3 июня 2025 Москва, LOFT HALL#2

БЕКОН '25

Конференция по БЕзопасности
КОНтейнеров и контейнерных сред

Тонкая настройка безопасности OS Linux для контейнеров: вредно и полезно

Альмир Сарваров

Эксперт Банк ДОМ.РФ

whoami

БЕКОН

- Эксперт по управлению уязвимостями
- Аудит инфраструктуры АСУ ТП и корпоративных сетей
- 15 лет в IT, из которых 7 в ИБ
- Я твой прод труба шатал

СОДЕРЖАНИЕ

БЕКОН

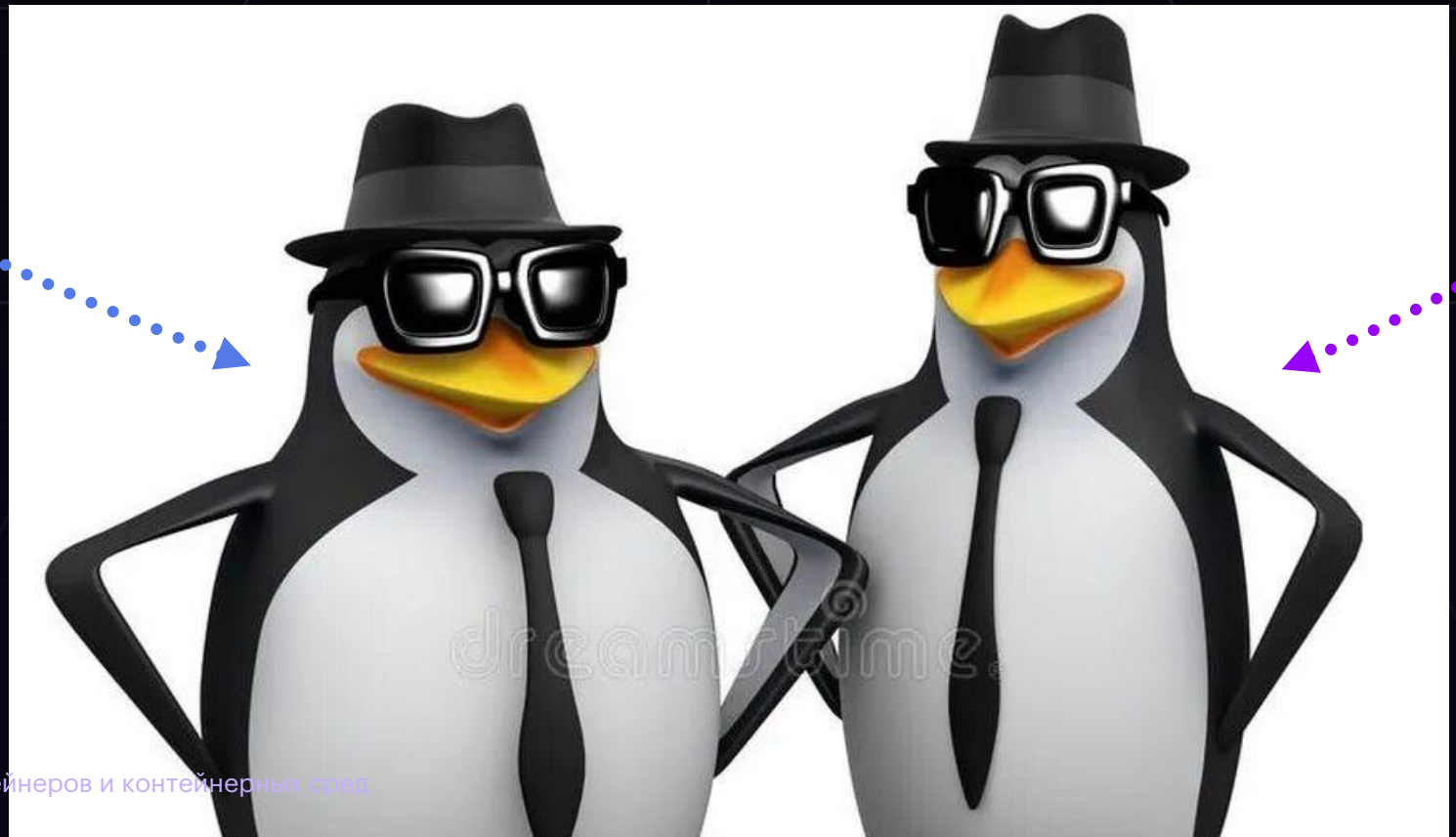
- Linux Kernel Defence Map (LKDM) и Linux Kernel Self Protection Project (LKSPP)
- CIS Benchmark
- Сравнение практик CIS Benchmark и Linux Kernel Self Protection Project
- Hardening OS Linux (Node)
- Пример защиты параметров CVE
- Положительный опыт – проблемы и падение сервисов, потеря доступности

KSPP and LKDM

БЕКОН

Limux Kernel Self Protection Project (LKSPP)

Linux Kernel Defence Map (LKDM)



БЕКОН

Linux Kernel Defence Map

LKDM

Linux Kernel Defence Map (LKDM)

БЕКОН

Цель:

- Упростить навигацию по механизмам защиты ядра Linux
- Связать классы уязвимостей, методы эксплуатации, механизмы обнаружения багов и технологии защиты
- Обеспечить визуальную карту зависимостей и покрытий
- Предоставить CWE-ссылки на классы уязвимостей
- Повысить осознанность по аппаратным и программным защитам

Linux Kernel Defence Map (LKDM)

БЕКОН

Цель:

- Упростить навигацию по механизмам защиты ядра Linux
- Связать классы уязвимостей, методы эксплуатации, механизмы обнаружения багов и технологии защиты
- Обеспечить визуальную карту зависимостей и покрытий
- Предоставить CWE-ссылки на классы уязвимостей
- Повысить осознанность по аппаратным и программным защитам

Особенности проекта

- Написан на DOT языке → легко поддерживать в Git
- Не охватывает LSM и userspace-защиты
- Лицензия: GPL v3. Последняя версия: v6.10
- Репозиторий:

https://github.com/a13xp0p0v/linux-kernel-defence-mapCodeberg

https://codeberg.org/a13xp0p0v/linux-kernel-defence-map

https://gitflic.ru/project/a13xp0p0v/linux-kernel-defence-map

Linux Kernel Defence Map (LKDM)

БЕКОН

Цель:

- Упростить навигацию по механизмам защиты ядра Linux
- Связать классы уязвимостей, методы эксплуатации, механизмы обнаружения багов и технологии защиты
- Обеспечить визуальную карту зависимостей и покрытий
- Предоставить CWE-ссылки на классы уязвимостей
- Повысить осознанность по аппаратным и программным защитам

Особенности проекта

- Написан на DOT языке → легко поддерживать в Git
- Не охватывает LSM и userspace-защиты
- Лицензия: GPL v3. Последняя версия: v6.10
- Репозиторий:

https://github.com/a13xp0p0v/linux-kernel-defence-mapCodeberg

https://codeberg.org/a13xp0p0v/linux-kernel-defence-map

https://gitflic.ru/project/a13xp0p0v/linux-kernel-defence-map

Источники вдохновения:

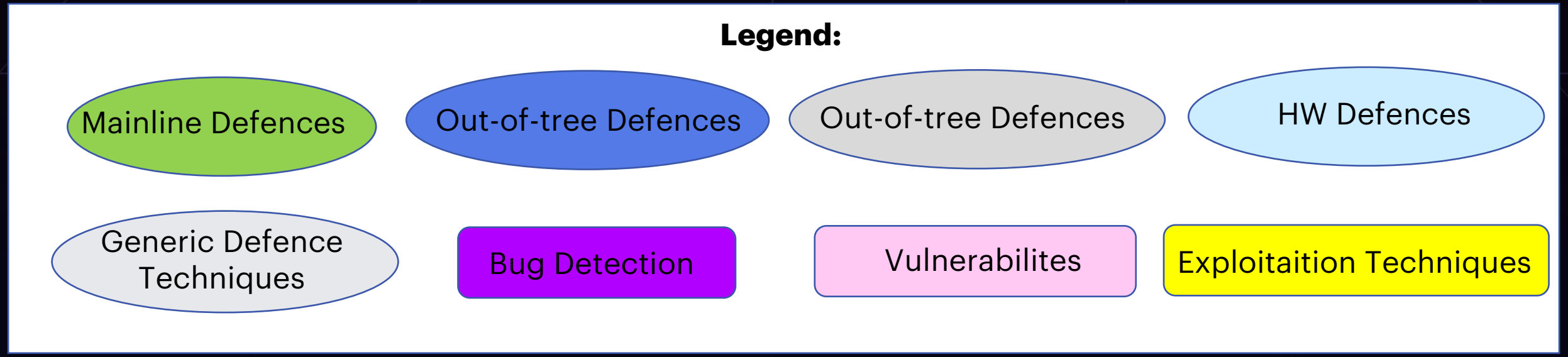
- Документация Grsecurity
- Kees Cook: Kernel Self-Protection Project
- Документация Linux kernel security
- CWE и исследовательские статьи

Linux Kernel Defence Map (LKDM)

БЕКОН

Ключевые понятия карты

- Vulnerability classes — классы уязвимостей
- Exploitation techniques — методы эксплуатации
- Bug detection mechanisms — механизмы обнаружения ошибок
- Defence technologies — технологии защиты (в ядре и вне его)
- Связи между узлами не означают полной защиты, а только наличие отношения
- Карта не затрагивает способы уменьшения поверхности атак
- На схеме ключевые понятия:



kernel-hardening-checker

БЕКОН

Инструмент для автоматизированной проверки параметров безопасности ядра Linux

ЧТО УМЕЕТ ИНСТРУМЕНТ?

- Проверять настройки ядра Linux на наличие включенной защиты
- Удобно анализировать системные конфигурации
- Сравнивать настройки с рекомендациями KSPP, Grsecurity, GrapheneOS, CLIP OS

GitHub: kernel-hardening-checker

kernel-hardening-checker

БЕКОН

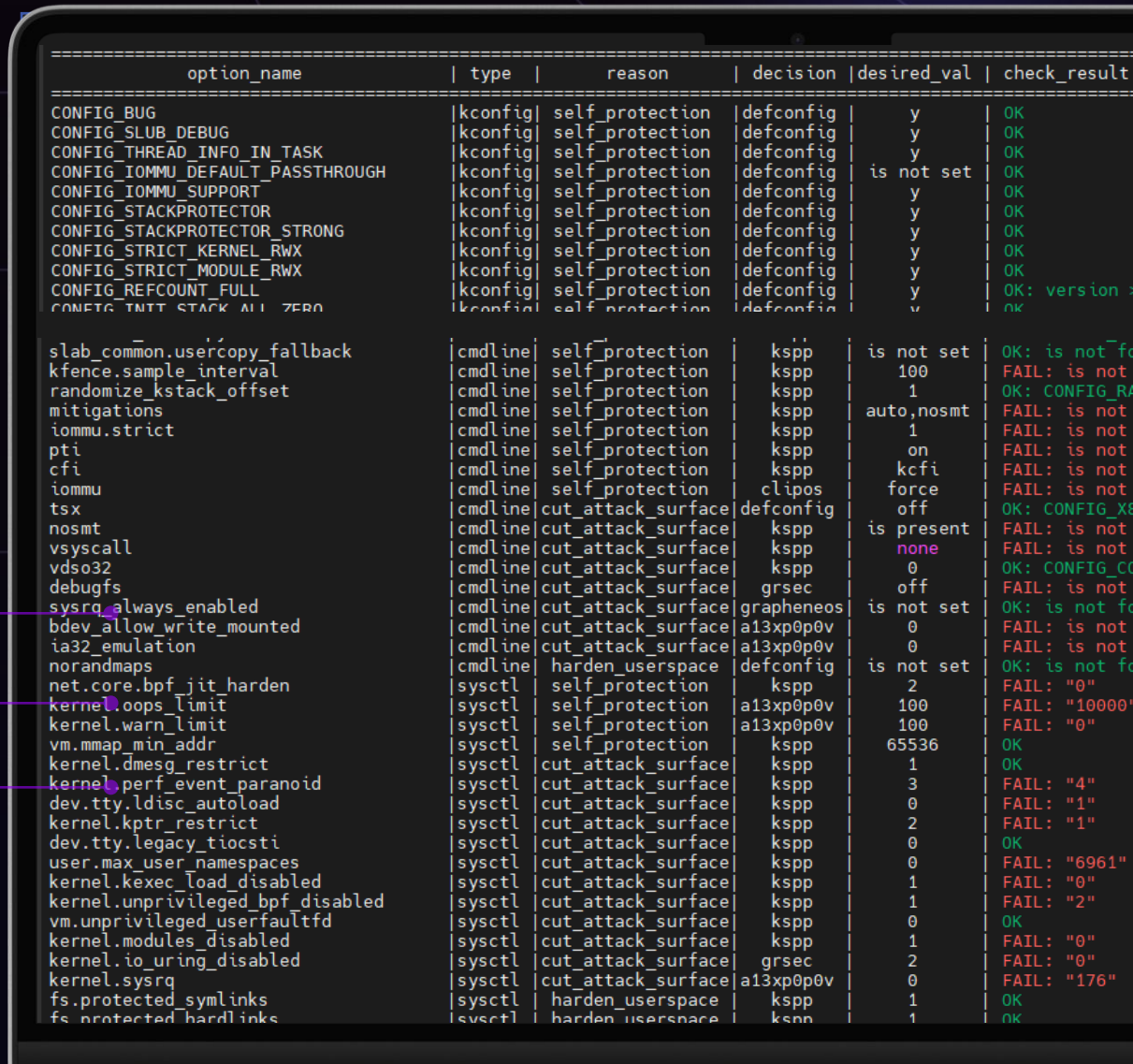
Инструмент для автоматизированной проверки параметров безопасности ядра Linux

ЧТО УМЕЕТ ИНСТРУМЕНТ?

- Проверять настройки ядра Linux на наличие включенной защиты
- Удобно анализировать системные конфигурации
- Сравнивать настройки с рекомендациями KSPP, Grsecurity, GrapheneOS, CLIP OS

КАКИЕ ПАРАМЕТРЫ ЯДРА ПРОВЕРЯЕТ?

- Во время компиляции: **Kconfig**
- На этапе загрузки: **kernel cmdline**
- В рантайме: **sysctl**



option_name	type	reason	decision	desired_val	check_result
CONFIG_BUG	kconfig	self_protection	defconfig	y	OK
CONFIG_SLUB_DEBUG	kconfig	self_protection	defconfig	y	OK
CONFIG_THREAD_INFO_IN_TASK	kconfig	self_protection	defconfig	y	OK
CONFIG_IOMMU_DEFAULT_PASSTHROUGH	kconfig	self_protection	defconfig	is not set	OK
CONFIG_IOMMU_SUPPORT	kconfig	self_protection	defconfig	y	OK
CONFIG_STACKPROTECTOR	kconfig	self_protection	defconfig	y	OK
CONFIG_STACKPROTECTOR_STRONG	kconfig	self_protection	defconfig	y	OK
CONFIG_STRICT_KERNEL_RWX	kconfig	self_protection	defconfig	y	OK
CONFIG_STRICT_MODULE_RWX	kconfig	self_protection	defconfig	y	OK
CONFIG_REFCOUNT_FULL	kconfig	self_protection	defconfig	y	OK: version
slab_common.usercopy_fallback	cmdline	self_protection	kspp	is not set	OK: is not fo
kfence.sample_interval	cmdline	self_protection	kspp	100	FAIL: is not
randomize_kstack_offset	cmdline	self_protection	kspp	1	OK: CONFIG_RA
mitigations	cmdline	self_protection	kspp	auto,nosmt	FAIL: is not
iommu.strict	cmdline	self_protection	kspp	1	FAIL: is not
pti	cmdline	self_protection	kspp	on	FAIL: is not
cfi	cmdline	self_protection	kspp	kcfi	FAIL: is not
iommu	cmdline	self_protection	clipos	force	FAIL: is not
tsx	cmdline	cut_attack_surface	defconfig	off	OK: CONFIG_X8
nosmt	cmdline	cut_attack_surface	kspp	is present	FAIL: is not
vsyscall	cmdline	cut_attack_surface	kspp	none	FAIL: is not
vdso32	cmdline	cut_attack_surface	kspp	0	OK: CONFIG_CO
debugfs	cmdline	cut_attack_surface	grsec	off	FAIL: is not
sysrq_always_enabled	cmdline	cut_attack_surface	grapheneos	is not set	OK: is not fo
bdev_allow_write_mounted	cmdline	cut_attack_surface	a13xp0p0v	0	FAIL: is not
ia32_emulation	cmdline	cut_attack_surface	a13xp0p0v	0	FAIL: is not
norandmaps	cmdline	harden_userspace	defconfig	is not set	OK: is not fo
net.core.bpf_jit_harden	sysctl	self_protection	kspp	2	FAIL: "0"
kernel.oops_limit	sysctl	self_protection	a13xp0p0v	100	FAIL: "10000"
kernel.warn_limit	sysctl	self_protection	a13xp0p0v	100	FAIL: "0"
vm.mmap_min_addr	sysctl	self_protection	kspp	65536	OK
kernel.dmesg_restrict	sysctl	cut_attack_surface	kspp	1	OK
kernel.perf_event_paranoid	sysctl	cut_attack_surface	kspp	3	FAIL: "4"
dev.tty.ldisc_autoload	sysctl	cut_attack_surface	kspp	0	FAIL: "1"
kernel.kptr_restrict	sysctl	cut_attack_surface	kspp	2	FAIL: "1"
dev.tty.legacy_tiocsti	sysctl	cut_attack_surface	kspp	0	OK
user.max_user_namespaces	sysctl	cut_attack_surface	kspp	0	FAIL: "6961"
kernel.kexec_load_disabled	sysctl	cut_attack_surface	kspp	1	FAIL: "0"
kernel.unprivileged_bpf_disabled	sysctl	cut_attack_surface	kspp	1	FAIL: "2"
vm.unprivileged_userfaultfd	sysctl	cut_attack_surface	kspp	0	OK
kernel.modules_disabled	sysctl	cut_attack_surface	kspp	1	FAIL: "0"
kernel.io_uring_disabled	sysctl	cut_attack_surface	grsec	2	FAIL: "0"
kernel.sysrq	sysctl	cut_attack_surface	a13xp0p0v	0	FAIL: "176"
fs.protected_symlinks	sysctl	harden_userspace	kspp	1	OK

БЕКОН

Linux Kernel Self Protection Project

LKSPP

Linux Kernel Self Protection Project (LKSPP)

БЕКОН

Цель:

- Защитить ядро Linux от уязвимостей, даже если они ещё не найдены.
- Предотвратить классы ошибок, а не просто исправлять баги.
- Сделать ядро устойчивым к эксплуатации уязвимостей, даже в случае локального или привилегированного атакующего.

https://kspp.github.io/Recommended_Settings.html#:~:text=,max_user_namespaces%20%3D%200

Linux Kernel Self Protection Project (LKSPP)

БЕКОН

Цель:

- Защитить ядро Linux от уязвимостей, даже если они ещё не найдены.
- Предотвратить классы ошибок, а не просто исправлять баги.
- Сделать ядро устойчивым к эксплуатации уязвимостей, даже в случае локального или привилегированного атакующего.

Особенности проекта

- Self-Protection: внутренняя защита ядра от атак.
- Hardening: снижение риска эксплуатации через ограничения в памяти, правах и API.
- Elimination of Classes: удаление целых категорий уязвимостей (например, stack overflow).
- No Developer Opt-In: защита активна по умолчанию, покрывает всё ядро (включая внешние модули).
- Проактивный подход: не ждать атак, а предотвращать их заранее

https://kspp.github.io/Recommended_Settings.html#:~:text=,max_user_namespaces%20%3D%200

Linux Kernel Self Protection Project (LKSPP)

БЕКОН

Цель:

- Защитить ядро Linux от уязвимостей, даже если они ещё не найдены.
- Предотвратить классы ошибок, а не просто исправлять баги.
- Сделать ядро устойчивым к эксплуатации уязвимостей, даже в случае локального или привилегированного атакующего.

Особенности проекта

- Self-Protection: внутренняя защита ядра от атак.
- Hardening: снижение риска эксплуатации через ограничения в памяти, правах и API.
- Elimination of Classes: удаление целых категорий уязвимостей (например, stack overflow).
- No Developer Opt-In: защита активна по умолчанию, покрывает всё ядро (включая внешние модули).
- Проактивный подход: не ждать атак, а предотвращать их заранее

Снижение поверхности атаки:

- Запрет на доступ ядра к памяти userspace без необходимости.
- Защита от модификации чувствительных структур (GDT, IDT, sys_call_table).
- Отключение устаревших или опасных API (compat, kexec, user namespaces).
- Обязательное обнуление или «отравление» памяти при её освобождении.

https://kspp.github.io/Recommended_Settings.html#:~:text=,max_user_namespaces%20%3D%200

БЕКОН

Sysctls

редактирование файла
/etc/sysctl.conf

```
kernel.kptr_restrict = 2
kernel.modules_disabled = 1
kernel.perf_event_paranoid = 3
kernel.randomize_va_space = 2
```

БЕКОН

Sysctls

редактирование файла
/etc/sysctl.conf

```
kernel.kptr_restrict = 2
kernel.modules_disabled = 1
kernel.perf_event_paranoid = 3
kernel.randomize_va_space = 2
```

CONFIGs

CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT
CONFIG_HARDENED_USERCOPY
CONFIG_INIT_ON_ALLOC_DEFAULT_ON
CONFIG_STACKPROTECTOR

при компиляции ядра задаются следующие конфигурации и опции безопасности

https://kspp.github.io/Recommended_Settings

Kernel space

ИНИЦИАЛИЗАЦИЯ И РАНДОМИЗАЦИЯ ПАМЯТИ

обнуление памяти при выделении освобождении	init_on_alloc	1
	init_on_free	1
случайное размещение страниц и стека	page_alloc.shuffle	1
	randomize_kstack_offset	on

Kernel space

БЕКОН

ИНИЦИАЛИЗАЦИЯ И РАНДОМИЗАЦИЯ ПАМЯТИ

обнуление памяти при выделении освобождении	init_on_alloc	1
	init_on_free	1
случайное размещение страниц и стека	page_alloc.shuffle	1
	randomize_kstack_offset	on

КОНТРОЛЬ ДОСТУПА И ИЗОЛЯЦИЯ

запрет загрузки модулей	kernel.modules_disabled	1
отключение kexec	kernel.kexec_load_disabled	1
запрет на user namespaces	user.max_user_namespaces	0
блокировка ptrace для не-предков	kernel.yama.ptrace_scope	3
запрет eBPF для обычных пользователей	kernel.unprivileged_bpf_disabled	1

Kernel space

БЕКОН

ИНИЦИАЛИЗАЦИЯ И РАНДОМИЗАЦИЯ ПАМЯТИ

обнуление памяти при выделении освобождении	init_on_alloc	1
	init_on_free	1
случайное размещение страниц и стека	page_alloc.shuffle	1
	randomize_kstack_offset	on

СКРЫТИЕ АДРЕСОВ ЯДРА

защита информации в /proc и dmesg	kernel.kexec_load_disabled	1

КОНТРОЛЬ ДОСТУПА И ИЗОЛЯЦИЯ

запрет загрузки модулей	kernel.modules_disabled	1
отключение kexec	kernel.kexec_load_disabled	1
запрет на user namespaces	user.max_user_namespaces	0
блокировка ptrace для не-предков	kernel.yama.ptrace_scope	3
запрет eBF для обычных пользователей	kernel.unprivileged_bpf_disabled	1

БЕКОН

Kernel space

ИНИЦИАЛИЗАЦИЯ И РАНДОМИЗАЦИЯ ПАМЯТИ

обнуление памяти при выделении освобождении	init_on_alloc	1
	init_on_free	1
случайное размещение страниц и стека	page_alloc.shuffle	1
	randomize_kstack_offset	on

СКРЫТИЕ АДРЕСОВ ЯДРА

защита информации в /proc и dmesg	kernel.kexec_load_disabled	1

КОНТРОЛЬ НАД ФАЙЛОВОЙ СИСТЕМОЙ

защита от атак через ссылки	fs.protected_symlinks	1
	fs.protected_hardlinks	1
усиленная проверка при создании объектов	fs.protected_fifos	2
	fs.protected_regular	2
запрет дампа привилегированных процессов	fs.suid_dumpable	0

КОНТРОЛЬ ДОСТУПА И ИЗОЛЯЦИЯ

запрет загрузки модулей	kernel.modules_disabled	1
отключение kexec	kernel.kexec_load_disabled	1
запрет на user namespaces	user.max_user_namespaces	0
блокировка ptrace для не-предков	kernel.yama.ptrace_scope	3
запрет eBF для обычных пользователей	kernel.unprivileged_bpf_disabled	1

БЕКОН

Kernel space

ИНИЦИАЛИЗАЦИЯ И РАНДОМИЗАЦИЯ ПАМЯТИ

обнуление памяти при выделении освобождении	init_on_alloc	1
	init_on_free	1
случайное размещение страниц и стека	page_alloc.shuffle	1
	randomize_kstack_offset	on

СКРЫТИЕ АДРЕСОВ ЯДРА

защита информации в /proc и dmesg	kernel.kexec_load_disabled	1

КОНТРОЛЬ НАД ФАЙЛОВОЙ СИСТЕМОЙ

защита от атак через ссылки	fs.protected_symlinks	1
	fs.protected_hardlinks	1
усиленная проверка при создании объектов	fs.protected_fifos	2
	fs.protected_regular	2
запрет дампа привилегированных процессов	fs.suid_dumpable	0

КОНТРОЛЬ ДОСТУПА И ИЗОЛЯЦИЯ

запрет загрузки модулей	kernel.modules_disabled	1
отключение kexec	kernel.kexec_load_disabled	1
запрет на user namespaces	user.max_user_namespaces	0
блокировка ptrace для не-предков	kernel.yama.ptrace_scope	3
запрет eBF для обычных пользователей	kernel.unprivileged_bpf_disabled	1

ДОПОЛНИТЕЛЬНЫЕ МЕРЫ

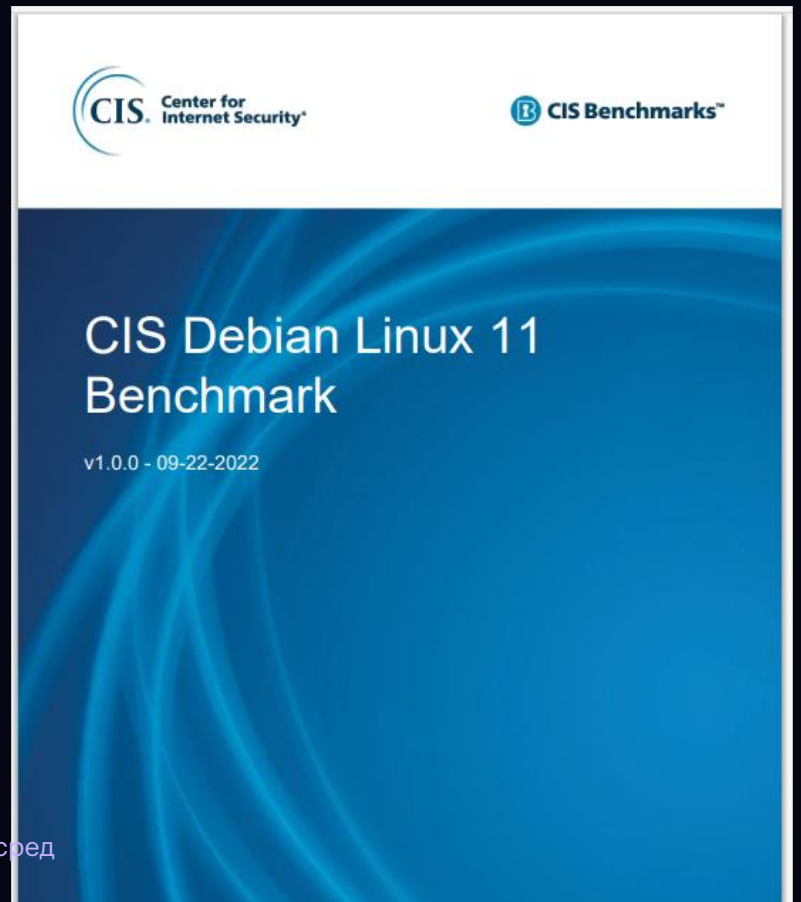
защита при копировании между user- и kernel-space	hardened_usercopy	1
защита JIT-компиляции BPF	net.core.bpf_jit_harden	2
запрет профилирования	user.max_user_namespaces	3
немедленная перезагрузка при ошибках	kernel.warn_limit kernel.oops_limit	1 1

CIS Benchmark

БЕКОН

Лучшие практики, гайдлайны и рекомендации по безопасности

Регулярное обновление для отражения новых угроз и технологий



Обеспечение устойчивости и улучшение безопасности ОС

140+

Benhcmark документов

CIS Benchmark

БЕКОН

USER space

1.2
Package Management
GPG, apt, yum только с доверенных репозиториев

1.4
Настройка загрузчика
Защита GRUB (/boot/grub2/grub.cfg, set superusers)

2.1-2.2
Services
отключить FTP, NFS (systemctl disable), оставить только нужное

2.4
Job Schedulers
запрет cron/at обычным пользователям (/etc/cron.deny)

5.1
SSH
PermitRootLogin no, MaxAuthTries 3, Banner /etc/issue.net

5.2
sudo/su
requiretty
logfile=/var/log/sudo.log

5.3
PAM
pam_pwquality
pam_faillock

Users
пароли
учетные записи

6.1-6.3
Logging & Auditing
централизованное логирование
audit.rules

KERNEL space

CIS Benchmark

БЕКОН

USER space

1.2 Package Management
GPG, apt, yum только
с доверенных репозиториев

1.4 Настройка загрузчика
Защита GRUB
(/boot/grub2/grub.cfg, set superusers)

2.1-2.2 Services
отключить FTP, NFS
(systemctl disable),
оставить только нужное

2.4 Job Schedulers
запрет cron/at обычным пользователям
(/etc/cron.deny)

5.1 SSH
PermitRootLogin no,
MaxAuthTries 3, Banner
/etc/issue.net

5.2 sudo/su
requiretty
logfile=/var/log/sudo.log

5.3 PAM
pam_pwquality
pam_faillock

Users
пароли
учетные записи

6.1-6.3 Logging & Auditing
централизованное
логирование
audit.rules

KERNEL space

1.3 Mandatory Access Control
SELinux/AppArmor

1.5 Process Hardening
kernel.randomize_va_space = 2
kernel.yama.ptrace_scope = 1
fs.suid_dumpable = 0

3.1-3.2 Настройка сетевых параметров
Отключение ненужных
сетевых модулей
IPv6 is disabled

3.3 Настройка сетевых параметров ядра
net.ipv4.ip_forward = 0
send_redirects = 0
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

4 Firewall
net. ufw, nftables, iptables

CIS Benchmark

БЕКОН

USER space

1.2
Package Management
GPG, apt, yum только с доверенных репозиториев

1.4
Настройка загрузчика
Защита GRUB (/boot/grub2/grub.cfg, set superusers)

2.1-2.2
Services
отключить FTP, NFS (systemctl disable), оставить только нужное

2.4
Job Schedulers
запрет cron/at обычным пользователям (/etc/cron.deny)

5.1
SSH
PermitRootLogin no, MaxAuthTries 3, Banner /etc/issue.net

5.2
sudo/su
requiretty
logfile=/var/log/sudo.log

5.3
PAM
pam_pwquality
pam_faillock

Users
пароли
учетные записи

6.1-6.3
Logging & Auditing
централизованное логирование
audit.rules

KERNEL space

1.3
Mandatory Access Control
SELinux/AppArmor

1.5
Process Hardening
kernel.randomize_va_space = 2
kernel.yama.ptrace_scope = 1
fs.suid_dumpable = 0

3.1-3.2
Настройка сетевых параметров
Отключение ненужных сетевых модулей
IPv6 is disabled

DNS-туннелирование через IPv6

3.3
Настройка сетевых параметров ядра
net.ipv4.ip_forward = 0
send_redirects = 0
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

4
Firewall
net. ufw, nftables, iptables

Аудит безопасности Ubuntu: Lynis по CIS Benchmark

БЕКОН

Выявлены отклонения от рекомендаций CIS Benchmark, включая включённую пересылку IP-пакетов и небезопасные настройки sysctl

Итоговый hardening index демонстрирует, что системе требуется дополнительная настройка для повышения уровня защиты.

```
[+] Kernel Hardening
------------------------------------
  - Comparing sysctl key pairs with scan profile
    - dev.tty.ldisc_autoload (exp: 0)                    [ DIFFERENT ]
    - fs.protected_fifos (exp: 2)                        [ DIFFERENT ]
    - fs.protected_hardlinks (exp: 1)                    [ OK ]
    - fs.protected_regular (exp: 2)                      [ OK ]
    - fs.protected_symlinks (exp: 1)                     [ OK ]
    - fs.suid_dumpable (exp: 0)                          [ DIFFERENT ]
    - kernel.core_uses_pid (exp: 1)                      [ DIFFERENT ]
    - kernel.ctrl-alt-del (exp: 0)                       [ OK ]
    - kernel.dmesg_restrict (exp: 1)                     [ OK ]
    - kernel.kptr_restrict (exp: 2)                      [ DIFFERENT ]
    - kernel.modules_disabled (exp: 1)                   [ DIFFERENT ]
    - kernel.perf_event_paranoid (exp: 3)                [ DIFFERENT ]
    - kernel.randomize_va_space (exp: 2)                 [ OK ]
    - kernel.sysrq (exp: 0)                              [ DIFFERENT ]
    - kernel.unprivileged_bpf_disabled (exp: 1)          [ DIFFERENT ]
    - kernel.yama.ptrace_scope (exp: 1 2 3)              [ OK ]
    - net.core.bpf_jit_harden (exp: 2)                   [ DIFFERENT ]
    - net.ipv4.conf.all.accept_redirects (exp: 0)        [ DIFFERENT ]
    - net.ipv4.conf.all.accept_source_route (exp: 0)     [ OK ]
    - net.ipv4.conf.all.bootp_relay (exp: 0)             [ OK ]
    - net.ipv4.conf.all.forwarding (exp: 0)              [ OK ]
    - net.ipv4.conf.all.log_martians (exp: 1)            [ DIFFERENT ]
    - net.ipv4.conf.all.mc_forwarding (exp: 0)           [ OK ]
    - net.ipv4.conf.all.proxy_arp (exp: 0)               [ OK ]
    - net.ipv4.conf.all.rp_filter (exp: 1)               [ DIFFERENT ]
    - net.ipv4.conf.all.send_redirects (exp: 0)          [ DIFFERENT ]
    - net.ipv4.conf.default.accept_redirects (exp: 0)    [ DIFFERENT ]
    - net.ipv4.conf.default.accept_source_route (exp: 0) [ DIFFERENT ]
    - net.ipv4.conf.default.log_martians (exp: 1)        [ DIFFERENT ]
```

БЕКОН

CIS Benchmark

Linux

vs.

KSPP

CIS (Ubuntu) vs LKSP

БЕКОН

	CIS Benchmark	Kernel Self-Protection
Sysctl (ядро)	**kernel.kptr_restrict ≥1** скрыть адреса ядра от непривилегированных пользователей	**kernel.kptr_restrict=2** максимальное скрытие адресов ядра
	kernel.dmesg_restrict=1 скрыть буфер ядра от непривилегированных пользователей	**kernel.dmesg_restrict=1** закрыть доступ к dmesg
	kernel.perf_event_paranoid ≥3 ограничить unprivilege-профилирование	**kernel.perf_event_paranoid=3** строгий режим профилирования
	kernel.randomize_va_space=2 включить ASLR	**kernel.randomize_va_space=2** полная рандомизация адресного пространства
	kernel.yama.ptrace_scope=2 ptrace только от родителя	**kernel.yama.ptrace_scope=3** запрет ptrace вообще
		kernel.unprivileged_bpf_disabled=1 запрет eBPF кто не root

CIS (Ubuntu) vs LKSP

БЕКОН

	CIS Benchmark	Kernel Self-Protection
Sysctl (ядро)	в CIS нет специфических данных параметров ядра	**hardened_usercopy = 1** проверяет копирование памяти между ядром и пользователем (блокирует выход за границы буфера при copy_{to,from}_user, предотвращая heap-overflow) **randomize_kstack_offset = on** рандомизирует смещение ядра-стека (≈5 бит энтропии, усложняет атаки через переполнение стека ядра) **pti = on — включает Page Table Isolation** разделение адресного пространства пользователя и ядра (защита от Spectre/Meltdown, блокирует утечки ядра) **nosmt** отключает SMT/гиперпоточность (уменьшает утечки через схемы кэш-сайд-каналов, повышает изоляцию ядра) **vsyscall = none** отключает устаревший механизм vsyscall access.redhat.com (устраняет фиксированные адреса для быстрых вызовов, которые могли быть)

CIS (Ubuntu) vs LKSP

БЕКОН

Сеть

CIS Benchmark	Kernel Self-Protection
net.ipv4.ip_forward = 0 отключает пересылку IPv4-пакетов	
net.ipv4.conf.all.accept_source_route = 0 запрещают прием пакетов со «source routing» (предотвращает обход маршрутов злоумышленниками)	
net.ipv4.conf.all.accept_redirects = 0 отключают принятие ICMP-редиректов (не позволяют внешним узлам изменять таблицу маршрутизации)	LKSPP параметры настройки сети отсутствуют
net.ipv4.icmp_echo_ignore_broadcasts = 1 — игнорирует широковещательные ICMP-эхо (защита от ICMP-флуда)	
net.ipv4.conf.all.rp_filter = 1 включает обратную фильтрацию путей (reverse path)	
net.ipv4.tcp_syncookies = 1 включает SYN-cookies	

CIS (Ubuntu) vs LKSP

БЕКОН

	CIS Benchmark	Kernel Self-Protection
auditd и аудит событий	**audit=1** установлен и включён auditd	
	Настроены правила аудита критичных событий (логины, sudo, загрузка модулей и т.д.)	LKSPP не задаёт конкретных настроек auditd. (Наличие auditd – часть ОС, не KSP.)
	Immutable конфигурация аудита неизменяемая	

CIS (Ubuntu) vs LKSP

БЕКОН

	CIS Benchmark	Kernel Self-Protection
доступ аутентификация	**PermitRootLogin no** требует аутентификации по ключам/паролям под	LKSPP не затрагивает SSH, т.к. это уровень ядра. Защита SSH – задача ОС/сервиса (CIS)
	AllowUsers/AllowGroups или DenyUsers/DenyGroups Лимит доступа	
	LogLevel INFO включён лог. фиксировать входы	
	X11Forwarding no, **AllowAgentForwarding no, HostbasedAuthentication no, PermitEmptyPasswords no** Отключены неиспользуемые функции/закрыты доступа удаленного канала	

БЕКОН

ПРИМЕРЫ МИТИГАЦИИ АТАК

CVE-2022-0185

fs_context CVSS: ~8.4

БЕКОН

Переполнение heap overflow в подсистеме файловых контекстов (fs_context) ядра Linux.
Обход изоляции контейнера через манипуляции с файловыми контекстами

↓

Условия эксплуатации:

- Используется устаревший (legacy) режим монтирования
- Наличие CAP_SYS_ADMIN в контейнере
- Доступ к unshare создания новых namespace

↓

Побег из контейнера

Выполнение произвольного кода от root на хосте

https://www.opennet.ru/opennews/art.shtml?num=56556

CVE-2022-0185

БЕКОН

Категория	LKSPP	CIS
Анализ/ утечеки	kernel.yama.ptrace_scope=3 параноидальный уровень запрет ptrace kernel.randomize_va_space=2 рандомизации расположения адресного пространства (ASLR) kernel.kptr_restrict=2 скрытие адресов ядра	kernel.yama.ptrace_scope=1>3 ограничивает ptrace текущим пользователем и потомками kernel.randomize_va_space=2 рандомизации расположения адресного пространства (ASLR)
Эскалация	user.max_user_namespaces=0 запрет user namespaces (блокировка unshare) kernel.modules_disabled=1 запрет загрузки модулей	SELinux в рекомендованной конфигурации CIS блокируют чтение/запись контейнером файлов Даже если у процесса есть CAP_SYS_ADMIN, SELinux может запретить ему монтировать, загружать модули, обращаться к хосту

CVE-2022-0492

cgroup release_agent bypass

CVSS: ~6.9

БЕКОН

Уязвимость в подсистеме cgroup v1 в ядре Linux (release_agent bypass)

↓

Условия эксплуатации:

- Наличие CAP_SYS_ADMIN
- root в контейнере
- доступ к cgroup v1
- Нет apparmor/seccomp/selinux для блокировки системных вызовов

↓

Выполнение произвольного кода от root на хосте

https://xmlisse.wordpress.com/2023/10/09/container-security/
https://unit42.paloaltonetworks.com/cve-2022-0492-cgroups/

CVE-2022-0492

БЕКОН

Категория	LKSPP	CIS
Анализ/ утечеки	kernel.yama.ptrace_scope=3 параноидальный уровень запрет ptrace kernel.randomize_va_space=2 рандомизации расположения адресного пространства (ASLR) kernel.kptr_restrict=2 скрытие адресов ядра	kernel.yama.ptrace_scope=1>3 ограничивает ptrace текущим пользователем и потомками kernel.randomize_va_space=2 рандомизации расположения адресного пространства (ASLR)
Эскалация	user.max_user_namespaces=0 запрет user namespaces (блокировка unshare) kernel.modules_disabled=1 блокировка загрузки руткитов/модулей	SELinux в рекомендованной конфигурации CIS блокируют чтение/запись контейнером файлов Даже если у процесса есть CAP_SYS_ADMIN, SELinux может запретить ему монтировать, загружать модули, обращаться к хосту

CVE-2021-3490

eBPF CVSS: ~7.2

БЕКОН

Ошибка в верификаторе eBPF-программ позволяет обходить ограничения и выполнять произвольное чтение и запись в памяти ядра (kernel R/W primitive)

↓

Условия эксплуатации:

- Контейнер с доступом к системному вызову bpf(), через CAP_BPF или CAP_SYS_ADMIN
- Отсутствие seccomp-профиля, блокирующего bpf()

↓

Повышение привилегий до root в контейнере
доступ к файловой системе хоста

Полноценный container escape и выполнение команд от имени root на хосте

https://www.crowdstrike.com/en-us/blog/exploiting-cve-2021-3490-for-container-escapes/

CVE-2021-3490

БЕКОН

Категория	LKSPP	CIS
Анализ/ утечеки	kernel.yama.ptrace_scope=3 параноидальный уровень запрет ptrace kernel.randomize_va_space=2 рандомизации расположения адресного пространства (ASLR) kernel.kptr_restrict=2 скрытие адресов ядра	kernel.yama.ptrace_scope=1>3 ограничивает ptrace текущим пользователем и потомками kernel.randomize_va_space=2 рандомизации расположения адресного пространства (ASLR)
Эскалация	kernel.modules_disabled=1 блокировка загрузки руткитов/модулей kernel.unprivileged_bpf_disabled=1 мешает использованию eBPF после побега — ограничивает постэксплуатацию	SELinux в рекомендованной конфигурации CIS ограничивает чтение/запись контейнером файлов Даже если у процесса есть CAP_SYS_ADMIN и CAP_BPF, SELinux может запретить ему монтировать, загружать модули, обращаться к хосту

LKSPP и CIS Benchmark — не абсолютная защита, а только часть общей стратегии

БЕКОН

❌ Не защищают от:

Прямого побега из привилегированного контейнера, особенно с флагом --privileged, возможностями CAP_SYS_ADMIN, CAP_SYS_MODULE доступ к hostPath

Эксплуатации ошибок в рантаймах Kubernetes или Docker. Уязвимости в среде выполнения контейнера (например, в механизме контрольных групп (cgroup)) могут быть использованы для побега

Действий привилегированного процесса внутри контейнера, если он уже запущен с правами root

LKSPP и CIS Benchmark — не абсолютная защита, а только часть общей стратегии

БЕКОН

✅ Но помогают:

Ограничить последствия побега (запретить загрузку модулей, скрыть адреса ядра, отключить ptrace)

Снизить вероятность успешной эксплуатации некоторых уязвимостей ядра:

user.max_user_namespaces=0 – защита от CVE-2022-0185, CVE-2022-0492

kernel.unprivileged_bpf_disabled=1 – защита от CVE-2021-3490

kernel.modules_disabled=1 – запрет на загрузку backdoor-ядерных модулей

Дополнительные механизмы безопасности AppArmor/SELinux

БЕКОН

ЛУЧШИЕ ПРАКТИКИ ПРИМЕНЕНИЯ ПАРАМЕТРОВ ХАРДЕНИНГА

или

как положить сервис

Примеры параметров и механизмов влияния

БЕКОН

kernel.kptr_restrict=1/2

Цель:
Скрыть адреса ядра (значение 1 или 2): из контейнера нельзя прочитать символы ядра через /proc

Риски:
Инструменты профилирования/трассировки (perf, flame graphs) в контейнере перестают отображать символы ядра

Затронутые сервисы:
Инструменты профилирования/мониторинга (perf, Sysdig, Falco)

https://stackoverflow.com/questions/21284906/perf-couldnt-record-kernel-reference-relocation-symbol#:~:text=Kernel%20address%20maps%20%28%2Fproc%2F,can%27t%20be%20resolved%20as%20well

Примеры параметров и механизмов влияния

БЕКОН

kernel.unprivileged_bpf_disabled=1

Цель:
При установке значения 1 запрещает непривилегированным процессам использовать системный вызов bpf()

Риски:
Процессы без CAP_BPF/CAP_SYS_ADMIN не могут загружать eBPF-программы.
Контейнеры без специальных прав (например, Docker без CAP_BPF) не смогут использовать eBPF
XDPFail2ban и другие сетевые фильтры — не работают без привилегий

Затронутые сервисы:
Cilium (BPF-сеть), Falco, Sysdig Secure (при отсутствии CAP_BPF)

https://docs.cilium.io/en/latest/reference-guides/bpf/architecture/

Примеры параметров и механизмов влияния

БЕКОН

user.max_user_namespaces=0

Цель:
Значение 0 запрещает создание пользовательских неймспейсов

Риски:
Нарушает работу rootless контейнеров
Для непривилегированных пользователей может нарушаться работа сетевых инструментов, использующих сетевые пространства имен (network namespaces)
Старые версии bpftrace и bcc-tools, BPF программы, зависящие от фиксированных адресов, перестают работать
Контейнеры (Kubernetes/Docker по умолчанию) не затрагиваются

Затронутые сервисы:
Rootless-контейнеры (rootless Docker/Podman) и sandbox-приложения
Старые версии bpftrace и bcc-tools

https://www.debian.org/releases/bullseye/amd64/release-notes/ch-information.en.html#linux-user-namespaces#:~:text=user

https://wiki.astralinux.ru/pages/viewpage.action?pageId=348164331

Примеры параметров и механизмов влияния

БЕКОН

net.ipv4.ip_forward=0

Цель:
При 0 отключается пересылка IPv4 пакетов на хосте

Риски:
В Kubernetes это отключит сетевую связь подов (нет маршрутизации между узлами), в Docker нарушит NAT. Для работы сетей контейнеров (CNI/Docker bridge) значение должно быть 1

Затронутые сервисы:
Сетевое взаимодействие контейнеров: Kubernetes (Calico, Flannel) и Docker сети

https://www.suse.com/support/kb/doc/?id=000020166#:~:text=If%20the%20sysctl%20,prevent%20Pod%20networking%20from%20functioning

https://docs.cilium.io/en/latest/network/concepts/routing/

Примеры параметров и механизмов влияния

БЕКОН

net.ipv4.ip_forward=0

Цель:
При 0 отключается пересылка IPv4 пакетов на хосте

Риски:
В Kubernetes это отключит сетевую связь подов (нет маршрутизации между узлами), в Docker нарушит NAT.
Для работы сетей контейнеров (CNI/Docker bridge) значение должно быть 1

Cilium не использует bridge и его работа не зависит напрямую от net.ipv4.ip_forward = 1, в отличие от Docker или CNI-плагинов вроде Flannel

Затронутые сервисы:
Сетевое взаимодействие контейнеров: Kubernetes (Calico, Flannel) и Docker сети

https://www.suse.com/support/kb/doc/?id=000020166#:~:text=If%20the%20sysctl%20,prevent%20Pod%20networking%20from%20functioning

https://docs.cilium.io/en/latest/network/concepts/routing/

Примеры параметров и механизмов влияния

БЕКОН

kernel.modules_disabled=1

Цель:
При значении 1 полностью отключает загрузку модулей

Риски:
Falco (использующий по умолчанию LKM-демон) и другие инструменты на основе модулей перестанут работать
eBPF-программы

Затронутые сервисы:
Falco (режим с яд. модулем), другие LKM-инструменты
eBPF-программы (Cilium) сами по себе не затрагиваются, если были загружены заранее

https://dfir.ch/posts/today_i_learned_lkm_kernel.modules_disabled/#:~:text=

Примеры параметров и механизмов влияния

БЕКОН

net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

Цель:
Reverse Path Filtering — фильтрация по обратному маршруту. Защита от IP spoofing и атак на маршрутизацию

Риски:
Проблемы с VPN, WireGuard, CNI
Прерывание связи при сложной маршрутизации (overlay, VXLAN)

Затронутые сервисы:
Нарушает работу CNI-плагины с policy routing (Calico, Cilium)
Контейнеров с host или bridge сетями

https://dfir.ch/posts/today_i_learned_lkm_kernel.modules_disabled/#:~:text=

Примеры параметров и механизмов влияния

БЕКОН

CONFIG_DEBUG_INFO_BTF = is not set

Цель:
Необходима для продвинутой работы с **eBPF** (Extended BPF)

Влияние:
- Повышает совместимость eBPF-программ и наблюдаемость ядра
- Необходим для многих инструментов мониторинга и безопасности

Риски:
❌ Увеличивает размер ядра, может потенциально раскрыть структуру ядра злоумышленнику при наличии доступа

Рекомендации:
- На монолитных серверах для минимизации поверхности атаки — отключать
- На нодах наблюдаемости, где работают Falco, Cilium, и пр. — включать

Вывод

БЕКОН

LKSPP: защита ядра "по умолчанию", устраняет целые классы уязвимостей — идеальна для серверов и критичных узлов.

CIS Benchmark: комплексный гайд, охватывающий как ядро, так и userspace — применим шире, но менее глубок по части hardening ядра.

Вместе — сильнее: совместное применение LKSPP и CIS даёт баланс безопасности и совместимости.

Важно тестировать: многие настройки влияют на работу сервисов — нужно учитывать риски и окружение.

Ссылки

БЕКОН

LKDM

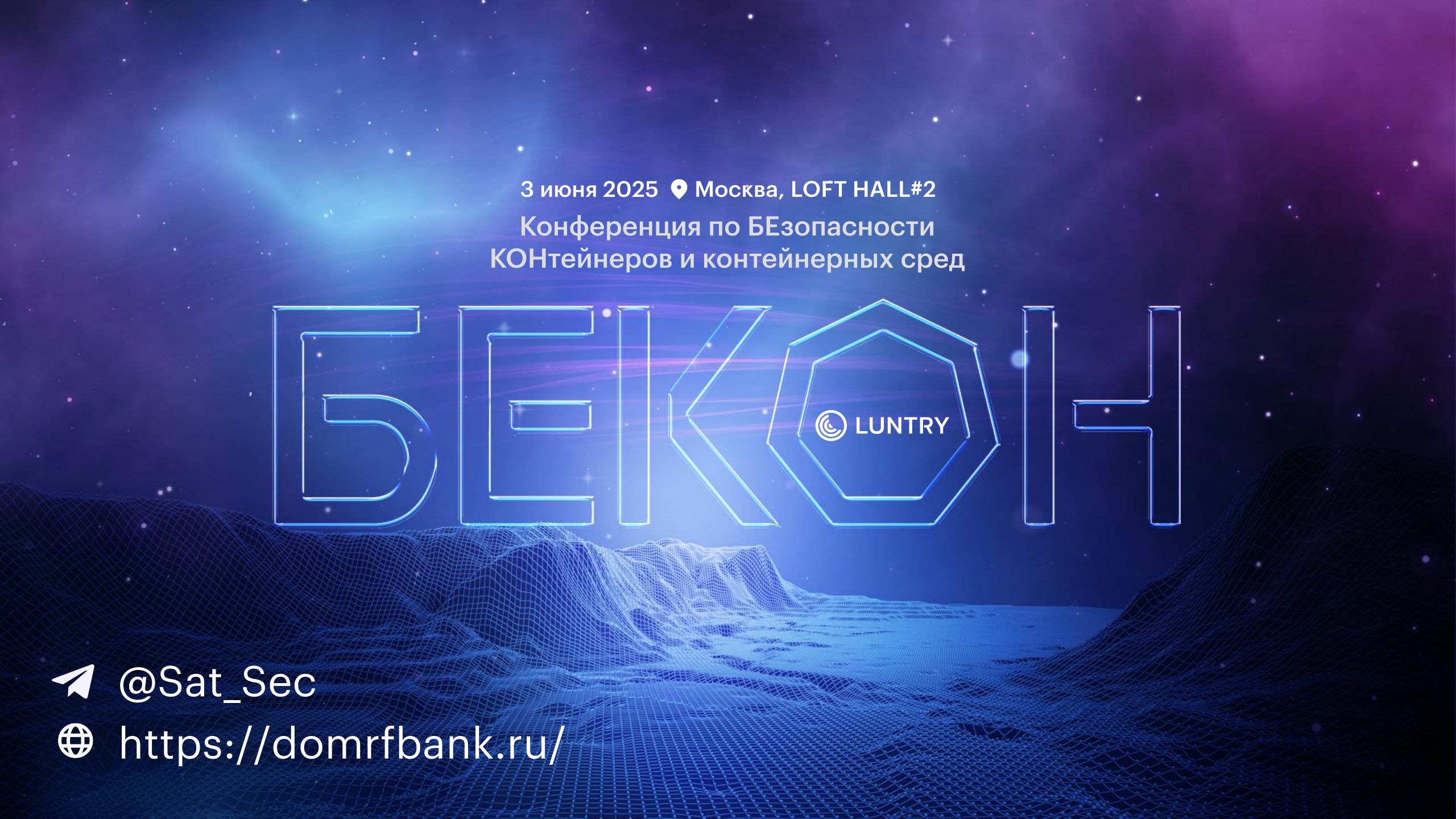
https://github.com/a13xp0p0v/linux-kernel-defence-mapCodeberg
https://codeberg.org/a13xp0p0v/linux-kernel-defence-map
https://gitflic.ru/project/a13xp0p0v/linux-kernel-defence-map

LKSPP

https://kspp.github.io/Recommended_Settings.html#:~:text=,max_user_namespaces%20%3D%200
kernel-hardening-checker

Источники для доклада

https://www.crowdstrike.com/en-us/blog/cve-2022-0185-kubernetes-container-escape-using-linux-kernel-exploit/#:~:text=Seccomp%20profile%20protects%20Linux%20namespace,as%20CAP_SYS_ADMIN%20for%20further%20attack
https://www.armosec.io/blog/cve-2022-0185-kubernetes-users/#:~:text=Linux%20maintainers%20disclosed%20a%20broadly,Linux%20kernel%20and%20all%20major
https://security-tracker.debian.org/tracker/CVE-2022-0185
https://www.opennet.ru/opennews/art.shtml?num=45848
https://stackoverflow.com/questions/21284906/perf-couldnt-record-kernel-reference-relocation-symbol#:~:text=Kernel%20address%20maps%20%28%2Fproc%2F,can%27t%20be%20resolved%20as%20well
https://docs.cilium.io/en/latest/reference-guides/bpf/architecture/
https://www.suse.com/support/kb/doc/?id=000020166#:~:text=If%20the%20sysctl%20,prevent%20Pod%20networking%20from%20functioning
https://dfir.ch/posts/today_i_learned_lkm_kernel.modules_disabled/#:~:text


3 июня 2025 Москва, LOFT HALL#2
Конференция по БЕзопасности
КОНтейнеров и контейнерных сред
БЕКОН
LUNTRY
@Sat_Sec
https://domrfbank.ru/