ФЕДЕРАЛЬНАЯ СЛУЖБА

ПО ТЕХНИЧЕСКОМУ И ЭКСПОРТНОМУ КОНТРОЛЮ

Утвержден ФСТЭК России

25 декабря 2022 г.

МЕТОДИЧЕСКИЙ ДОКУМЕНТ

РЕКОМЕНДАЦИИ ПО БЕЗОПАСНОЙ НАСТРОЙКЕ ОПЕРАЦИОННЫХ СИСТЕМ LINUX

2022

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Настоящие Рекомендации по безопасной настройке операционных систем Linux (далее – Рекомендации) разработаны в соответствии с подпунктом 4 пункта 8 Положения о Федеральной службе по техническому и экспортному контролю, утверждённого Указом Президента Российской Федерации от 16 августа 2004 г. № 1085.

1.2. Рекомендации определяют содержание работ по настройке операционных систем на базе ядра Linux (далее — операционные системы Linux) и направлены на повышение защищенности информационных (автоматизированных) систем, построенных с использованием указанных операционных систем.

1.3. Настоящие Рекомендации подлежат реализации в государственных информационных системах и на объектах критической информационной инфраструктуры Российской Федерации, построенных с использованием операционных систем Linux, несертифицированных по требованиям безопасности информации, до их замены на сертифицированные отечественные операционные системы.

Настройка сертифицированных операционных систем на базе ядра Linux осуществляется в соответствии с эксплуатационной документацией разработчиков операционных систем.

2. СОДЕРЖАНИЕ РЕКОМЕНДАЦИЙ ПО БЕЗОПАСНОЙ НАСТРОЙКЕ ОПЕРАЦИОННЫХ СИСТЕМ LINUX

При осуществлении безопасной настройки операционных систем Linux выполняются следующие процедуры:

настройка авторизации в операционной системе;

ограничение механизмов получения привилегий;

настройка прав доступа к объектам файловой системы;

настройка механизмов защиты ядра Linux;

уменьшение периметра атаки ядра Linux;

настройка средств защиты пользовательского пространства со стороны ядра Linux.

2.1. Настройка авторизации в операционных системах Linux

При настройке авторизации в операционной системе Linux необходимо:

2.1.1. Не допускать использование учетных записей пользователей с пустыми паролями. Настроить учетные записи таким образом, чтобы каждый пользователь системы либо имел пароль, либо был заблокирован по паролю. В системах Linux данную возможность обеспечивает файл /etc/shadow.

2.1.2. Обеспечить отключение входа суперпользователя в систему по протоколу SSH путём установки для параметра PermitRootLogin значения no в файле /etc/ssh/sshd_config.

2.2. Ограничение механизмов получения привилегий

При ограничении механизмов получения привилегий необходимо:

2.2.1. Обеспечить ограничение доступа к команде su путём добавления в файл /etc/pam.d/su следующей строки: 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>.

2.2.2. Ограничить список пользователей, которым разрешено использовать команду sudo и разрешенных к выполнению через sudo команд путём пересмотра файла /etc/sudoers.

2.3. Настройка прав доступа к объектам файловой системы

При настройке прав доступа к объектам файловой системы необходимо:

2.3.1. Установить корректные права доступа к файлам настройки пользователей, а именно к файлам с перечнями пользовательских идентификаторов (/etc/passwd) и групп (/etc/group), либо хранилищам хешей паролей (в операционных системах GNU/Linux, Solaris, HP-UX: /etc/shadow, AIX: /etc/security/passwd), с помощью команд:

chmod 644 /etc/passwd;

chmod 644 /etc/group;

chmod go-rwx /etc/shadow.

2.3.2. Установить корректные права доступа к файлам запущенных процессов путём выполнения команды вида: chmod go-w /путь/к/файлу для всех исполняемых файлов, запущенных в настоящий момент, и соответствующих библиотек. После этого необходимо осуществить проверку, что директория, содержащая данный файл, а также все родительские директории недоступны для записи непривилегированным пользователям.

2.3.3. Установить корректные права доступа к файлам, выполняющимся с помощью планировщика задач cron неавторизованными пользователями путём выполнения команды chmod go-w путь_к_файлу для каждого файла (либо команды), который вызывается из заданий cron. В противном случае это может привести к выполнению произвольного кода от имени владельца задания cron (в том числе root, что приведет к полной компрометации операционной системы).

2.3.4. Установить корректные права доступа к файлам, выполняемым с помощью sudo путём изменения владельца командой chown root путь_к_файлу для

каждого исполняемого файла, который можно запускать с привилегиями суперпользователя root, но владельцем которого является обычный пользователь и выполнения команды chmod go-w путь_к_файлу для каждого исполняемого файла, который можно запускать с привилегиями суперпользователя root и к которому имеют доступ на запись все пользователи.

2.3.5. Установить корректные права доступа к стартовым скриптам системы путём выполнения команды chmod o-w <filename> к каждому файлу в директориях /etc/rc#.d, а также к файлам .service, присутствующим в системе.

2.3.6. Установить корректные права доступа к системным файлам заданий (конфигурационным файлам) cron при помощи команды chmod go-wx путь_к_файлу_или_директории.

К системным файлам-описаниям очередей cron относятся следующие файлы (могут присутствовать не всегда, в зависимости от операционной 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).

2.3.7. Установить корректные права доступа к пользовательским файлам заданий cron при помощи команды вида: chmod go-w путь_к_файлу_заданий.

2.3.8. Установить корректные права доступа к исполняемым файлам и библиотекам операционной системы путём анализа корректности прав доступа к утилитам и системным библиотекам, расположенным по стандартным путям (/bin, /usr/bin, /lib, /lib64 и другим путям), а также к модулям ядра (для Linux: /lib/modules/версия-текущего-ядра). Местоположение большинства стандартных исполняемых файлов указано в переменной $PATH пользователя root.

2.3.9. Установить корректные права доступа к SUID/SGID-приложениям путём проведения аудита 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. Проверить, 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.

2.3.10. Установить корректные права доступа к содержимому домашних директорий пользователей (.bash_history, .history, .sh_history и т. п. - файлы истории команд оболочек, .bash_profile, .bashrc, .profile, .bash_logout и т. п. - файлы

настройки оболочки, .rhosts - настройки R-подсистем) путём установки на каждый из указанных файлов корректных прав доступа с помощью команды вида: chmod go-rwx путь_к_файлу.

2.3.11. Установить корректные права доступа к домашним директориям пользователей с помощью команды chmod 700 домашняя_директория.

2.4. Настройка механизмов защиты ядра Linux

При настройке механизмов защиты ядра Linux необходимо:

2.4.1. Ограничить доступ к журналу ядра путём установки значения sysctl-опции kernel.dmesg_restrict=1. Журнал ядра должен быть доступен только пользователям, которые обладают разрешением CAP_SYSLOG (администраторы системы).

2.4.2. Заменить ядерные адреса в /proc и других интерфейсах на 0 путём установки значения sysctl-опции kernel.kptr_restrict=2.

2.4.3. Инициализировать динамическую ядерную память нулем при ее выделении путём установки значения 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).

2.4.4. Запретить слияние кэшей ядерного аллокатора путём установки опции загрузки ядра slab_nomerge. Эта настройка позволяет изменить значение опции сборки ядра CONFIG_SLAB_MERGE_DEFAULT при запуске системы (per boot).

2.4.5. Инициализировать механизм IOMMU путём установки значения для следующих опций загрузки ядра:

iommu=force;

iommu.strict=1;

iommu.passthrough=0.

2.4.6. Рандомизировать расположение ядерного стека путём установки значения 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).

2.4.7. Включить средства защиты от аппаратных уязвимостей центрального процессора (для платформы x86) путём установки значения опции загрузки ядра mitigations=auto,nosmt.

2.4.8. Включить защиту подсистемы eBPF JIT ядра Linux путём установки значения sysctl-опции net.core.bpf_jit_harden=2.

2.5. Уменьшение периметра атаки ядра Linux

При уменьшении периметра атаки ядра Linux необходимо:

2.5.1. Отключить устаревший интерфейс vsyscall путём установки значения опции загрузки ядра vsyscall=none. Эта настройка позволяет изменить значение опции сборки ядра CONFIG_LEGACY_VSYSCALL_NONE при запуске системы (per boot).

2.5.2. Ограничить доступ к событиям производительности путём установки значения sysctl-опции kernel.perf_event_paranoid=3.

2.5.3. Отключить монтирование виртуальной файловой системы debugfs путём установки значения опции загрузки ядра debugfs=no-mount (по возможности off).

2.5.4. Отключить системный вызов kexec_load путём установки значения sysctl-опции kernel.kexec_load_disabled=1.

2.5.5. Ограничить использование user namespaces путём установки значения sysctl-опции user.max_user_namespaces=0. Если система на базе Linux не использует user namespaces для выполнения своей задачи, то данная настройка никак не повлияет на работу системы. Рекомендуется предварительно проверить реализацию данной рекомендации на тестовой системе.

2.5.6. Запретить системный вызов bpf для непривилегированных пользователей путём установки значения sysctl-опции kernel.unprivileged_bpf_disabled=1. Если непривилегированные процессы в системе на базе Linux не используют BPF для выполнения своей задачи, данная настройка никак не повлияет на работу системы. Рекомендуется предварительно проверить реализацию данной рекомендации на тестовой системе.

2.5.7. Запретить системный вызов userfaultfd для непривилегированных пользователей путём установки значения sysctl-опции vm.unprivileged_userfaultfd=0.

2.5.8. Запретить автоматическую загрузку модулей ядра, отвечающих за поддержку дисциплины линии терминала путём установки значения sysctl-опции dev.tty.ldisc_autoload=0.

2.5.9. Отключить технологию Transactional Synchronization Extensions (TSX) путём установки значения опции загрузки ядра tsx=off.

2.5.10. Настроить параметр ядра, который определяет минимальный виртуальный адрес, который процессу разрешено использовать для mmap, путём использования sysctl-опции vm.mmap_min_addr = 4096 или больше.

2.5.11. Реализовать рандомизацию адресного пространства, которая защищает от атак на переполнение буфера, путём использования команды после тестирования kernel.randomize_va_space = 2.

2.6. Настройка средств защиты пользовательского пространства со стороны ядра Linux

При настройке средств защиты пользовательского пространства со стороны

ядра Linux необходимо:

2.6.1. Запретить подключение к другим процессам с помощью ptrace путём установки значения sysctl-опции kernel.yama.ptrace_scope=3.

2.6.2. Ограничить небезопасные варианты прохода по символическим ссылкам (symlinks) путём установки значения sysctl-опции fs.protected_symlinks=1. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

2.6.3. Ограничить небезопасные варианты работы с жесткими ссылками (hardlinks) путём установки значения sysctl-опции fs.protected_hardlinks=1. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

2.6.4. Включить защиту от непреднамеренной записи в FIFO-объект путём установки значения sysctl-опции fs.protected_fifos=2. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

2.6.5. Включить защиту от непреднамеренной записи в файл путём установки значения sysctl-опции fs.protected_regular=2. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.

2.6.6. Запретить создание core dump для некоторых исполняемых файлов путём установки значения sysctl-опции fs.suid_dumpable=0. Данная настройка не влияет на нормальную функциональность userspace и блокирует только вредоносное поведение.