Как внедрить стандарты безопасной настройки

и не поругаться с IT

Кирилл Евтушенко,
генеральный директор ООО «Кауч»

Кто мы

- Аудиторы, консалтеры и пентестеры в душе (и трудовых книжках)
- 10+ лет решаем проблемы, связанные с безопасными настройками и их внедрением
- Разрабатываем крутую платформу класса Security Configuration Management
- Амбассадоры дружбы ИБ и ИТ

Зачем настраивать безопасно

- Заставляют
  Регуляторы (например, PCI DSS)…
- Настоятельно рекомендуют
  CIS, NIST, ГОСТ 27001/27002…
- Ломают
  Злые редиски
- Почему бы и нет
  Фундаментальная защита встроенными средствами

Сложно…

- Настроек безопасности **очень** много

  CIS Benchmarks ОС: ~300-400, СУБД: ~200, ППО…
- **ИБ**: не может выполнить задачу безопасной настройки без привлечения ИТ
- **ИТ**: не понимают, за что им это все

- **Хорошо**: просканировать 1 000 настроек на каждом хосте, чтобы оценить уровень защищённости и GAP

- **Не очень хорошо**: предлагать ИТ-администраторам настраивать на каждом хосте в «боевой» эксплуатации 500-1000 настроек сразу

Тише едешь – дальше будешь

❌ Отказываемся от попытки настроить всё сразу: ограничения времени, ресурсов, возможностей…

✅ За системы и их настройку отвечают живые люди: «красный» отчёт на 500 страниц по каждой системе деморализует, а ещё:

- многое непонятно,
- кажется невыполнимым и бесконечным,
- неясно, достигнут ли результат

По частям

- При внедрении стандартов конфигурирования первично формируем сокращенный перечень наиболее критичных настроек по каждой системе, чтобы ИТ-администраторы:
  - понимали, что настройка ограниченного количества параметров займет немного времени,
  - сразу осознавали важность и значимость каждой настройки в стандарте,
  - не реагировали остро на попытки «завалить ИБ-шной» работой,
  - ощущали, что поставленные задачи и цели реалистичны, посильны, конечны.
- Участники процесса должны иметь возможность «ощущать» результат: поставить ✅, закрыть тикет, выполнить KPI, отчитаться, повысить ЧСВ, получить премию…

Сколько?

- 30-50 требований на старте в одном стандарте
- Если есть конкретная «обязаловка» (например, PCI DSS) – берем, чтобы не «налететь» на штрафы и санкции
- Самые критичные и понятные с точки зрения важности требования: парольная политика, доступ, логирование и аудит, шифрование и защита данных и пр.
- Стараемся не засовывать в одно требование большие вложенные куски, например, полсотни параметров аудита

- Быстрая настройка базового перечня наиболее важных настроек на всех или на большинстве систем поможет значимо поднять общий уровень защищенности

Маловато будет. Что дальше?

- Регулярно оцениваем прогресс и обновляем стандарты настройки (заодно выполняем регуляторные требования по пересмотру)
- От 1 раза в 3 месяца до 1 раза в год
- Проводим отдельно оценку по каждому типу систем и группе ИТ-администраторов: все могут двигаться с разной скоростью. Оцениваем динамику
- «Отличникам накидываем» побольше, «отстающим» – поменьше
- Выявляем и устраняем причины проблем в процессе

Друзья по несчастью

- «Обязаловка» по безопасной настройке вызывает реакцию
- Можно начинать с небольших «пилотных проектов»: если с каким-то подразделением в ИТ очень хорошие отношения – идём к ним
- Следующие на очереди:
  - секьюрити чемпионы,
  - энтузиасты,
  - гики,
  - эксперты с активной жизненной позицией,
- Успешный пример поможет с меньшими усилиями и более эффективно вовлечь в процесс другие подразделение и ИТ-руководителей

Хотим как лучше

- Согласовываем стандарты с конкретными администраторами
- Проводим тестирование совместно с ИТ: настраиваем тестовые системы и выборку «боевых», идентифицируем и обрабатываем отклонения, собираем обратную связь
- Делаем вместе «рабочие» стандарты, копипаст из отчетов, бенчмарков и стэкэксченьджа «не полетит»
- Объясняем с точки зрения ИБ каждое требование, которое вызывает вопросы
- Разбираемся вместе в проблемах и «трудных местах»
- Не конфликтуем и получаем удовольствие от общего благого дела

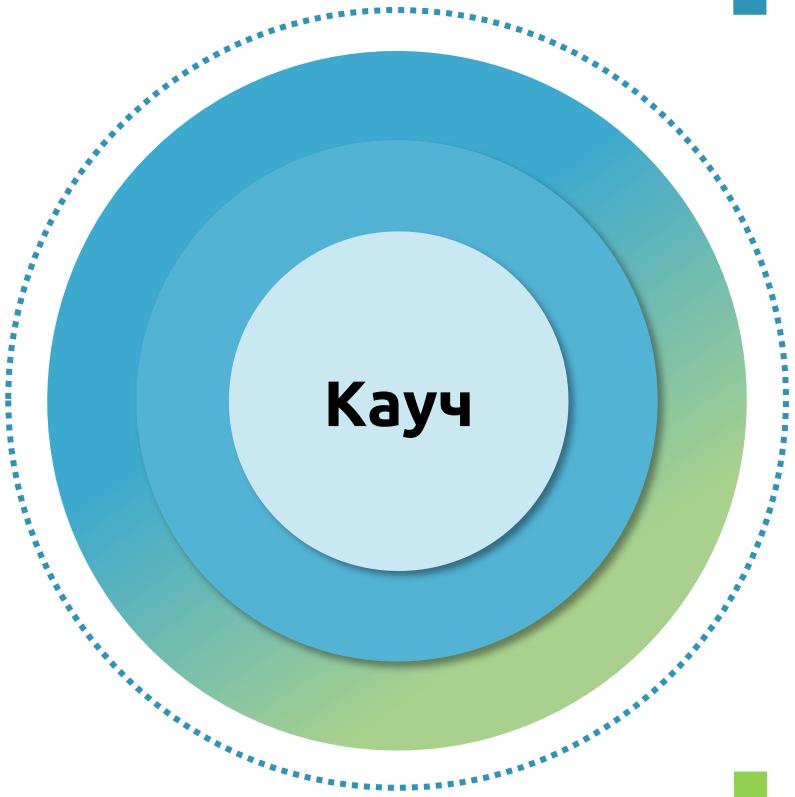
Хорошая новость

Мы разработали платформу управления безопасностью настроек!

И активно её развиваем!

Кауч

All-in-one платформа для управления безопасностью конфигураций



Сканирование на соответствие лучшим мировым практикам

Совместная работа и разделение задач

Настройка ресурсов инструментарием «из коробки»

Создание политик любой сложности

Мониторинг изменений

Единый процесс

Управление безопасностью конфигураций включено в общий процесс мониторинга и реагирования на инциденты

Couch

Объекты

Сканирования

Профили

Пользователи

Настройки

Инфо

Сканирование

Сканирования / RedOS Couch — 2.04.2025 17:57 / #15077

События ↗ PDF, XLSX, HTML

#15077 at 02.04.2025 17:57:03 UTC+03

Профиль: СБК-РедОС-04 (вер. 7)

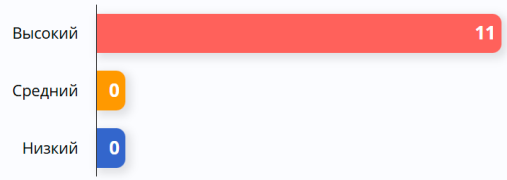
Ресурс: RedOS Couch

Компонент: RED OS 7.3

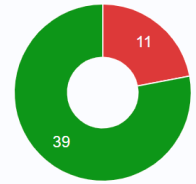
Адрес: 10.0.50.7:22

Запустить снова | Скрипт конфигурации | Удалить

Риски



Соответствие



# Заголовок	Описание	Устранение	Риск	Оценка	Значение (Искать... / Развернуть ↓)
1.1. Минимальная длина паролей	Эта настройка ограничивает минимально допустимую длину паролей учетных записей в символах. Типы атак на пароли включают атаки по словарю (с использованием распространенных слов и фраз) и атаки методом перебора (по всем	Общая часть конфигурации: В файле /etc/security/pwquality.conf укажите ограничение минимальной длины паролей в соответствии с требуемым значением с помощью параметра minlen. Пример: minlen = 12 или более символов	Высокий	✓	password requisite pam_pwquality.so retry=3 try_first_pass local_users_only↵ ↵ password requisite pam_pwquality.so retry=3 try_first_pass
1.2. Сложность паролей	Этот параметр задает требование использования в пароле символов разных типов (заглавных и строчных букв, цифр, специальных символов). Пароли должны содержать минимум по одному символу из каждой из четырех категорий:	Общая часть конфигурации: В файле /etc/security/pwquality.conf активируйте требование использования в пароле четырех различных типов символов с помощью параметра 'minclass' или параметров 'dcredit', 'lcredit', 'ocredit', 'ucredit'. Пример с параметром	Высокий	✓	password requisite pam_pwquality.so retry=3 try_first_pass local_users_only↵ ↵ password requisite pam_pwquality.so retry=3 try_first_pass
1.3. Максимальный срок действия паролей	Этот параметр определяет срок действия пароля после его создания. После истечения срока действия пароль не может быть использован для доступа к системе и должен быть заменен. Чем дольше существует пароль, тем выше вероятность того, что он будет	Настройте параметр PASS_MAX_DAYS в файле /etc/login.defs в соответствии с требуемым значением. Данная настройка действует на новые создаваемые учетные записи. Для существующих учетных записей с правом интерактивного входа активируйте ограничение	Высокий	✗	PASS_MAX_DAYS 90↵ ↵ root: Maximum number of days between password change : 999999↵ couch: Maximum number of days between password change : 999999
1.4. Предупреждение о предстоящем истечении срока действия пароля	Этот параметр определяет, за какое время пользователи будут предупреждены об истечении срока действия их пароля. Предупреждение пользователей о предстоящем истечении срока действия пароля необходимо для его своевременного обновления с целью	Настройте параметр PASS_WARN_AGE в файле /etc/login.defs в соответствии с требуемым значением. Данная настройка действует на новые создаваемые учетные записи. Для существующих учетных записей с правом интерактивного входа активируйте	Средний	✓	PASS_WARN_AGE 7↵ ↵ root: Number of days of warning before password expires : 7↵ couch: Number of days of warning before password expires : 7
1.5. История паролей	Этот параметр определяет количество обновленных уникальных паролей, которые необходимо связать с учетной записью пользователя, прежде чем для нее можно будет повторно использовать старый пароль. Если требуется изменить пароль, но повторное	Конфигурация в случае использования authselect: В файле /etc/security/pwhistory.conf укажите ограничение повторного использования предыдущих паролей в соответствии с требуемым значением с помощью параметра	Высокий	✓	password required pam_pwhistory.so remember=5
1.6. Блокировка учетной записи при последовательных неуспешных попытках аутентификации	Параметры блокировки учетной записи определяют количество неудачных попыток аутентификации, после которых учетная запись будет заблокирована, и время, на которое она будет заблокирована. Активация блокировки учетной записи после	Конфигурация в случае использования authselect: Если конфигурация модулей PAM осуществляется в системе с помощью authselect и модуль pam_faillock.so не активирован, активируйте его с помощью команды:	Высокий	✓	/etc/pam.d/system-auth:auth required pam_faillock.so preauth silent↵ /etc/pam.d/system-auth:auth required pam_faillock.so authfail↵ /etc/pam.d/system-auth:account required pam_faillock.so↵

Сканирование

#15077 at 02.04.2025 17:57:03 UTC+03

Профиль:
СБК-РедОС-04 (вер. 7)

Ресурс:
RedOS Couch

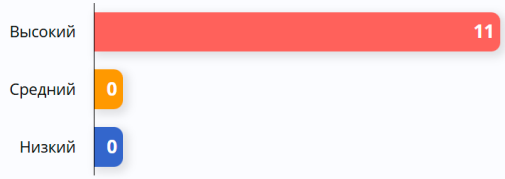
Компонент:
RED OS 7.3

Адрес:
10.0.50.7:22

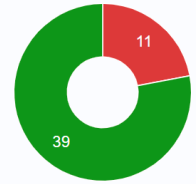
Запустить снова | Скрипт конфигурации | Удалить

Скачать выбранные | Отмена

Риски

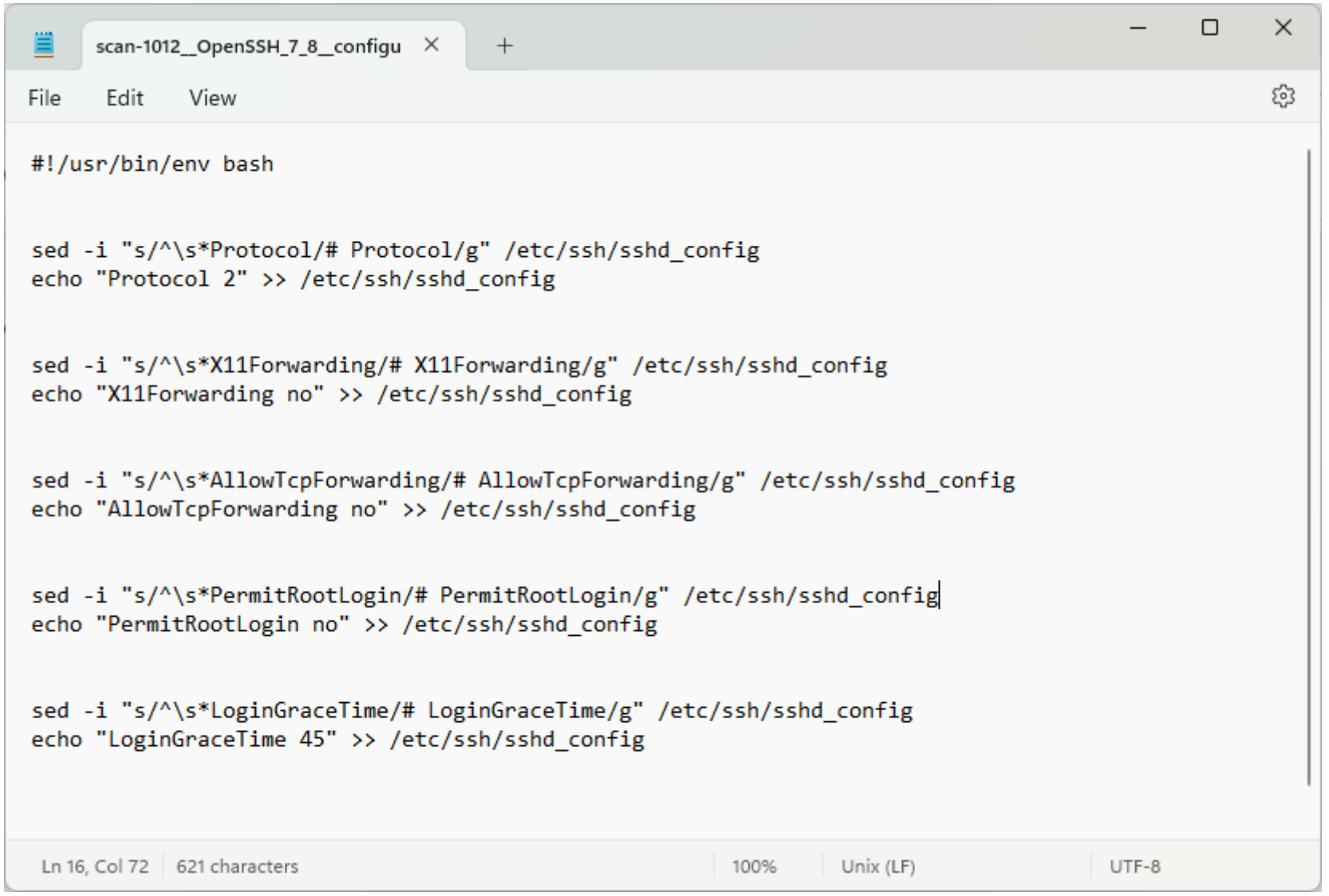


Соответствие

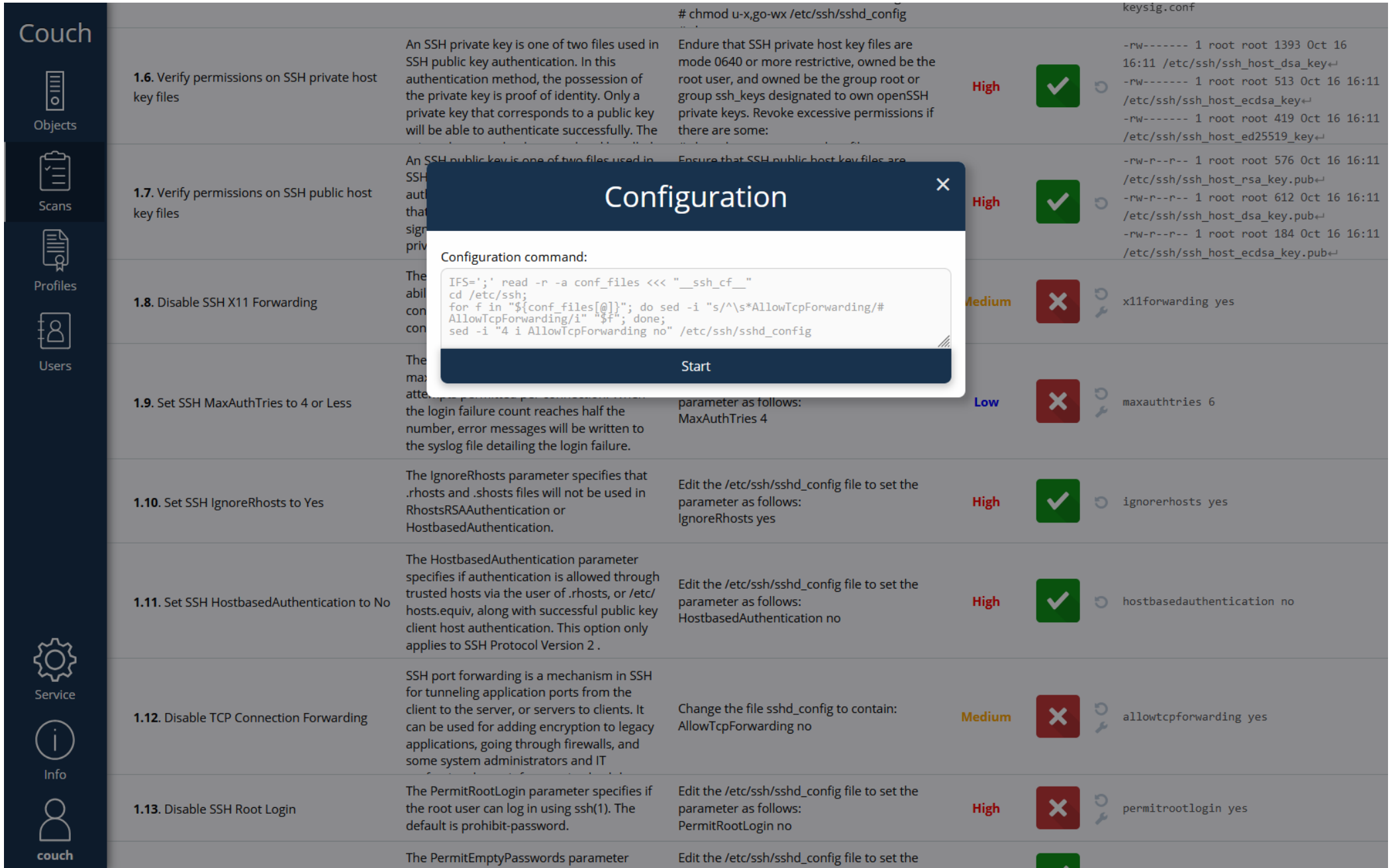


# Заголовок	Описание	Устранение	Риск	Оценка	Значение
1.3. Максимальный срок действия паролей	Этот параметр определяет срок действия пароля после его создания. После истечения срока действия пароль не может быть использован для доступа к системе и должен быть заменен. Чем дольше существует пароль, тем выше вероятность того, что он будет	Настройте параметр PASS_MAX_DAYS в файле /etc/login.defs в соответствии с требуемым значением. Данная настройка действует на новые создаваемые учетные записи. Для существующих учетных записей с правом интерактивного входа активируйте ограничение	Высокий	✗	PASS_MAX_DAYS 90↵ ↵ root: Maximum number of days between password change : 999999↵ couch: Maximum number of days between password change : 999999
8.1. Инициализация памяти нулями при выделении	Активированный параметр init_on_alloc заставляет ядро инициализировать вновь выделенные страницы и объекты в куче нулями. Инициализация выполняется в тех местах, где выполняются проверки __GFP_ZERO. По умолчанию для init_on_alloc установлено	Добавьте параметр загрузки ядра Linux в конфигурацию загрузчика в соответствии с требуемым значением. В системах, поддерживающих команду 'grubby --update-kernel', для изменения параметров загрузки ядра может быть использована	Высокий	✗	BOOT_IMAGE=/vmlinuz-5.15.72-1.el7.3.x86_64 root=/dev/mapper/ro_redos-root ro resume=/dev/mapper/ro_redos-swap rd.lvm.lv=ro_redos/root rd.lvm.lv=ro_redos/swap rhgb quiet↵ ↵
8.2. Рандомизация смещения стека ядра Linux	Параметр randomize_kstack_offset доступен с версии ядра Linux 5.13 и позволяет рандомизировать смещение стека ядра для каждого системного вызова, чтобы затруднить злоумышленникам атаки на ядро на основе стека для тех эксплойтов, которые полагаются	Добавьте параметр загрузки ядра Linux в конфигурацию загрузчика в соответствии с требуемым значением. В системах, поддерживающих команду 'grubby --update-kernel', для изменения параметров загрузки ядра может быть использована	Высокий	✗	BOOT_IMAGE=/vmlinuz-5.15.72-1.el7.3.x86_64 root=/dev/mapper/ro_redos-root ro resume=/dev/mapper/ro_redos-swap rd.lvm.lv=ro_redos/root rd.lvm.lv=ro_redos/swap rhgb quiet↵ ↵
8.3. Меры по устранению уязвимостей CPU	Параметр ядра mitigations – это метапараметр, который управляет дополнительными мерами по устранению уязвимостей CPU. Это набор тщательно подобранных, независимых от архитектуры [X86,PPC,S390,ARM64] параметров, каждый из которых представляет собой	Добавьте параметр загрузки ядра Linux в конфигурацию загрузчика в соответствии с требуемым значением. В системах, поддерживающих команду 'grubby --update-kernel', для изменения параметров загрузки ядра может быть использована	Высокий	✗	BOOT_IMAGE=/vmlinuz-5.15.72-1.el7.3.x86_64 root=/dev/mapper/ro_redos-root ro resume=/dev/mapper/ro_redos-swap rd.lvm.lv=ro_redos/root rd.lvm.lv=ro_redos/swap rhgb quiet↵ ↵
8.4. Отключение vsyscall	Виртуальные системные вызовы являются устаревшим предшественником вызовов vDSO. И vsyscall=native, и vsyscall=emulate (значением по умолчанию) оказывают негативное влияние на безопасность. Механизм vsyscall является устаревшим по	Добавьте параметр загрузки ядра Linux в конфигурацию загрузчика в соответствии с требуемым значением. В системах, поддерживающих команду 'grubby --update-kernel', для изменения параметров загрузки ядра может быть использована	Высокий	✗	BOOT_IMAGE=/vmlinuz-5.15.72-1.el7.3.x86_64 root=/dev/mapper/ro_redos-root ro resume=/dev/mapper/ro_redos-swap rd.lvm.lv=ro_redos/root rd.lvm.lv=ro_redos/swap rhgb quiet↵ ↵
8.5. Отключение debugfs	Файловая система debugfs — это простая в использовании файловая система на базе оперативной памяти, специально разработанная для целей отладки. Это простой способ для разработчиков ядра сделать	Добавьте параметр загрузки ядра Linux в конфигурацию загрузчика в соответствии с требуемым значением. Если нет возможности полностью отключить функцию, достаточно заблокировать возможность монтирования	Высокий	✗	BOOT_IMAGE=/vmlinuz-5.15.72-1.el7.3.x86_64 root=/dev/mapper/ro_redos-root ro resume=/dev/mapper/ro_redos-swap rd.lvm.lv=ro_redos/root rd.lvm.lv=ro_redos/swap rhgb quiet↵

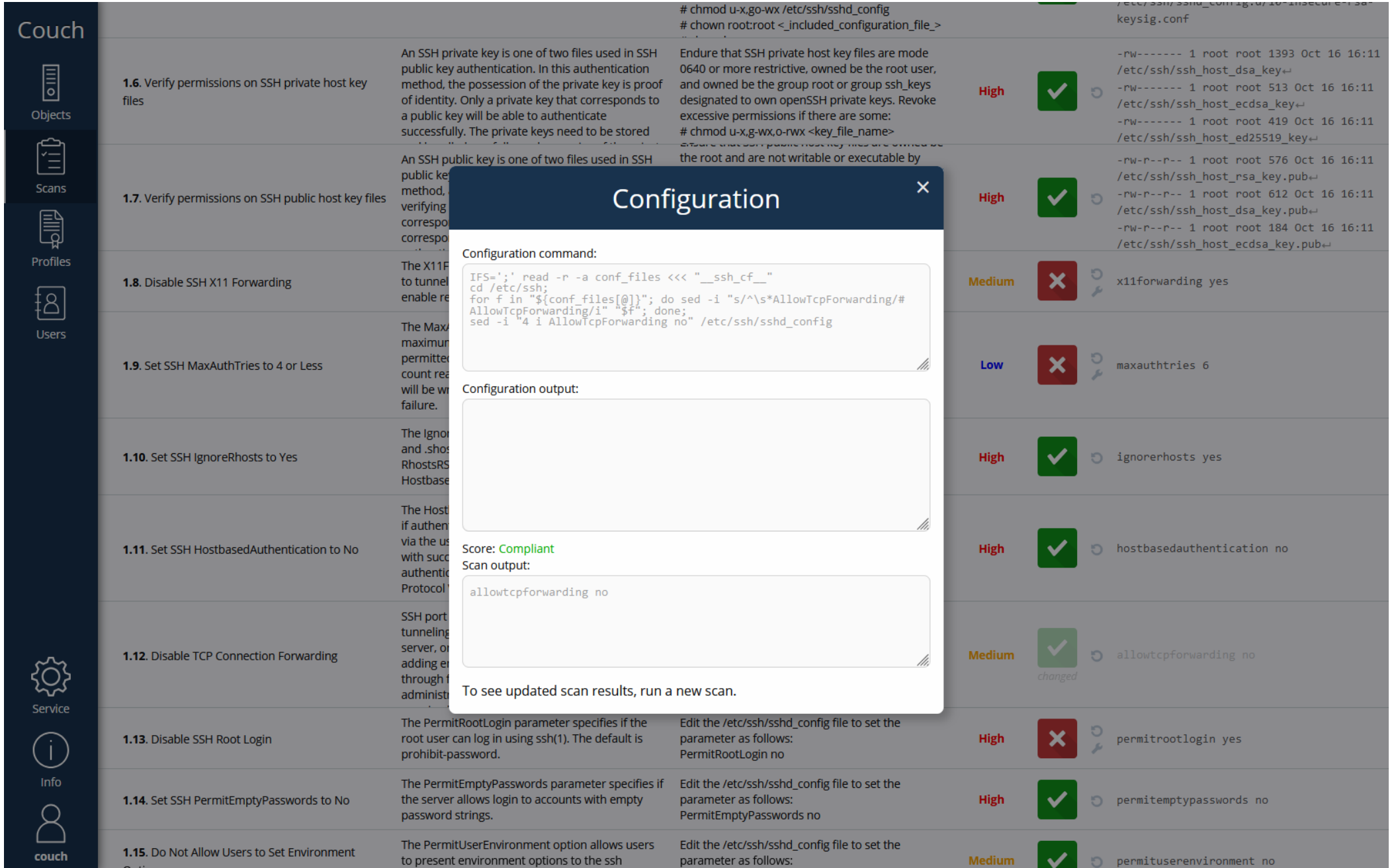
Настройка


scan-1012__OpenSSH_7_8__configu
File
Edit
View
#!/usr/bin/env bash
sed -i "s/^\s*Protocol/# Protocol/g" /etc/ssh/sshd_config
echo "Protocol 2" >> /etc/ssh/sshd_config
sed -i "s/^\s*X11Forwarding/# X11Forwarding/g" /etc/ssh/sshd_config
echo "X11Forwarding no" >> /etc/ssh/sshd_config
sed -i "s/^\s*AllowTcpForwarding/# AllowTcpForwarding/g" /etc/ssh/sshd_config
echo "AllowTcpForwarding no" >> /etc/ssh/sshd_config
sed -i "s/^\s*PermitRootLogin/# PermitRootLogin/g" /etc/ssh/sshd_config
echo "PermitRootLogin no" >> /etc/ssh/sshd_config
sed -i "s/^\s*LoginGraceTime/# LoginGraceTime/g" /etc/ssh/sshd_config
echo "LoginGraceTime 45" >> /etc/ssh/sshd_config
Ln 16, Col 72
621 characters
100%
Unix (LF)
UTF-8

Настройка: онлайн



Настройка: онлайн



Хотите демонстрацию Couch?

Оставьте заявку на сайте:

Или свяжитесь с нами по адресу
info@couch.ru