## РЕКОМЕНДАЦИИ

ПО УСИЛЕНИЮ ЗАЩИТЫ ОПЕРАЦИОННЫХ СИСТЕМ НА БАЗЕ LINUX

# Оглавление

[Введение 4](#_TOC_250039)

1. [Настройка авторизации в системе 5](#_TOC_250038)
   1. [Убедиться в отсутствии пользователей с пустыми паролями 5](#_TOC_250037)
   2. [Обеспечить отключение входа суперпользователя в систему по протоколу SSH 5](#_TOC_250036)
2. [Контроль за механизмами получения привилегий 7](#_TOC_250035)
   1. [Обеспечить ограничение доступа к команде su 7](#_TOC_250034)
   2. [Ограничить список пользователей sudo и выполняемых команд 7](#_TOC_250033)
3. [Настройка прав доступа к объектам файловой системы 9](#_TOC_250032)
   1. [Установить корректные права доступа к файлам настройки пользователей 9](#_TOC_250031)
   2. [Установить корректные права доступа к файлам запущенных процессов 9](#_TOC_250030)
   3. [Установить корректные права доступа к файлам, выполняющимся с помощью cron 10](#_TOC_250029)
   4. [Установить корректные права доступа к файлам, выполняемым с помощью sudo 11](#_TOC_250028)
   5. [Установить корректные права доступа к стартовым скриптам системы 11](#_TOC_250027)
   6. [Установить корректные права доступа к системным файлам заданий (конфигурационным файлам) cron 12](#_TOC_250026)
   7. [Установить корректные права доступа к пользовательским файлам заданий cron 13](#_TOC_250025)
   8. [Установить корректные права доступа к исполняемым файлам и библиотекам ОС 13](#_TOC_250024)
   9. Установить корректные права доступа к SUID/SGID-приложениям 14

3 10. Установить корректные права доступа к содержимому домашних директорий пользователей15

[3.11. Установить корректные права доступа к домашним директориям пользователей 15](#_TOC_250023)

1. Настройка средств защиты ядра Linux 17
   1. [Ограничить доступ к журналу ядра 17](#_TOC_250022)
   2. [Заменить ядерные адреса в /proc и других интерфейсах на 0 17](#_TOC_250021)
   3. [Инициализировать динамическую ядерную память нулем при ее выделении 18](#_TOC_250020)
   4. [Запретить слияние кэшей ядерного аллокатора 18](#_TOC_250019)
   5. [Инициализировать механизм IOMMU 19](#_TOC_250018)
   6. [Рандомизировать расположение ядерного стека 19](#_TOC_250017)
   7. [Включить средства защиты от аппаратных уязвимостей центрального процессора (для платформы x86) 20](#_TOC_250016)
   8. Включить защиту подсистемы eBPF JIT ядра Linux 21
2. Уменьшение периметра атаки ядра Linux 22
   1. [Отключить устаревший интерфейс vsyscall 22](#_TOC_250015)
   2. [Ограничить доступ к событиям производительности 22](#_TOC_250014)
   3. [Отключить монтирование виртуальной файловой системы debugfs 23](#_TOC_250013)
   4. [Отключить системный вызов kexec\_load 23](#_TOC_250012)
   5. [Ограничить использование user namespaces 24](#_TOC_250011)
   6. [Запретить системный вызов bpf для непривилегированных пользователей 24](#_TOC_250010)
   7. [Запретить системный вызов userfaultfd для непривилегированных пользователей 25](#_TOC_250009)
   8. [Запретить автоматическую загрузку модулей ядра, отвечающих за поддержку дисциплины линии терминала 26](#_TOC_250008)
   9. [Отключить технологию Transactional Synchronization Extensions (TSX) 26](#_TOC_250007)
3. [Настройка средств защиты пользовательского пространства со стороны ядра Linux 28](#_TOC_250006)
   1. [Запретить подключение к другим процессам с помощью ptrace 28](#_TOC_250005)
   2. [Ограничить небезопасные варианты прохода по символическим ссылкам (symlinks) 28](#_TOC_250004)
   3. [Ограничить небезопасные варианты работы с жесткими ссылками (hardlinks) 29](#_TOC_250003)
   4. Включить защиту от непреднамеренной записи в FIFO-объект, контролируемый атакующим 30
   5. [Включить защиту от непреднамеренной записи в файл, контролируемый атакующим 30](#_TOC_250002)
   6. [Запретить создание core dump для некоторых исполняемых файлов 31](#_TOC_250001)

[Приложение. Чек-лист выполнения настроек безопасности ОС на базе Linux 32](#_TOC_250000)

# Введение

В данном документе представлены рекомендации по усилению защищенности Linux-систем. Они позволяют снизить риски реализации угроз безопасности.

В данные рекомендации входит включение средств защиты операционной системы и уменьшение ее периметра атаки.

Рекомендации и соответствующие условия, ограничения и комментарии сгруппированы по настраиваемым подсистемам.

# Настройка авторизации в системе

## Убедиться в отсутствии пользователей с пустыми паролями

**Описание:** пользователи с пустым паролем представляют существенную угрозу для защиты целевой системы, так как в большинстве случаев злоумышленнику достаточно знать имя соответствующего пользователя для успешной удаленной авторизации в системе.

**Рекомендуемые настройки:** убедиться, что каждый пользователь системы либо имеет пароль, либо заблокирован по паролю. В Linux-системах данную возможность обеспечивает файл /etc/shadow.

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

**Скорость отмены внесенных изменений:** высокая **Возможность автоматизации:** Ansible (ограниченно) **Приоритет:** высокий

**Контроль MP8:** PT – Essential (603610)

## Обеспечить отключение входа суперпользователя в систему по протоколу SSH

**Описание:** отключение возможности авторизации в системе по протоколу SSH для суперпользователя потребует от администраторов сервера выполнять авторизацию с помощью их персональной учетной записи с последующим повышением прав до суперпользователя через команду sudo или su. Это не позволит пользователям отказаться от своих действий и обеспечит наличие последовательности зафиксированных событий в случае инцидента безопасности.

**Рекомендуемые настройки:** в файле /etc/ssh/sshd\_config установить для параметра PermitRootLogin значение no

#### Известные проблемы: нет

**Худшие последствия:** нет

**Последующие ограничения:** необходимость использования персональных учетных записей администраторов

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** CIS - Distribution Independent Linux (602009)

# Контроль за механизмами получения привилегий

## Обеспечить ограничение доступа к команде su

**Описание:** ограничение доступа к команде su и использование вместо нее sudo позволяет более эффективно контролировать повышение пользовательских привилегий для исполнения привилегированных команд. Утилита sudo также предоставляет более совершенные механизмы журналирования и аудита, поскольку она способна регистрировать все команды, выполненные с помощью sudo, в то время как su фиксирует только сам факт использования программы пользователем.

**Рекомендуемые настройки:** добавить в файл /etc/pam.d/su следующую строку:

auth required pam\_wheel.so use\_uid

Задать список пользователей в записи для группы wheel в файле /etc/group: wheel:x:10:root,<user list>

#### Известные проблемы: нет

**Худшие последствия:** нет

**Последующие ограничения:** изменение технологии работы администраторов

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** низкий

**Контроль MP8:** CIS - Distribution Independent Linux (602026)

## Ограничить список пользователей sudo и выполняемых команд

**Описание:** ограничение списка пользователей, которым разрешено использовать команду sudo и, по возможности, разрешенных к выполнению через sudo команд, позволяет снизить вероятность повышения привилегий злоумышленника в системе.

**Рекомендуемые настройки:** провести ревью файла /etc/sudoers.

#### Известные проблемы: нет

**Худшие последствия:** нет

**Последующие ограничения:** изменение технологии работы администраторов

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** нет

**Приоритет:** высокий

**Контроль MP8:** —

# Настройка прав доступа к объектам файловой системы

## Установить корректные права доступа к файлам настройки пользователей

**Описание:** при слабых правах доступа к файлам с перечнями пользовательских идентификаторов (/etc/passwd) и групп (/etc/group), либо хранилищам хешей паролей (в ОС GNU/Linux, Solaris, HP-UX: /etc/shadow, AIX: /etc/security/passwd, FreeBSD:

/etc/master.passwd) пользователи могут прочесть либо перезаписать хеши паролей, создать учетные записи администраторов, повысить свои привилегии в системе.

**Рекомендуемые настройки:** установить корректные права доступа к указанным конфигурационным файлам с помощью команд:

* + - chmod 644 /etc/passwd
    - chmod 644 /etc/group
    - chmod go-rwx /etc/shadow **Известные проблемы:** нет **Худшие последствия:** неизвестны **Последующие ограничения:** нет

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** PT - Essential (UNIX) 603607

## Установить корректные права доступа к файлам запущенных процессов

**Описание:** контроль всех исполняемых файлов, запускаемых во время работы ОС, является нетривиальной задачей. Однако, в каждый момент времени можно установить, какие процессы запущены, и соответственно – какие исполняемые файлы с ними связаны.

Права доступа к этим исполняемым файлам должны быть в значении 755 или строже, а их владельцами в подавляющем большинстве случаев должен являться суперпользователь root, либо системные пользователи (то есть не имеющие возможности авторизоваться в системе с паролем).

В противном случае, если исполняемый файл процесса может быть изменен непривилегированным пользователем, будет выполнен код с правами пользователя,

запустившего процесс. Если таким пользователем был root, то злоумышленник получит полный контроль над системой.

**Рекомендуемые настройки:** для всех исполняемых файлов, запущенных в настоящий момент, и соответствующих библиотек, выполнить команду вида:

chmod go-w /путь/к/файлу

Также необходимо убедиться, что директория, содержащая данный файл, а также все родительские директории недоступны для записи непривилегированным пользователям.

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** PT - Essential (UNIX) 603606

## Установить корректные права доступа к файлам, выполняющимся с помощью cron

**Описание:** с помощью подсистемы cron могут вызываться как как системные команды, так и различные пользовательские скрипты. Требуется предотвратить возможность изменения файлов, которые вызываются из заданий cron неавторизованными пользователями. В противном случае это может привести к выполнению произвольного кода от имени владельца задания cron (в том числе root, что приведет к полной компрометации системы).

**Рекомендуемые настройки:** для каждого файла (либо команды), который вызывается из заданий cron, необходимо выполнить команду вида:

chmod go-w путь\_к\_файлу

Указание команд, выполняющихся через cron, может быть как в пользовательских файлах заданий (располагающихся в директории /var), так и в системных файлах заданий (/etc/cron\*).

Также следует убедиться, что права доступа ко всем директориям, где расположены соответствующие команды и скрипты, имеют значение 755 или строже.

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** средние

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible (тяжело)

**Приоритет:** высокий

**Контроль MP8:** PT - Essential (UNIX) 603605

## Установить корректные права доступа к файлам, выполняемым с помощью sudo

**Описание:** некорректные права доступа к файлам, которые разрешено выполнять пользователям с помощью утилиты sudo, могут привести к возможности неавторизованного выполнения команд.

**Рекомендуемые настройки:** для каждого исполняемого файла, который можно запускать с привилегиями суперпользователя root, но владельцем которого является обычный пользователь, изменить владельца командой вида:

chown root путь\_к\_файлу

Для каждого исполняемого файла, который можно запускать с привилегиями суперпользователя root и к которому имеют доступ на запись все пользователи, выполнить команду:

chmod go-w путь\_к\_файлу **Известные проблемы:** нет **Худшие последствия:** нет **Последующие ограничения:** нет

#### Снижение производительности: нет

**Трудозатраты:** средние

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible (значительная сложность реализации)

**Приоритет:** высокий

**Контроль MP8:** PT - Essential (UNIX) 603604

## Установить корректные права доступа к стартовым скриптам системы

**Описание:** все UNIX-системы осуществляют запуск сервисов при старте ОС от имени суперпользователя root (независимо от используемой подсистемы старта: SysV init, upstart. systemd). Следовательно, если на исполняемые файлы сервисов (скрипты, бинарные файлы) установлены некорректные права доступа, то злоумышленник, имеющий локальный доступ к системе, может изменить их содержимое и выполнить произвольный код с правами root при следующем старте ОС или перезапуске сервиса.

**Рекомендуемые настройки:** к каждому файлу в директориях /etc/rc#.d, а также к файлам .service, присутствующим в системе, выполнить команду вида:

chmod o-w <filename>

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** PT - Essential (UNIX) 603603

## Установить корректные права доступа к системным файлам заданий (конфигурационным файлам) cron

**Описание:** подсистема cron позволяет выполнять задания (команды, скрипты) с заданной периодичностью.

Если права доступа к системным файлам-описаниям очередей cron установлены некорректно, злоумышленник с локальным доступом получит возможность выполнять произвольные команды от с правами root, тем самым полностью скомпрометировав систему.

К системным файлам-описаниям очередей cron относятся следующие файлы (могут присутствовать не всегда, в зависимости от системы и ее настроек):

* + - /etc/crontab
    - /etc/cron.d (директория и файлы внутри нее)
    - /etc/cron.hourly (директория и файлы внутри нее)
    - /etc/cron.daily (директория и файлы внутри нее)
    - /etc/cron.weekly (директория и файлы внутри нее)
    - /etc/cron.monthly (директория и файлы внутри нее)

**Рекомендуемые настройки:** установить корректные права доступа к файлам очередей cron при помощи команды вида:

chmod go-wx путь\_к\_файлу\_или\_директории

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** PT - Essential (UNIX) 603602

## Установить корректные права доступа к пользовательским файлам заданий cron

**Описание:** подсистема cron позволяет выполнять задания (команды, скрипты) с заданной периодичностью.

Если права доступа к файлам-описаниям очередей cron установлены некорректно, это даст возможность злоумышленнику с локальным доступом к системе выполнить произвольные команды от имени пользователя-владельца задания.

**Рекомендуемые настройки:** установить корректные права доступа к файлам очередей cron при помощи команды вида:

chmod go-w путь\_к\_файлу\_заданий

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** средний

**Контроль MP8:** PT - Essential (UNIX) 603601

## Установить корректные права доступа к исполняемым файлам и библиотекам ОС

**Описание:** во время запуска и работы ОС выполняется множество программ и скриптов, зачастую это происходит от имени root. При некорректных правах доступа к подобным утилитам (например, позволяющих любому пользователю системы перезаписать содержимое файла утилиты) система c большой вероятностью окажется полностью скомпрометированной. Ещё большую опасность представляет недостаточное разграничение доступа к файлам модулей ядра Linux, в особенности к запускаемым при старте системы – при их подмене злоумышленник полностью скомпрометирует систему.

**Рекомендуемые настройки:** определение полного списка исполняемых файлов и библиотек, выполняемых при старте системы, входящих в установленное ПО и периодические задания (утилиты cron, at) представляется трудоемкой задачей. Однако, местоположение большинства стандартных исполняемых файлов указано в переменной

$PATH пользователя root. Необходимо убедиться в корректных правах доступа к утилитам и системным библиотекам, расположенным по стандартным путям (/bin, /usr/bin, /lib,

/lib64 и т.п.), а также к модулям ядра (для Linux: /lib/modules/версия-текущего-

ядра).

Права доступа к указанным файлам должны иметь значение 755 или более строгое.

Для каждого обнаруженного несоответствия следует выполнить команду вида:

chmod go-w /путь/к/приложению **Известные проблемы:** нет **Худшие последствия:** нет **Последующие ограничения:** нет

#### Снижение производительности: нет

**Трудозатраты:** средние

**Скорость отмены внесенных изменений:** высокая **Возможность автоматизации:** Ansible (частично) **Приоритет:** высокий

**Контроль MP8:** PT - Essential (UNIX) 603600

## Установить корректные права доступа к

**SUID/SGID-приложениям**

**Описание:** приложения с установленным битом SUID выполняются с правами их владельца, с битом SGID – с правами группы владения. Если права доступа хотя бы к одному из этих приложений позволят остальным пользователям изменять содержимое этого приложения, это приведет к выполнению произвольного кода от имени пользователя- владельца приложения; если это будет SUID-приложение и его владельцем окажется суперпользователь root, это приведет к полной компрометации системы.

**Рекомендуемые настройки:** провести аудит системы на предмет поиска всех SUID/SGID-приложений – права доступа к каждому из них не должны позволять остальным пользователям изменять его содержимое (в особенности если это SUID-приложение и его владелец root). В противном случае следует выполнить команду вида:

chmod go-w /путь/к/приложению

Также убедиться, что среди выявленных SUID/SGID-приложений не присутствуют лишние (например, если определен «белый» список таких приложений), в противном случае следует снять с таких приложений SUID/SGID-биты.

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** PT - Essential (UNIX) 603599

## Установить корректные права доступа к содержимому домашних директорий пользователей

**Описание:** домашние директории интерактивных (не системных) пользователей содержат, как минимум, следующие ценные для злоумышленника файлы:

* + - При возможности чтения: файлы истории команд оболочек (.bash\_history,

.history, .sh\_history и т.п.)

* + - При возможности записи:
      * файлы настройки оболочки (.bash\_profile, .bashrc, .profile,

.bash\_logout и т.п.);

* + - * файлы настройки R-подсистем (.rhosts).

Доступ к указанным файлам должен быть только у их владельца.

**Рекомендуемые настройки:** на каждый из указанных файлов установить корректные права доступа с помощью команды вида:

chmod go-rwx путь\_к\_файлу **Известные проблемы:** нет **Худшие последствия:** нет **Последующие ограничения:** нет

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** средний

**Контроль MP8:** PT - Essential (UNIX) 603594

## Установить корректные права доступа к домашним директориям пользователей

**Описание:** в домашних директориях пользователей часто хранится информация, представляющая интерес для злоумышленника: файлы журналов, история действий, различные архивы. Для домашних директорий интерактивных (не системных) пользователей требуется установить права доступа, запрещающие чтение содержимого, изменение и обращение к данным для всех, кроме самого пользователя (в некоторых случаях допустимо установить разрешения также для группы).

**Рекомендуемые настройки:** установить права доступа 700 с помощью команды

вида:

chmod 700 домашняя\_директория **Известные проблемы:** нет **Худшие последствия:** нет **Последующие ограничения:** нет

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** средний

**Контроль MP8:** PT - Essential (UNIX) 603587

# Настройка средств защиты ядра

**Linux**

В данном разделе перечислены sysctl-опции и опции загрузки ядра (kernel cmdline parameters), которые позволяют усложнить эксплуатацию уязвимостей в ядре Linux.

Если же есть возможность изменить опции компиляции ядра Linux в вашей системе, то рекомендуем воспользоваться открытым инструментом **kconfig-hardened-check**. Это open source проект, который при поддержке сообщества разрабатывает Александр Попов. Данный инструмент предоставляет рекомендации по настройкам безопасности ядра Linux: https://github.com/a13xp0p0v/kconfig-hardened-check

## Ограничить доступ к журналу ядра

**Описание:** ограничение возможности непривилегированных пользователей использовать dmesg для чтения журнала ядра предотвращает утечку информации, которой может воспользоваться атакующий при эксплуатации уязвимостей ядра.

Журнал ядра должен быть доступен только пользователям, которые обладают разрешением CAP\_SYSLOG (администраторы системы). Это никак не вредит обычным сценариям использования систем.

**Рекомендуемые настройки:** установить значение sysctl-опции

kernel.dmesg\_restrict=1 **Известные проблемы:** нет **Худшие последствия:** нет

#### Последующие ограничения: нет Снижение производительности: нет Трудозатраты: низкие

**Скорость отмены внесенных изменений:** высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Заменить ядерные адреса в /proc и других интерфейсах на 0

**Описание:** замена ядерных адресов в /proc и других интерфейсах на 0 предотвращает утечку информации, которой может воспользоваться атакующий при эксплуатации уязвимостей ядра.

**Рекомендуемые настройки:** установить значение sysctl-опции

kernel.kptr\_restrict=2

#### Известные проблемы: нет

**Худшие последствия:** нет

**Последующие ограничения:** это в некоторой мере затрудняет отладку для разработчиков ядра Linux, но никак не влияет на обычные сценарии использования систем

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Инициализировать динамическую ядерную память нулем при ее выделении

**Описание:** инициализация динамической ядерной памяти нулем при ее выделении затрудняет эксплуатацию уязвимостей типа use-after-free (использование памяти после освобождения). Данная технология успешно применяется в ядре Linux для Android.

**Рекомендуемые настройки:** установить значение опции загрузки ядра init\_on\_alloc=1. Эта настройка позволяет изменить значение опции сборки ядра CONFIG\_INIT\_ON\_ALLOC\_DEFAULT\_ON при запуске системы (per boot).

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** возможно незначительное падение производительности (считается приемлемым для Android)

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Запретить слияние кэшей ядерного аллокатора

**Описание:** запрет слияния кэшей ядерного аллокатора существенно затрудняет эксплуатацию повреждения памяти в ядерной куче (heap).

**Рекомендуемые настройки:** установить опцию загрузки ядра slab\_nomerge. Эта настройка позволяет изменить значение опции сборки ядра CONFIG\_SLAB\_MERGE\_DEFAULT при запуске системы (per boot).

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Инициализировать механизм IOMMU

**Описание:** механизм IOMMU противодействует DMA-атакам, в ходе которых выполняется произвольный доступ к системной памяти со стороны устройств на шине ввода- вывода, например PCI.

Защита от DMA-атак напрямую зависит от возможностей IOMMU на конкретной платформе, а также от того, как именно настраиваются домены IOMMU со стороны прошивки и операционной системы.

Как бы то ни было, включать IOMMU обязательно стоит.

**Рекомендуемые настройки:** установить значения для следующих опций загрузки

ядра:

* + - iommu=force
    - iommu.strict=1
    - iommu.passthrough=0

**Известные проблемы:** необходимо проверить, что устройства на шине ввода-

вывода работают корректно

#### Худшие последствия: нет

**Последующие ограничения:** нет

**Снижение производительности:** возможно при определенных сценариях

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Рандомизировать расположение ядерного стека

**Описание:** рандомизация расположения ядерного стека при каждом очередном системном вызове мешает работе эксплойтов, использующих данные в ядерном стеке.

**Рекомендуемые настройки:** установить значение опции загрузки ядра randomize\_kstack\_offset=1. Эта настройка позволяет изменить значение опции сборки ядра CONFIG\_RANDOMIZE\_KSTACK\_OFFSET\_DEFAULT при запуске системы (per boot).

**Известные проблемы:** нет

**Худшие последствия:** нет

**Последующие ограничения:** нет

**Снижение производительности:** соизмеримо с погрешностью измерения

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Включить средства защиты от аппаратных уязвимостей центрального процессора (для платформы x86)

**Описание:** в ядре Linux присутствуют средства защиты от аппаратных уязвимостей центрального процессора, которые также известны как Transient Execution Vulnerabilities. Это Meltdown, Spectre, MDS и другие.

**Рекомендуемые настройки:** установить значение опции загрузки ядра

mitigations=auto,nosmt.

Значение auto включает имеющиеся средства защиты, но при этом оставляет включенной одновременную многопоточность (Simultaneous Multithreading — SMT), даже если эта технология подвержена описанным уязвимостям.

Значение auto,nosmt дополнительно отключает SMT, что влияет на производительность системы.

Важно заметить, что многие их данных средств защиты требуют также обновления микрокода.

На решение о включении данных средств защиты влияет:

* модель безопасности информационной системы,
* возможность исполнения недоверенного кода на данной системе,
* использование средств виртуализации на данной системе.

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** зависит от типа нагрузки системы, поэтому рекомендуется предварительно провести нагрузочное тестирование информационной системы

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Включить защиту подсистемы eBPF JIT ядра

**Linux**

**Описание:** в ядре Linux существует механизм защиты подсистемы eBPF JIT, который служит против техники "JIT spraying", используемой в ядерных эксплойтах.

**Рекомендуемые настройки:** установить значение sysctl-опции

net.core.bpf\_jit\_harden=2 **Известные проблемы:** нет **Худшие последствия:** нет **Последующие ограничения:** нет

**Снижение производительности:** данное средство защиты незначительно замедляет только движок eBPF JIT. Его замедление на 8% можно измерить на микробенчмарке из test\_bpf.ko. С включением bpf\_jit\_harden движок eBPF JIT все равно остается почти в два раза быстрее, чем интерпретатор eBPF.

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

# Уменьшение периметра атаки ядра

**Linux**

## Отключить устаревший интерфейс vsyscall

**Описание:** если устаревший интерфейс vsyscall включен, то ядро размещает vsyscall- функции по фиксированным адресам 0xffffffffff600x00, что дает атакующему большие возможности при перехвате потока управления. Присутствие в системе этого устаревшего механизма нежелательно.

**Рекомендуемые настройки:** установить значение опции загрузки ядра vsyscall=none. Эта настройка позволяет изменить значение опции сборки ядра CONFIG\_LEGACY\_VSYSCALL\_NONE при запуске системы (per boot).

#### Известные проблемы: нет

**Худшие последствия:** нет

**Последующие ограничения:** это никак не помешает, если на системе не запускаются статически слинкованные исполняемые файлы и программы, использующие glibc до версии 2.14

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Ограничить доступ к событиям производительности

**Описание:** события производительности (performance events) представляют ценность для атакующего. Поэтому они должны быть доступны только пользователям, которые обладают разрешением CAP\_PERFMON (администраторы системы). Это не вредит обычным сценариям использования систем.

**Рекомендуемые настройки:** установить значение sysctl-опции

kernel.perf\_event\_paranoid=3 **Известные проблемы:** нет **Худшие последствия:** нет **Последующие ограничения:** нет

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Отключить монтирование виртуальной файловой системы debugfs

**Описание:** виртуальная файловая система debugfs дает дополнительные возможности атакующему и также значительно увеличивает периметр атаки ядра. Если есть возможность, эту функциональность стоит полностью отключить.

**Рекомендуемые настройки:** установить значение опции загрузки ядра

debugfs=no-mount (или лучше off, если возможно).

#### Известные проблемы: нет

**Худшие последствия:** нет

**Последующие ограничения:** данная настройка делает интерфейсы debugfs недоступными для userspace, поэтому предварительно стоит уточнить необходимость в них

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Отключить системный вызов kexec\_load

**Описание:** с помощью системного вызова kexec\_load атакующий может подменить образ ядра и сломать цепочку доверенной загрузки. Это один из способов установки руткитов на Linux-системах.

**Рекомендуемые настройки:** установить значение sysctl-опции

kernel.kexec\_load\_disabled=1 **Известные проблемы:** нет **Худшие последствия:** нет

**Последующие ограничения:** сценарии использования систем, задействующие kexec\_load, становятся недоступны. Однако системы с повышенными требованиями по безопасности и с построенной цепочкой доверенной загрузки не используют такие сценарии.

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Ограничить использование user namespaces

**Описание:** механизм user namespaces используется для некоторых режимов работы контейнеров в пользовательском пространстве. USER\_NS позволяет независимо назначать uid внутри контейнера. В частности, USER\_NS дает создать ненастоящего пользователя root, который будет администратором только в этом контейнере.

Однако у данной функциональности есть серьезный побочный эффект для безопасности ядра. Ненастоящий root внутри контейнера имеет доступ к привилегированным API ядра. Это критически увеличивает периметр атаки на ядро.

USER\_NS в основном используется веб-браузерами для изоляции своих процессов друг от друга. А для серверного ПО, не требующего этот функционал, USER\_NS несет только дополнительные риски безопасности ядра.

Большое количество ядерных эксплойтов работает именно из USER\_NS. Поэтому множество авторитетных источников рекомендуют отключать USER\_NS, если эта функциональность не является критически важной для выполнения бизнес-задачи информационной системы.

Статья по поводу дискуссии о USER\_NS в списке рассылки LKML: https://lwn.net/Articles/673597/

**Рекомендуемые настройки:** установить значение sysctl-опции

user.max\_user\_namespaces=0 **Известные проблемы:** нет **Худшие последствия:** нет

**Последующие ограничения:** если система на базе Linux не использует user namespaces для выполнения своей задачи, то данная настройка никак не повлияет на работу системы. Рекомендуется проверить функциональность на тестовой системе после включения данной настройки.

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Запретить системный вызов bpf для непривилегированных пользователей

**Описание:** необходимо ограничить возможность непривилегированного пользователя загружать код в ядро Linux с помощью BPF (Berkeley Packet Filter). Это важно, так как множество атак на ядро Linux полагаются на непривилегированный вызов bpf().

**Рекомендуемые настройки:** установить значение sysctl-опции

kernel.unprivileged\_bpf\_disabled=1

#### Известные проблемы: нет

**Худшие последствия:** нет

**Последующие ограничения:** если непривилегированные процессы в системе на базе Linux не используют BPF для выполнения своей задачи, то данная настройка никак не повлияет на работу системы. Рекомендуется проверить функциональность на тестовой системе после включения данной настройки.

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Запретить системный вызов userfaultfd для непривилегированных пользователей

**Описание:** системный вызов userfaultfd() очень часто используется атакующими в эксплойтах для уязвимостей ядра Linux. Один из многочисленных примеров: https://a13xp0p0v.github.io/2021/02/09/CVE-2021-26708.html

Данная функциональность, доступная непривилегированному пользователю, позволяет построить надежный эксплойт-примитив для выполнения heap spraying: https://duasynt.com/blog/linux-kernel-heap-spray

При этом программы в пользовательском пространстве не используют userfaultfd() для памяти ядра. Поэтому ограничение данной механизма в ядре версии выше v5.11 не влияет на функциональность userspace.

**Рекомендуемые настройки:** установить значение sysctl-опции

vm.unprivileged\_userfaultfd=0 **Известные проблемы:** нет **Худшие последствия:** нет **Последующие ограничения:** нет

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Запретить автоматическую загрузку модулей ядра, отвечающих за поддержку дисциплины линии терминала

**Описание:** есть возможность уменьшить периметр атаки ядра Linux, если запретить автоматической загрузки модулей ядра, отвечающих за поддержку дисциплины линии (ldisc) для терминала (tty).

Есть ряд примеров, когда атакующие эксплуатировали уязвимости в старых ldisc- модулях для поднятия привилегий в системе.

**Рекомендуемые настройки:** установить значение sysctl-опции

dev.tty.ldisc\_autoload=0

С этой настройкой ldisc-модули могут быть загружены только привилегированным пользователем. Это не вредит обычным сценариям использования систем.

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Отключить технологию Transactional Synchronization Extensions (TSX)

**Описание:** технология Transactional Synchronization Extensions (TSX), которая присутствует в процессорах Intel, подвержена аппаратной уязвимости TSX Asynchronous Abort (TAA) CVE-2019-11135.

Рекомендуется отключать инструкции TSX-NI для работы с транзакционной памятью, если они не используются на системе.

**Рекомендуемые настройки:** установить значение опции загрузки ядра tsx=off

#### Известные проблемы: нет Худшие последствия: нет Последующие ограничения: нет

**Снижение производительности:** нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

# Настройка средств защиты пользовательского пространства со стороны ядра Linux

## Запретить подключение к другим процессам с помощью ptrace

**Описание:** важно запрещать подключение к другим процессам с помощью ptrace, так как атакующие используют этот механизм для перехвата SSH-сессий и внедрения произвольного кода.

Механизм ptrace используется разработчиками при отладке. Системы в промышленной эксплуатации обычно его не используют.

**Рекомендуемые настройки:** установить значение sysctl-опции

kernel.yama.ptrace\_scope=3 **Известные проблемы:** нет **Худшие последствия:** нет **Последующие ограничения:** нет

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Ограничить небезопасные варианты прохода по символическим ссылкам (symlinks)

**Описание:** в Linux существует старинная проблема с эксплуатацией состояния гонки при обработке символических ссылок (symlinks). Этот тип ошибок используется для атак с целью повышения привилегий в системе.

Пример: привилегированный процесс проверяет отсутствие некоторого файла, который собирается создать. В момент между этой проверкой и созданием файла атакующий может успеть создать символическую ссылку с таким же именем, но указывающую на другой файл. В результате привилегированный процесс произведет запись в файл, указанный атакующим.

https://en.wikipedia.org/wiki/Symlink\_race

Необходимо ограничивать небезопасные варианты прохода по символическим ссылкам.

**Рекомендуемые настройки:** установить значение sysctl-опции

fs.protected\_symlinks=1 **Известные проблемы:** нет **Худшие последствия:** нет

**Последующие ограничения:** нет. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Ограничить небезопасные варианты работы с жесткими ссылками (hardlinks)

**Описание:** в Linux также существует проблема с эксплуатацией состояния гонки при обработке жестких ссылок (hardlinks). Этот тип ошибок используется для атак с целью повышения привилегий в системе.

Метод эксплуатации аналогичен с атакой через symlink\_race: привилегированный процесс проходит по ссылке, созданной другим пользователем.

Необходимо ограничивать небезопасную работу с жесткими ссылками: пользователь должен иметь возможность создать жесткую ссылку на файл, только если он является владельцем этого файла или обладает правами чтения и записи для него.

**Рекомендуемые настройки:** установить значение sysctl-опции

fs.protected\_hardlinks=1 **Известные проблемы:** нет **Худшие последствия:** нет

**Последующие ограничения:** нет. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Включить защиту от непреднамеренной записи в

**FIFO-объект, контролируемый атакующим**

**Описание:** в ядре Linux существует механизм, защищающий пользовательские программы от непреднамеренной записи в FIFO-объект, контролируемый атакующим, вместо нормальной работы с файлом.

С помощью данного механизма ядро запрещает небезопасную работу с FIFO, чтобы предотвратить data spoofing атаки. А именно, операция open для FIFO с флагом O\_CREAT блокируется, если файл уже существует, пользователь не является владельцем директории и в ряде других случаев.

**Рекомендуемые настройки:** установить значение sysctl-опции

fs.protected\_fifos=2 **Известные проблемы:** нет **Худшие последствия:** нет

**Последующие ограничения:** нет. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Включить защиту от непреднамеренной записи в файл, контролируемый атакующим

**Описание:** в ядре Linux также существует механизм, защищающий пользовательские программы от непреднамеренной записи файл, контролируемый атакующим, вместо нормальной работы с оригинальным файлом.

Это средство защиты работает аналогично с механизмом protected\_fifos.

**Рекомендуемые настройки:** установить значение sysctl-опции

fs.protected\_regular=2 **Известные проблемы:** нет **Худшие последствия:** нет

**Последующие ограничения:** нет. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

**Контроль MP8:** –

## Запретить создание core dump для некоторых исполняемых файлов

**Описание:** создание core dump для некоторых исполняемых файлов несет угрозы безопасности системы. Это файлы со следующими свойствами:

* + - измененный уровень привилегий (suid),
    - доступ только на исполнение (execute only).

Необходимо ограничивать возможность создавать core dump для данных исполняемых файлов.

**Рекомендуемые настройки:** установить значение sysctl-опции

fs.suid\_dumpable=0

#### Известные проблемы: нет

**Худшие последствия:** нет

**Последующие ограничения:** нет. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

#### Снижение производительности: нет

**Трудозатраты:** низкие

#### Скорость отмены внесенных изменений: высокая

**Возможность автоматизации:** Ansible

**Приоритет:** высокий

#### Контроль MP8: –

### Приложение. Чек-лист выполнения настроек безопасности ОС на базе Linux

|  |  |
| --- | --- |
| **Имя хоста:** |  |
| **IP-адрес:** |  |
| **ОС:** |  |
| **Версия ядра:** |  |
| **Комментарии:** |  |

|  |  |  |
| --- | --- | --- |
| **№ Название пункта Отметка о**  **выполнении** | | |
| **1.** | **Настройка авторизации в системе** |  |
| 1.1. | Убедиться в отсутствии пользователей с пустыми паролями |  |
| 1.2. | Обеспечить отключение входа суперпользователя в систему по протоколу SSH |  |
| **2.** | **Контроль за механизмами получения привилегий** |  |
| 2.1. | Обеспечить ограничение доступа к команде su |  |
| 2.2. | Ограничить список пользователей sudo и выполняемых команд |  |
| **3.** | **Настройка прав доступа к объектам файловой системы** |  |
| 3.1. | Установить корректные права доступа к файлам настройки пользователей |  |
| 3.2. | Установить корректные права доступа к файлам запущенных процессов |  |
| 3.3. | Установить корректные права доступа к файлам, выполняющимся с помощью cron |  |
| 3.4. | Установить корректные права доступа к файлам, выполняемым с помощью sudo |  |
| 3.5. | Установить корректные права доступа к стартовым скриптам системы |  |
| 3.6. | Установить корректные права доступа к системным файлам заданий (конфигурационным файлам)  cron |  |
| 3.7. | Установить корректные права доступа к пользовательским файлам заданий cron |  |
| 3.8. | Установить корректные права доступа к исполняемым файлам и библиотекам ОС |  |
| 3.9. | Установить корректные права доступа к SUID/SGID-приложениям |  |
| 3.10. | Установить корректные права доступа к содержимому домашних директорий пользователей |  |
| 3.11. | Установить корректные права доступа к домашним директориям пользователей |  |
| **4.** | **Настройка средств защиты ядра Linux** |  |
| 4.1. | Ограничить доступ к журналу ядра |  |
| 4.2. | Заменить ядерные адреса в /proc и других интерфейсах на 0 |  |
| 4.3. | Инициализировать динамическую ядерную память нулем при ее выделении |  |
| 4.4. | Запретить слияние кэшей ядерного аллокатора |  |
| 4.5. | Инициализировать механизм IOMMU |  |
| 4.6. | Рандомизировать расположение ядерного стека |  |
| 4.7. | Включить средства защиты от аппаратных уязвимостей центрального процессора (для платформы  x86) |  |
| 4.8. | Включить защиту подсистемы eBPF JIT ядра Linux |  |
| **5.** | **Уменьшение периметра атаки ядра Linux** |  |
| 5.1. | Отключить устаревший интерфейс vsyscall |  |
| 5.2. | Ограничить доступ к событиям производительности |  |
| 5.3. | Отключить монтирование виртуальной файловой системы debugfs |  |
| 5.4. | Отключить системный вызов kexec\_load |  |
| 5.5. | Ограничить использование user namespaces |  |
| 5.6. | Запретить системный вызов bpf для непривилегированных пользователей |  |
| 5.7. | Запретить системный вызов userfaultfd для непривилегированных пользователей |  |
| 5.8. | Запретить автоматическую загрузку модулей ядра, отвечающих за поддержку дисциплины линии  терминала |  |
| 5.9. | Отключить технологию Transactional Synchronization Extensions (TSX) |  |
| **6.** | **Настройка средств защиты пользовательского пространства со стороны ядра Linux** |  |
| 6.1. | Запретить подключение к другим процессам с помощью ptrace |  |
| 6.2. | Ограничить небезопасные варианты прохода по символическим ссылкам (symlinks) |  |
| 6.3. | Ограничить небезопасные варианты работы с жесткими ссылками (hardlinks) |  |
| 6.4. | Включить защиту от непреднамеренной записи в FIFO-объект, контролируемый атакующим |  |
| 6.5. | Включить защиту от непреднамеренной записи в файл, контролируемый атакующим |  |
| 6.6. | Запретить создание core dump для некоторых исполняемых файлов |  |