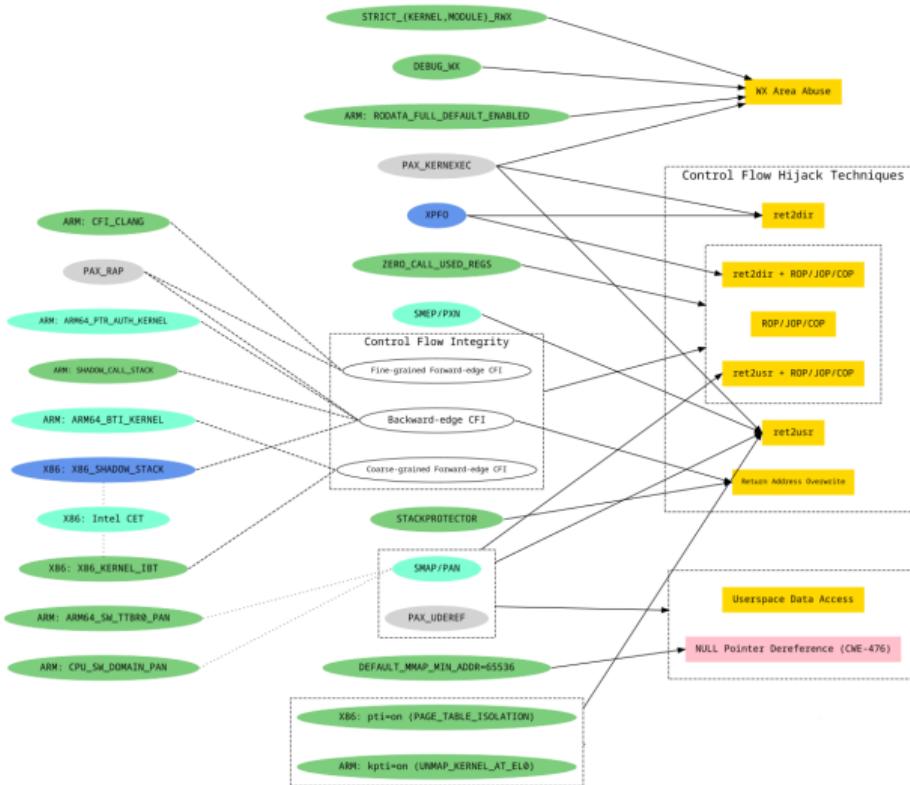
Mainline Defences

Commercial Defences

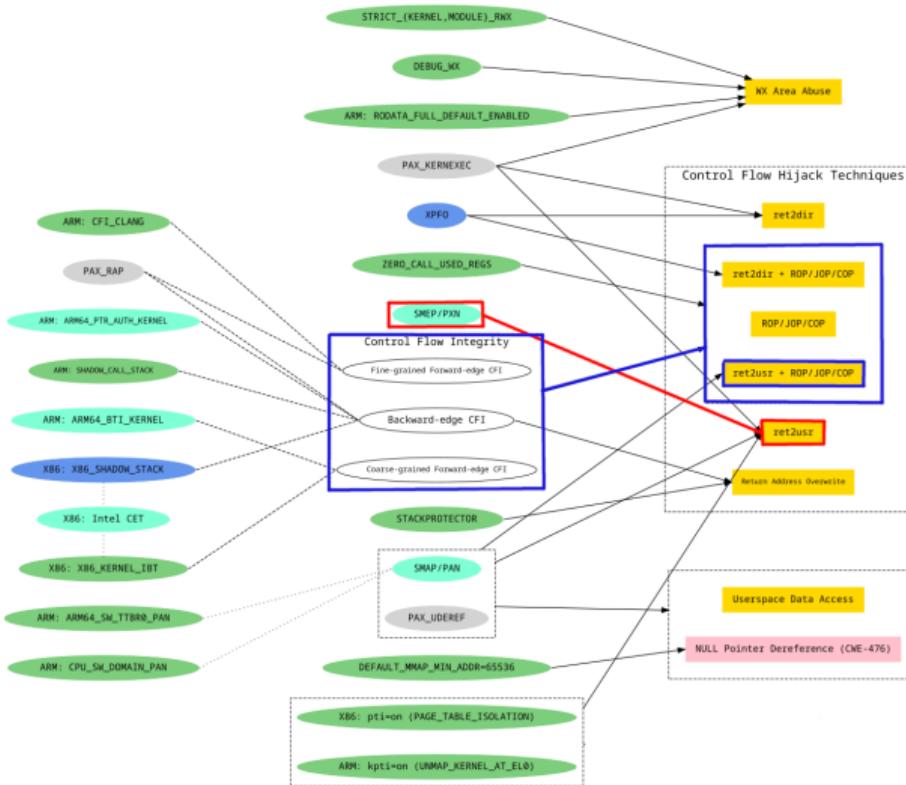
Bug Detection

Vulnerabilities

# Примеры из карты: Control-Flow Hijack — перехват потока управления

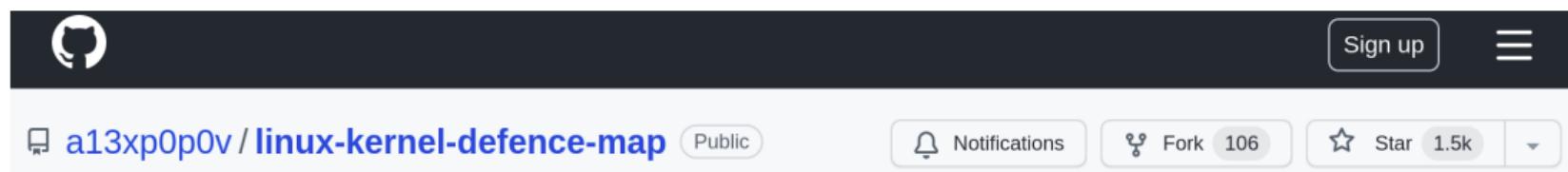


# Примеры из карты: Control-Flow Hijack — перехват потока управления



# Linux Kernel Defence Map: реализация

- Карту нужно обновлять (ядро Linux развивается)
- Желательно иметь исходник в текстовом виде и вести его в VCS
- Не хочется вручную расставлять объекты (с минимальным количеством пересечений связей)
- Поэтому я пишу карту на языке DOT, схему генерирую с помощью Graphviz:  
`# dot -Tsvg map.dot -o map.svg`
- Проект живой и успешный, участие приветствуется



## Пример кода карты

```
// Defences relations

edge [style=dotted, arrowhead=none, dir=none, headport=_ , tailport=_];
"STACKLEAK":e -> "PAX_MEMORY_STACKLEAK":w;

// Bug Detection Mechanisms vs. Vulnerabilities

edge [style=solid, arrowhead=normal, dir=back, headport=_ , tailport=_];
"Uninitialized Memory Usage (CWE-908)":e -> "KMSAN";
"Info Exposure (CWE-200)":e -> "KMSAN";

// Defences vs. Vulnerabilities and Exploitation Techniques

edge [style=solid, arrowhead=normal, dir=forward, headport=_ , tailport=_];
"PAX_MEMORY_STACKLEAK":e -> "Stack Depth Overflow (CWE-674,?)":sw;
"PAX_MEMORY_STACKLEAK":e -> "Uninitialized Memory Usage (CWE-908)":nw;
"PAX_MEMORY_STACKLEAK":e -> "Info Exposure (CWE-200)":w;
```

# Источники для Linux Kernel Defence Map

- Документация ядра Linux (раздел о безопасности)
- Документация grsecurity
- Рекомендации Kernel Self Protection Project
- Публикации Microsoft Security Response Center (MSRC)
- Другие источники: [github.com/a13xp0p0v/linux-kernel-defence-map#references](https://github.com/a13xp0p0v/linux-kernel-defence-map#references)

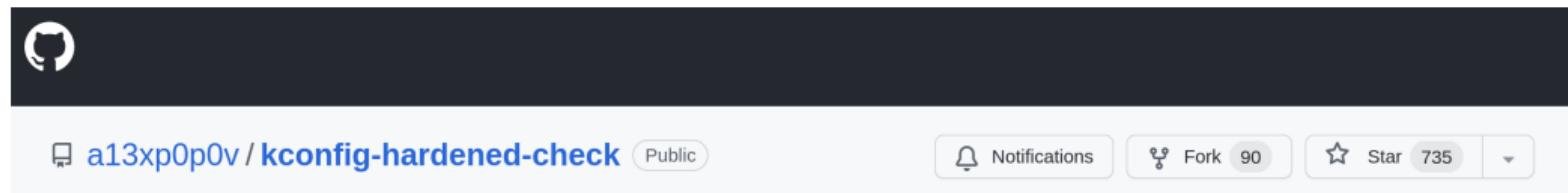
# Классная карта! А что на практике?



В. Е. Белов. «Советские ученые-теоретики» (1972)

# Параметры безопасности ядра

- Есть **огромное количество** параметров безопасности ядра Linux
- В популярных дистрибутивах многие из этих параметров **не настроены**
- Проверять конфигурации вручную **любят не все** (или **не любят все?**)
- **Так пусть компьютеры делают свою работу!**
- Я создал проект **kconfig-hardened-check** для проверки параметров безопасности ядра Linux: [github.com/a13xp0p0v/kconfig-hardened-check](https://github.com/a13xp0p0v/kconfig-hardened-check)
- Работа началась в 2018 году, идет активная разработка



# kconfig-hardened-check -h

```
[a13x@hackbase kconfig-hardened-check]$ ./bin/kconfig-hardened-check
usage: kconfig-hardened-check [-h] [--version] [-p {X86_64,X86_32,ARM64,ARM}] [-c CONFIG]
[-l CMDLINE] [-m {verbose,json,show_ok,show_fail}]
```

A tool for checking the security hardening options of the Linux kernel

optional arguments:

- h, --help show this help message and exit
- version 1 show program's version number and exit
- p {X86\_64,X86\_32,ARM64,ARM}, --print {X86\_64,X86\_32,ARM64,ARM} 2 print security hardening preferences for the selected architecture
- c CONFIG, --config CONFIG 3 check the kernel kconfig file against these preferences
- l CMDLINE, --cmdline CMDLINE 4 check the kernel cmdline file against these preferences
- m {verbose,json,show\_ok,show\_fail}, --mode {verbose,json,show\_ok,show\_fail} choose the report mode

# kconfig-hardened-check: пример вывода (1/5)

```
a13x@hackbase:~/land/Develop/Linux_Kernel/kconfig-hardened-check$ ./bin/kconfig-hardened-check -c /boot/config-5.19.14-200.fc36.x86_64 -l /proc/cmdline
[+] Kconfig file to check: /boot/config-5.19.14-200.fc36.x86_64
[+] Kernel cmdline file to check: /proc/cmdline
[+] Detected architecture: X86_64
[+] Detected kernel version: 5.19
[+] Detected compiler: GCC 120201
=====
option name | type | desired val | decision | reason | check result
=====
CONFIG_BUG | kconfig | y | defconfig | self_protection | OK
CONFIG_SLUB_DEBUG | kconfig | y | defconfig | self_protection | OK
CONFIG_GCC_PLUGINS | kconfig | y | defconfig | self_protection | OK
CONFIG_STACKPROTECTOR | kconfig | y | defconfig | self_protection | OK
CONFIG_STACKPROTECTOR_STRONG | kconfig | y | defconfig | self_protection | FAIL: "is not set"
CONFIG_STRICT_KERNEL_RMX | kconfig | y | defconfig | self_protection | OK
CONFIG_STRICT_MODULE_RMX | kconfig | y | defconfig | self_protection | OK
CONFIG_REFCOUNT_FULL | kconfig | y | defconfig | self_protection | OK: version >= 5.5
CONFIG_THREAD_INFO_IN_TASK | kconfig | y | defconfig | self_protection | OK
CONFIG_IOMMU_SUPPORT | kconfig | y | defconfig | self_protection | OK
CONFIG_RANDOMIZE_BASE | kconfig | y | defconfig | self_protection | OK
CONFIG_VMAP_STACK | kconfig | y | defconfig | self_protection | OK
CONFIG_X86_MCE | kconfig | y | defconfig | self_protection | OK
CONFIG_X86_MCE_INTEL | kconfig | y | defconfig | self_protection | OK
CONFIG_X86_MCE_AMD | kconfig | y | defconfig | self_protection | OK
CONFIG_MICROCODE | kconfig | y | defconfig | self_protection | OK
CONFIG_RETHOOKLINE | kconfig | y | defconfig | self_protection | OK
CONFIG_SYN_COOKIES | kconfig | y | defconfig | self_protection | OK: version >= 5.19
CONFIG_X86_UNIP | kconfig | y | defconfig | self_protection | OK
CONFIG_PAGE_TABLE_ISOLATION | kconfig | y | defconfig | self_protection | OK
CONFIG_RANDOMIZE_MEMORY | kconfig | y | defconfig | self_protection | OK
CONFIG_INTEL_IOMMU | kconfig | y | defconfig | self_protection | OK
CONFIG_AND_IOMMU | kconfig | y | defconfig | self_protection | OK
CONFIG_DEBUG_ON_DATA_CORRUPTION | kconfig | y | kspp | self_protection | OK
CONFIG_DEBUG_WX | kconfig | y | kspp | self_protection | OK
CONFIG_SCHED_STACK_END_CHECK | kconfig | y | kspp | self_protection | OK
CONFIG_SLAB_FREELIST_HARDENED | kconfig | y | kspp | self_protection | OK
CONFIG_SLAB_FREELIST_RANDOM | kconfig | y | kspp | self_protection | OK
CONFIG_SHUFFLE_PAGE_ALLOCATOR | kconfig | y | kspp | self_protection | OK
CONFIG_FORTIFY_SOURCE | kconfig | y | kspp | self_protection | OK
CONFIG_DEBUG_LIST | kconfig | y | kspp | self_protection | OK
CONFIG_DEBUG_VIRTUAL | kconfig | y | kspp | self_protection | FAIL: "is not set"
CONFIG_DEBUG_SG | kconfig | y | kspp | self_protection | FAIL: "is not set"
CONFIG_DEBUG_CREDENTIALS | kconfig | y | kspp | self_protection | FAIL: "is not set"
CONFIG_DEBUG_NOTIFIERS | kconfig | y | kspp | self_protection | FAIL: "is not set"
CONFIG_INIT_ON_ALLOC_DEFAULT_ON | kconfig | y | kspp | self_protection | FAIL: "is not set"
CONFIG_GCC_PLUGIN_LATENT_ENTROPY | kconfig | y | kspp | self_protection | FAIL: "is not set"
CONFIG_KFENCE | kconfig | y | kspp | self_protection | OK
```

# kconfig-hardened-check: пример вывода (2/5)

CONFIG_KFENCE	[kconfig]	y		kspp	self_protection	OK
CONFIG_WERROR	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_IOMMU_DEFAULT_DMA_STRICT	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_IOMMU_DEFAULT_PASSTHROUGH	[kconfig]	is not set		kspp	self_protection	OK
CONFIG_ZERO_CALL_USED_REGS	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_IW_RANDOM_TPM	[kconfig]	y		kspp	self_protection	OK
CONFIG_STATIC_USERMODEHELPER	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_SCHED_CORE	[kconfig]	y		kspp	self_protection	OK
CONFIG_RANDSTRUCT_FULL	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_RANDSTRUCT_PERFORMANCE	[kconfig]	is not set		kspp	self_protection	FAIL: CONFIG_RANDSTRUCT_FULL is not "y"
CONFIG_HARDENED_USERCOPY	[kconfig]	y		kspp	self_protection	OK
CONFIG_HARDENED_USERCOPY_FALLBACK	[kconfig]	is not set		kspp	self_protection	OK: is not found
CONFIG_HARDENED_USERCOPY_PAGESPAN	[kconfig]	is not set		kspp	self_protection	OK: is not found
CONFIG_MODULE_SIG	[kconfig]	y		kspp	self_protection	OK
CONFIG_MODULE_SIG_ALL	[kconfig]	y		kspp	self_protection	OK
CONFIG_MODULE_SIG_SHA512	[kconfig]	y		kspp	self_protection	OK
CONFIG_MODULE_SIG_FORCE	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_INIT_STACK_ALL_ZERO	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_INIT_ON_FREE_DEFAULT_ON	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_EFI_DISABLE_PCI_DMA	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_RESET_ATTACK_MITIGATION	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_UBSAN_BOUNDS	[kconfig]	y		kspp	self_protection	FAIL: is not found
CONFIG_UBSAN_LOCAL_BOUNDS	[kconfig]	y		kspp	self_protection	FAIL: is not found
CONFIG_UBSAN_TRAP	[kconfig]	y		kspp	self_protection	FAIL: CONFIG_UBSAN_BOUNDS is not "y"
CONFIG_UBSAN_SANITIZE_ALL	[kconfig]	y		kspp	self_protection	FAIL: CONFIG_UBSAN_BOUNDS is not "y"
CONFIG_GCC_PLUGIN_STACKLEAK	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_STACKLEAK_METRICS	[kconfig]	is not set		kspp	self_protection	FAIL: CONFIG_GCC_PLUGIN_STACKLEAK is not "y"
CONFIG_STACKLEAK_RUNTIME_DISABLE	[kconfig]	is not set		kspp	self_protection	FAIL: CONFIG_GCC_PLUGIN_STACKLEAK is not "y"
CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT	[kconfig]	y		kspp	self_protection	OK
CONFIG_CFI_CLANG	[kconfig]	y		kspp	self_protection	FAIL: is not found
CONFIG_CFI_PERMISSIVE	[kconfig]	is not set		kspp	self_protection	FAIL: CONFIG_CFI_CLANG is not "y"
CONFIG_DEFAULT_MMAP_MIN_ADDR	[kconfig]	65536		kspp	self_protection	OK
CONFIG_INTEL_IOMMU_DEFAULT_ON	[kconfig]	y		kspp	self_protection	FAIL: "is not set"
CONFIG_SLS	[kconfig]	y		kspp	self_protection	OK
CONFIG_INTEL_IOMMU_SVM	[kconfig]	y		kspp	self_protection	OK
CONFIG_AND_IOMMU_V2	[kconfig]	y		kspp	self_protection	FAIL: "y"
CONFIG_SLAB_MERGE_DEFAULT	[kconfig]	is not set		clipos	self_protection	OK
CONFIG_SECURITY	[kconfig]	y		defconfig	security_policy	OK
CONFIG_SECURITY_YAMA	[kconfig]	y		kspp	security_policy	OK
CONFIG_SECURITY_LANDLOCK	[kconfig]	y		kspp	security_policy	OK
CONFIG_SECURITY_SELINUX_DISABLE	[kconfig]	is not set		kspp	security_policy	OK
CONFIG_SECURITY_SELINUX_BOOTPARAM	[kconfig]	is not set		kspp	security_policy	FAIL: "y"
CONFIG_SECURITY_SELINUX DEVELOP	[kconfig]	is not set		kspp	security_policy	FAIL: "y"
CONFIG_SECURITY_LOCKDOWN_LSM	[kconfig]	y		kspp	security_policy	OK
CONFIG_SECURITY_LOCKDOWN_LSM_EARLY	[kconfig]	y		kspp	security_policy	OK
CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY	[kconfig]	y		kspp	security_policy	FAIL: "is not set"
CONFIG_SECURITY_WRITABLE_HOOKS	[kconfig]	is not set		kspp	security_policy	OK: is not found
CONFIG_BPF_UNPRIV_DEFAULT_OFF	[kconfig]	y		defconfig	cut_attack_surface	OK

# kconfig-hardened-check: пример вывода (3/5)

a13x@hackbase:~/land/Develop/Linux_Kernel/kconfig-hardened-check			
CONFIG_BPF_UNPRIV_DEFAULT_OFF	[kconfig]	y	[defconfig]   cut_attack_surface  OK
CONFIG_SECCOMP	[kconfig]	y	[defconfig]   cut_attack_surface  OK
CONFIG_SECCOMP_FILTER	[kconfig]	y	[defconfig]   cut_attack_surface  OK
CONFIG_STRICT_DEVMEM	[kconfig]	y	[defconfig]   cut_attack_surface  OK
CONFIG_SECURITY_DMESG_RESTRICT	[kconfig]	y	[kconfig]   kspc   cut_attack_surface  FAIL: "is not set"
CONFIG ACPI_CUSTOM_METHOD	[kconfig]	is not set	[kspc]   cut_attack_surface  OK
CONFIG_COMPAT_BRK	[kconfig]	is not set	[kspc]   cut_attack_surface  OK
CONFIG_DEVKMEM	[kconfig]	is not set	[kspc]   cut_attack_surface  OK: is not found
CONFIG_COMPAT_VDSO	[kconfig]	is not set	[kspc]   cut_attack_surface  OK
CONFIG_BINFMT_MISC	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "m"
CONFIG_INET_DIAG	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_KEXEC	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_PROC_KCORE	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_LEGACY_PTYS	[kconfig]	is not set	[kspc]   cut_attack_surface  OK
CONFIG_HIBERNATION	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_IA32_EMULATION	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_X86_X32	[kconfig]	is not set	[kspc]   cut_attack_surface  OK: is not found
CONFIG MODIFY_LDT_SYSCALL	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_OABI_COMPAT	[kconfig]	is not set	[kspc]   cut_attack_surface  OK: is not found
CONFIG_X86_MSR	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_MODULES	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_DEVHEN	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_IO_STRICT_DEVMEM	[kconfig]	y	[kspc]   cut_attack_surface  OK
CONFIG_LDTS_AUTOLOAD	[kconfig]	is not set	[kspc]   cut_attack_surface  FAIL: "y"
CONFIG_LEGACY_VSYSCALL_NONE	[kconfig]	y	[kspc]   cut_attack_surface  FAIL: "is not set"
CONFIG_ZSMALLOC_STAT	[kconfig]	is not set	[grsec]   cut_attack_surface  OK
CONFIG_PAGE_OWNER	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_DEBUG_KMEMLEAK	[kconfig]	is not set	[grsec]   cut_attack_surface  OK
CONFIG_BINFMT_AOUT	[kconfig]	is not set	[grsec]   cut_attack_surface  OK: is not found
CONFIG_KPROBE_EVENTS	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_UPROBE_EVENTS	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_GENERIC_TRACER	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_FUNCTION_TRACER	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_STACK_TRACER	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_HIST_TRIGGERES	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_BLK_DEV_IO_TRACE	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_PROC_VMCORE	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_PROC_PAGE_MONITOR	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_USELIB	[kconfig]	is not set	[grsec]   cut_attack_surface  OK
CONFIG_CHECKPOINT_RESTORE	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_USERFAULTFD	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_HMPOLISON_INJECT	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "m"
CONFIG_MEM_SOFT_DIRTY	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_DEVPORT	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_DEBUG_FS	[kconfig]	is not set	[grsec]   cut_attack_surface  FAIL: "y"
CONFIG_NOTIFIER_ERROR_INJECTION	[kconfig]	is not set	[grsec]   cut_attack_surface  OK
CONFIG_FAIL_FUTEX	[kconfig]	is not set	[grsec]   cut_attack_surface  OK: is not found
CONFIG_PUNIT_ATOM_DEBUG	[kconfig]	is not set	[grsec]   cut_attack_surface  OK

# kconfig-hardened-check: пример вывода (4/5)

CONFIG_PUNIT_ATOM_DEBUG	[kconfig] is not set   grsec	cut_attack_surface  OK		
CONFIG ACPI_CONFIGFS	[kconfig] is not set   grsec	cut_attack_surface  OK		
CONFIG_EDAC_DEBUG	[kconfig] is not set   grsec	cut_attack_surface  OK		
CONFIG_DRM_I915_DEBUG	[kconfig] is not set   grsec	cut_attack_surface  OK; is not found		
CONFIG_BCACHE_CLOSURES_DEBUG	[kconfig] is not set   grsec	cut_attack_surface  OK		
CONFIG_DVB_C8SECTPFE	[kconfig] is not set   grsec	cut_attack_surface  OK; is not found		
CONFIG_MTD_SLRAM	[kconfig] is not set   grsec	cut_attack_surface  OK		
CONFIG_MTD_PHRAM	[kconfig] is not set   grsec	cut_attack_surface  OK		
CONFIG_ID_URING	[kconfig] is not set   grsec	cut_attack_surface  FAIL: "y"		
CONFIG_KCMP	[kconfig] is not set   grsec	cut_attack_surface  FAIL: "y"		
CONFIG_RSEQ	[kconfig] is not set   grsec	cut_attack_surface  FAIL: "y"		
CONFIG_LATENCYTOP	[kconfig] is not set   grsec	cut_attack_surface  FAIL: "y"		
CONFIG_KCOV	[kconfig] is not set   grsec	cut_attack_surface  OK		
CONFIG_PROVIDE_OHCI1394_DMA_INIT	[kconfig] is not set   grsec	cut_attack_surface  FAIL: "y"		
CONFIG_SUNRPC_DEBUG	[kconfig] is not set   grsec	cut_attack_surface  FAIL: "y"		
CONFIG_PTIDUMP_DEBUGFS	[kconfig] is not set   grsec	cut_attack_surface  OK		
CONFIG_DRM_LEGACY	[kconfig] is not set   maintainer	cut_attack_surface  OK		
CONFIG_FB	[kconfig] is not set   maintainer	cut_attack_surface  FAIL: "y"		
CONFIG_VT	[kconfig] is not set   maintainer	cut_attack_surface  FAIL: "y"		
CONFIG_BLK_DEV_FD	[kconfig] is not set   maintainer	cut_attack_surface  FAIL: "m"		
CONFIG_BLK_DEV_FD_RAWCMD	[kconfig] is not set   maintainer	cut_attack_surface  OK		
CONFIG_AIO	[kconfig] is not set   grapheneos	cut_attack_surface  FAIL: "y"		
CONFIG_STAGING	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_KSM	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_KALLSYMS	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_X86_VSYSCALL_EMULATION	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_MAGIC_SYSRQ	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_KEXEC_FILE	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_USER_NS	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_X86_CPUID	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_X86_IOPL_IOPERM	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_ACPI_TABLE_UPGRADE	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_EFI_CUSTOM_SSDDT_OVERLAYS	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_COREDUMP	[kconfig] is not set   clipos	cut_attack_surface  FAIL: "y"		
CONFIG_X86_INTEL_TSX_MODE_OFF	[kconfig] y   clipos	cut_attack_surface  OK		
CONFIG_BPF_SYSCALL	[kconfig] is not set   lockdown	cut_attack_surface  FAIL: "y"		
CONFIG_EFI_TEST	[kconfig] is not set   lockdown	cut_attack_surface  FAIL: "m"		
CONFIG_MMOTRACE_TEST	[kconfig] is not set   lockdown	cut_attack_surface  OK		
CONFIG_KPROBES	[kconfig] is not set   lockdown	cut_attack_surface  FAIL: "y"		
CONFIG_TRIM_UNUSED_KSYMS	[kconfig] y   my	cut_attack_surface  FAIL: is not found		
CONFIG_MMOTRACE	[kconfig] is not set   my	cut_attack_surface  FAIL: "y"		
CONFIG_LIVEPATCH	[kconfig] is not set   my	cut_attack_surface  FAIL: "y"		
CONFIG_IP_DCCP	[kconfig] is not set   my	cut_attack_surface  OK		
CONFIG_IP_SCTP	[kconfig] is not set   my	cut_attack_surface  FAIL: "m"		
CONFIG_FTRACE	[kconfig] is not set   my	cut_attack_surface  FAIL: "y"		
CONFIG_VIDEO_VIVID	[kconfig] is not set   my	cut_attack_surface  OK; is not found		
CONFIG_INPUT_EVBUG	[kconfig] is not set   my	cut_attack_surface  OK		
CONFIG_KGDB	[kconfig] is not set   my	cut_attack_surface  FAIL: "y"		

# kconfig-hardened-check: пример вывода (5/5)

```
a13x@hackbase:~/land/Develop/Linux_Kernel/kconfig-hardened-check
```

Config Option	Value	Check	Result
CONFIG_XEXEC_FILE	[kconfig] is not set	clipos   cut_attack_surface	FAIL: "y"
CONFIG_USER_NS	[kconfig] is not set	clipos   cut_attack_surface	FAIL: "y"
CONFIG_X86_CPUID	[kconfig] is not set	clipos   cut_attack_surface	FAIL: "y"
CONFIG_X86_IOPL_IOPERM	[kconfig] is not set	clipos   cut_attack_surface	FAIL: "y"
CONFIG_ACPI_TABLE_UPGRADE	[kconfig] is not set	clipos   cut_attack_surface	FAIL: "y"
CONFIG_EFI_CUSTOM_SSDT_OVERLAYS	[kconfig] is not set	clipos   cut_attack_surface	FAIL: "y"
CONFIG_COREDUMP	[kconfig] is not set	clipos   cut_attack_surface	FAIL: "y"
CONFIG_X86_INTEL_TSX_MODE_OFF	[kconfig] y	clipos   cut_attack_surface	OK
CONFIG_BPF_SYSCALL	[kconfig] is not set	lockdown   cut_attack_surface	FAIL: "y"
CONFIG_EFI_TEST	[kconfig] is not set	lockdown   cut_attack_surface	FAIL: "y"
CONFIG_MMOTRACE_TEST	[kconfig] is not set	lockdown   cut_attack_surface	OK
CONFIG_XPROBES	[kconfig] is not set	lockdown   cut_attack_surface	FAIL: "y"
CONFIG_TRIM_UNUSED_KSYMS	[kconfig] y	my   cut_attack_surface	FAIL: is not found
CONFIG_MMOTRACE	[kconfig] is not set	my   cut_attack_surface	FAIL: "y"
CONFIG_LIVEPATCH	[kconfig] is not set	my   cut_attack_surface	FAIL: "y"
CONFIG_IP_DDCP	[kconfig] is not set	my   cut_attack_surface	OK
CONFIG_IP_SCTP	[kconfig] is not set	my   cut_attack_surface	FAIL: "y"
CONFIG_FTRACE	[kconfig] is not set	my   cut_attack_surface	FAIL: "y"
CONFIG_VIDEO_VIVID	[kconfig] is not set	my   cut_attack_surface	OK: is not found
CONFIG_INPUT_EVBUG	[kconfig] is not set	my   cut_attack_surface	OK
CONFIG_KGDB	[kconfig] is not set	my   cut_attack_surface	FAIL: "y"
CONFIG_INTEGRITY	[kconfig] y	defconfig   harden_userspace	OK
CONFIG_ARCH_MMAP_RND_BITS	[kconfig] 32	clipos   harden_userspace	FAIL: "28"
nosmp	[cmdline] is not set	defconfig   self_protection	OK: is not found
nosmap	[cmdline] is not set	defconfig   self_protection	OK: is not found
mokaslr	[cmdline] is not set	defconfig   self_protection	OK: is not found
noptl	[cmdline] is not set	defconfig   self_protection	OK: is not found
nospectre_v1	[cmdline] is not set	defconfig   self_protection	OK: is not found
nospectre_v2	[cmdline] is not set	defconfig   self_protection	OK: is not found
mitigations	[cmdline] is not off	defconfig   self_protection	OK: mitigations is not found
rodata	[cmdline] 1	defconfig   self_protection	OK: rodata is not found
nosmt	[cmdline]	kspc   self_protection	FAIL: is not present
init_on_alloc	[cmdline] 1	kspc   self_protection	FAIL: is not found
init_on_free	[cmdline] 1	kspc   self_protection	FAIL: is not found
slab_nomerge	[cmdline] 1	kspc   self_protection	OK: CONFIG_SLAB_MERGE_DEFAULT is "is not set"
iommu.strict	[cmdline] 1	kspc   self_protection	FAIL: is not found
iommu.passthrough	[cmdline] 0	kspc   self_protection	OK: CONFIG_IOMMU_DEFAULT_PASSTHROUGH is "is not set"
hardened_usercopy	[cmdline] 1	kspc   self_protection	OK: CONFIG_HARDENED_USERCOPY is "y"
slab_common.usercopy_fallback	[cmdline] 0	kspc   self_protection	OK: CONFIG_HARDENED_USERCOPY_FALLBACK is not found
randomize_kstack_offset	[cmdline] 1	kspc   self_protection	OK: CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT is "y"
pti	[cmdline] on	kspc   self_protection	FAIL: is not found
page_alloc.shuffle	[cmdline] 1	clipos   self_protection	FAIL: is not found
spectre_v2	[cmdline] on	clipos   self_protection	FAIL: is not found
vsyscall	[cmdline] none	kspc   cut_attack_surface	FAIL: is not found
debugfs	[cmdline] off	grsec   cut_attack_surface	FAIL: is not found

```
[+] Config check is finished: 'OK' - 182 / 'FAIL' - 182  
[a13x@hackbase kconfig-hardened-check]$
```

## Источники для kconfig-hardened-check

- Рекомендации KSPP
- Конфигурация ядра CLIP OS
- Последний публичный патч grsecurity (отключаемые параметры)
- SECURITY\_LOCKDOWN\_LSM
- Обратная связь непосредственно от мейнтейнеров ядра Linux

# kconfig-hardened-check: что под капотом

- Лицензия [GPL 3.0](#)
- Код на [Python](#). Извините, мой код выглядит, как код на C:  
это потому что я разработчик ядра :)
- Установка с помощью [pip/setuptools](#)
- Регулярные релизы (с привязкой к релизам ядра)
- [CI](#): автоматические функциональные тесты с подсчетом покрытия

## ↗ kconfig-hardened-check

release v0.5.17 functional test passing codecov 95%

- В работах участвуют контрибьюторы из международного сообщества (спасибо!)
- Инструмент используется несколькими дистрибутивами GNU/Linux  
при выпуске пакета ядра (что очень радует)
- Технологический центр исследования безопасности ядра Linux  
поддерживает мой проект (спасибо!)

# kconfig-hardened-check: идеи и планы

- Внедрить проверку динамических параметров безопасности ядра Linux:
  - ▶ kernel cmdline (WIP)
  - ▶ sysctl
- Реализовать возможность переопределения и расширения набора проверок
- Разработать инструмент для автоматизированной работы с Kconfig
- Добавить поддержку RISC-V
- Измерить накладные расходы производительности для рекомендуемых опций (зависят от типа рабочей нагрузки)
- Разработать документацию по параметрам безопасности ядра Linux

В. Е. Тихоненко. «Высотники. На новой стройке» (1975)



## В заключение: как пользоваться этими инструментами

- «Безопасность — это управление рисками» (Брюс Шнайер)
- Для оценки рисков необходима **модель угроз** информационной системы
- Определив угрозы ИС на базе Linux,  
можно заняться разработкой ее **модели безопасности**:
  - ▶ С помощью **Linux Kernel Defence Map** определить релевантные средства защиты ядра
  - ▶ С помощью **kconfig-hardened-check** проверить и настроить соответствующие параметры
- Пример отличной модели безопасности: The Android Platform Security Model  
[arxiv.org/abs/1904.05572](https://arxiv.org/abs/1904.05572)



# Спасибо! Вопросы?

Контакты:

✉ alex.popov@linux.com

    a13xp0p0v

■ **positive technologies**

 POSIdev

Оценить доклад:

