кауч

Управление уязвимостями конфигураций: импортозамещаем nessus с умом

Спикер:
Кирилл Евтушенко,
генеральный директор Кауч

кауч

.audit – основной способ проверок настроек в Nessus

```
<check_type:"Unix">

  <if>
    <condition type:"AND">
      <custom_item>
        type        : CMD_EXEC
        description : "Check if Linux kernel installed"
        cmd         : "uname -s"
        expect      : "^(?i)Linux"
      </custom_item>
    </condition>

    <then>

      <if>
        <condition type:"OR">

          <custom_item>
            type        : CMD_EXEC
            description : "Check if OpenSSH enabled"
            cmd         : "systemctl is-enabled sshd-server 2>/dev/null"
            expect      : "enabled"
          </custom_item>

          <custom_item>
            type        : CMD_EXEC
            description : "Check sshd version"
            cmd         : "sshd -v 2>/dev/null"
            expect      : "(?i)sshd"
          </custom_item>

        </condition>

        <then>
            <custom_item>
            type        : CMD_EXEC
            description : "Check SSH LogLevel via sshd -T and first LogLevel in main sshd_config"
            info        : "Checks the effective LogLevel from sshd -T and the first LogLevel directive found in the main config file only."
            solution    : "Set LogLevel INFO or VERBOSE in the main sshd_config file."
            cmd         : "{
                if [[ $(cat /etc/*-release 2>/dev/null | grep '^NAME' | tr -d '\"') =~ ALT$ ]]; then
                    ...
                fi
            }"
            expect      : "PASS"
            </custom_item>
```

кауч

Декларативный подход

- DSL (domain-specific language) Tenable
- Текстовый формат с «псевдо-блочной» структурой
- Похоже на смесь:
  - INI
  - DSL
  - Описания конфигураций

- Типы проверок:
  - FILE_CONTENT_CHECK
  - SERVICE_CHECK
  - REG_CHECK
  - FILE_CHECK
  - …

  - AUDIT_POWERSHEL
  - **CMD_EXEC**

кауч

Пример: простой конфиг openssh

```
1  # This is the sshd server system-wide configuration file.  See
2  # sshd_config(5) for more information.
3
4  Port 22
5  AddressFamily any
6  ListenAddress 0.0.0.0
7  ListenAddress ::
8
9  HostKey /etc/ssh/ssh_host_rsa_key
10 HostKey /etc/ssh/ssh_host_ecdsa_key
11 HostKey /etc/ssh/ssh_host_ed25519_key
12
13 # LogLevel
14 LogLevel verbose
15
16 # Ciphers and keying
17 RekeyLimit default none
18
19 ...
```

кауч

Пример: openssh – audit

```
<custom_item>
  type        : FILE_CONTENT_CHECK
  description : "Ensure SSH LogLevel is set to INFO or VERBOSE"
  file        : "/etc/ssh/sshd_config"
  regex       : "^[[:space:]]*LogLevel[[:space:]]+(INFO|VERBOSE)"
</custom_item>
```

Или

```
<custom_item>
  type        : CMD_EXEC
  description : "Ensure SSH LogLevel is set to INFO or VERBOSE"
  cmd         : "sshd -T 2>/dev/null | grep -i '^loglevel'"
  expect      : "(?i)loglevel[[:space:]]+(info|verbose)"
</custom_item>
```

couch.ru

кауч

openssh – audit сына маминой подруги

```
...
cmd : "{
    if [[ $(cat /etc/*-release 2>/dev/null | grep '^NAME' | tr -d '\"') =~ ALT$ ]]; then
        base_dir='/etc/openssh'
    else
        base_dir='/etc/ssh'
    fi

    config_file=\"$base_dir/sshd_config\"
    status=1
    output=''

    sshd_t_conf=$(sshd -T 2>&1)
    if [[ $? -ne 0 ]]; then
        output+=\"sshd -T error: $sshd_t_conf\"$'\\n'
        status=0
    else
        output+='sshd -T output:'$'\\n'
        while IFS= read -r line; do
            line=$(printf '%s' \"$line\" | sed 's/^[[:space:]]*//; s/[[:space:]]*$//')
            if [[ \"$line\" =~ ^loglevel[[:space:]]+([^[:space:]]+)$ ]]; then
                value_lc=$(printf '%s' \"${BASH_REMATCH[1]}\" | tr '[:upper:]' '[:lower:]')
                output+=\"  $line\"$'\\n'
                if [[ \"$value_lc\" != 'info' && \"$value_lc\" != 'verbose' ]]; then
                    status=0
                fi
            fi
        done <<< \"$sshd_t_conf\"
    fi

    output+=$'\\nmain config:'$'\\n'

    first_param=''
    while IFS= read -r line; do
        stripped=$(printf '%s' \"$line\" | sed 's/^[[:space:]]*//; s/[[:space:]]*$//')
        [[ -z \"$stripped\" ]] && continue
        [[ \"$stripped\" =~ ^# ]] && continue

        if [[ \"$stripped\" =~ ^[Ll][Oo][Gg][Ll][Ee][Vv][Ee][Ll][[:space:]]+([^[:space:]]+)$ ]]; then
            first_param=\"$stripped\"
            value_lc=$(printf '%s' \"${BASH_REMATCH[1]}\" | tr '[:upper:]' '[:lower:]')
            output+=\"  $stripped\"$'\\n'
            if [[ \"$value_lc\" != 'info' && \"$value_lc\" != 'verbose' ]]; then
                status=0
            fi
            break
        fi
    done < \"$config_file\"

    if [[ -z \"$first_param\" ]]; then
        output+='  LogLevel directive not found in main config file'$'\\n'
    fi

    printf '%d' \"$status\"
}"
expect : "^1$"
...
```

кауч

openssh: повышаем обороты

```
1  # This is the sshd server system-wide configuration file.  See
2  # sshd_config(5) for more information.
3
4  Port 22
5  AddressFamily any
6  ListenAddress 0.0.0.0
7  ListenAddress ::
8
9  include /etc/ssh/sshd_config.d/10-extra.conf
10
11 # LogLevel
12 LogLevel verbose
13
14 HostKey /etc/ssh/ssh_host_rsa_key
15 HostKey /etc/ssh/ssh_host_ecdsa_key
16 HostKey /etc/ssh/ssh_host_ed25519_key
17
18 include /etc/ssh/sshd_config.d/20-extra.conf
19
20 # Ciphers and keying
21 RekeyLimit default none
22 ...
```

кауч

openssh: еще выше

```
/etc/ssh/sshd_config.d/10-extra.conf:

LogLevel info
DisableForwarding yes
PubKeyAuthentication yes

Match User DevUser
  LogLevel Info
  Include /etc/ssh/match_includes.d/30-extra.conf

Match User AdminUser Address 10.10.10.5
    PermitRootLogin yes
    AllowTcpForwarding yes
    LogLevel fatal
```

openssh: game over

```
87                  while IFS= read -r line; do
88                      if [[ \"$line\" =~ ^[[:space:]]*[Mm]atch[[:space:]] ]]; then
89                          include_type='IN-MATCH'
90                      fi
91
92                      if [[ \"$line\" =~ ^[[:space:]]*[Ii]nclude[[:space:]] ]]; then
93                          local included_files=()
94                          local g search_dir each real_each
95                          local -a tmp=()
96
97                          while IFS= read -r g; do
98                              [[ -z \"$g\" ]] && continue
99
100                             if [[ $g != /* ]]; then
101                                 g=\"$base_dir/$g\"
102                             fi
103
104                             search_dir=$(dirname \"${g%%[*?\[]*}\")
105                             [[ -d \"$search_dir\" ]] || continue
106
107                             mapfile -d '' tmp < <(find \"$search_dir\" -path \"$g\" -print0 2>/dev/null)
108
109                             for each in \"${tmp[@]}\"; do
110                                 real_each=$(canon_path \"$each\")
111                                 included_files+=(\"$real_each\")

345             if [[ \"$param_global_found\" -eq 0 ]]; then
346                 output+=\$'\\n\\nLogLevel directive not found in global config. Base value: loglevel INFO\\n'
347             fi
348
349             printf '%d' \"$status\"
350         }"
351         expect      : "^1$"
352     </custom_item>
```

кауч

Недостатки .audit

- Нет:
  - Циклов
  - Функций
  - Удобных условий
  - Объектов
  - Связей между проверками
  - Инклудов/модулей/переиспользования
  - Разбора структурированных форматов

- Классический для сканеров match-подход

- Проблемы с:
  - include
  - override
  - runtime/детекты

кауч

Что приходится делать

- Писать очень-очень много кода на баше или пауэршелле
- Задействовать установленные на проверяемых серверах утилиты (и даже устанавливать их специально)
  - Парсинг XML, JSON, YAML и пр. (иначе – grep 😱)
  - Подключение к БД или другому ПО
- Вызывать локальные интерпретаторы (python, perl...) для облегчения bash-нагрузки

Проблемы: отсутствие (или присутствие 🙃) нужных прав, необходимых утилит, конкретных версий ПО/библиотек и пр.

кауч

Не все проблемы

- Compliance Policy / Scan может состоять из нескольких audit-файлов, но каждый «коннектор» в отдельном audit-файле
  - Файлы аудита не связаны между собой
  - Сложно писать проверки на нескольких уровнях
    (СУБД: есть параметры в БД и ФС)
- Проблемы при работе с неизвестным Nessus прикладным ПО, требующим аутентификации: сложности с хранением и передачей учетных данных (нужен NASL, возможности ограничены)
- Проверки де-факто выполняются на стороне сервера

кауч

NASL

- Высокий порог входа/нужно учить язык, мало документации
- Сложное описание логики проверок, контекст (КВ, креды, типы проверок)
- Часто, фактически, обертка над баш-кодом
- Логика на NASL часто получается сложнее, чем на баше
- Минимум возможностей по отладке и обработке ошибок
- Формально умеет, но на деле неполноценно работает со структурированными форматами
- Уступает современным языкам программирования примерно во всем

couch.ru

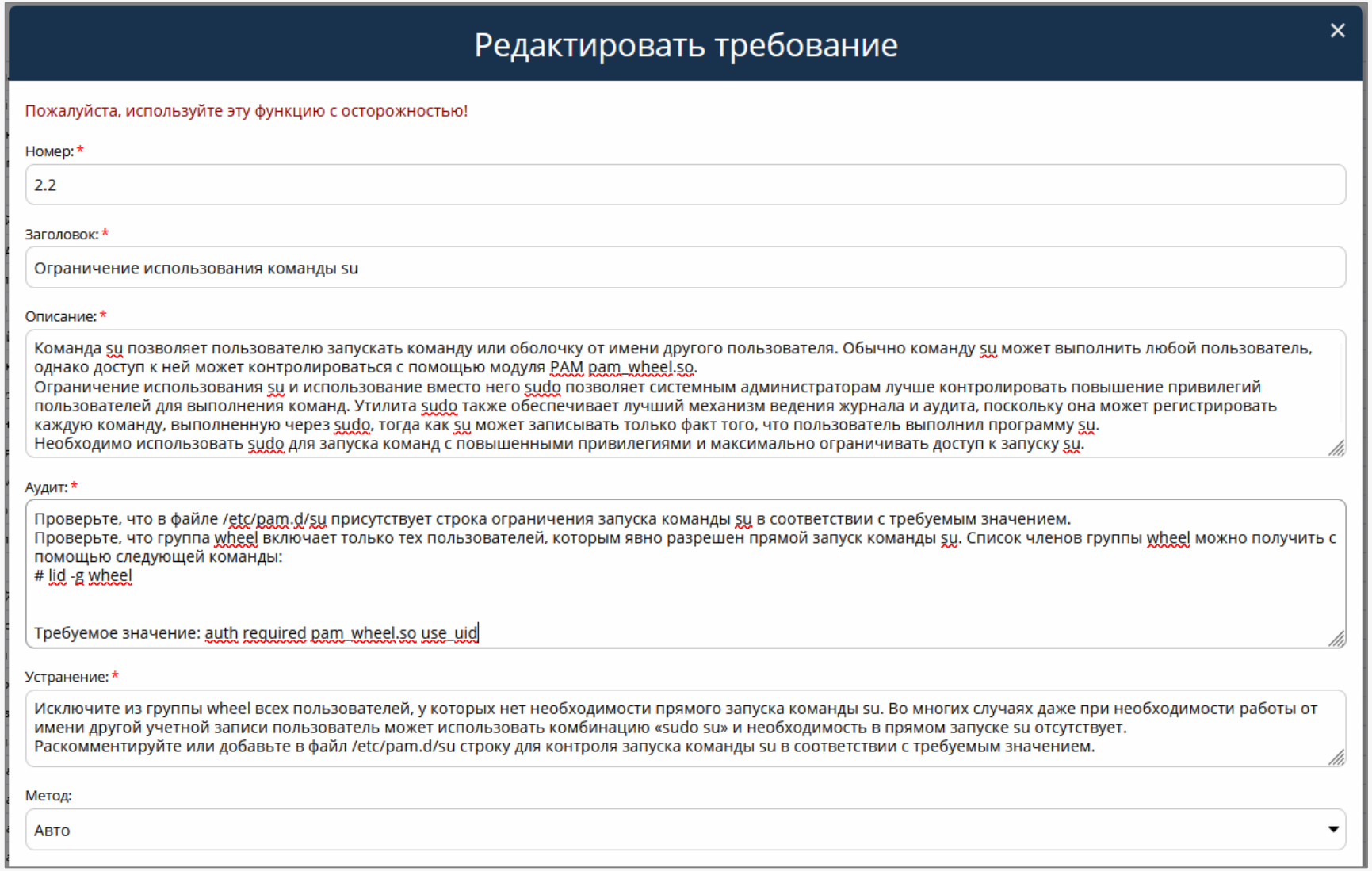
кауч

А теперь переезжаем со всем этим багажом

couch.ru

кауч

Проверки в Кауче: наглядно



кауч

Риск:

Высокий

Команда сбора:

```
egrep "^\s*auth\s+required\s+pam_wheel\.so" /etc/pam.d/su 2>&1
```

Скрипт валидации:* ? тест

```
def check(text):
    x = re.split(r'_#\[\r?\n?', text)
    return int(bool(re.search('^\s*auth\s+required\s+pam_wheel\.so\s(.*\s)?use_uid(\s|#|$)', x[0], re.MULTILINE))), x[0] or 'pam_wheel.so in not configured'
```

Команда конфигурации:

```
PTF=/etc/pam.d/su;
grep -E '^\s*auth\s+required\s+pam_wheel\.so' $PTF || sed -ri '0,/^\s*auth\s+sufficient\s+pam_rootok\.so/s/^\s*auth\s+sufficient\s+pam_rootok\.so.*$/&\nauth required pam_wheel.so use_uid/' $PTF;
[[ -z $(grep -E '^\s*auth\s+required\s+pam_wheel\.so.*\s+use_uid(\s|#|$)' $PTF) ]] && sed -ri 's/^\s*(auth\s+required\s+pam_wheel\.so.*)$/\1 use_uid/' $PTF || true;
```

Копировать

Сохранить

couch.ru

кауч

Кауч

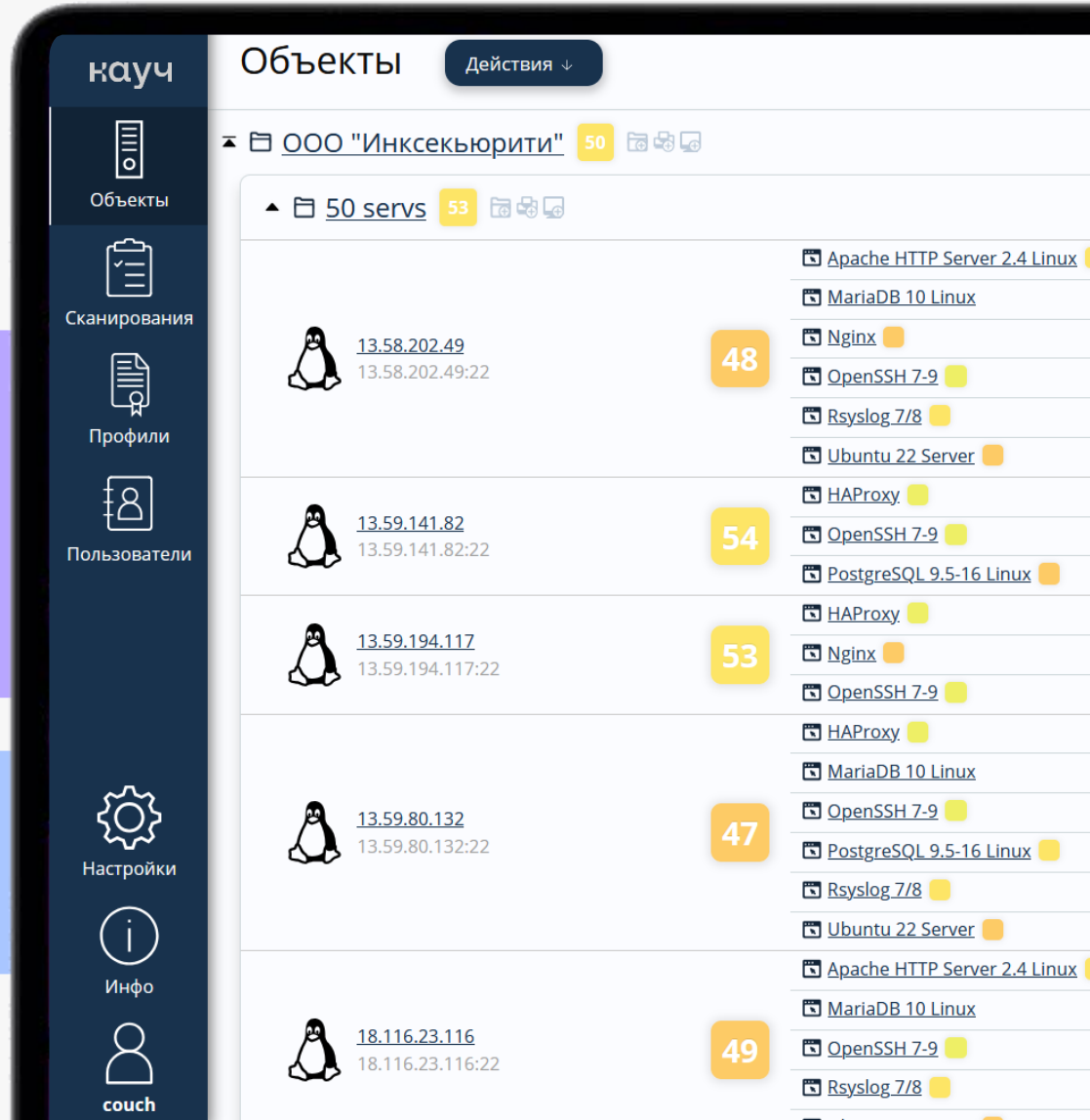
All-in-one платформа для управления безопасностью конфигураций ↗

Единая среда для ИБ и ИТ и всех операций, связанных с управлением безопасностью настроек ИТ-ресурсов

Автоматизированная настройка ресурсов по клику или офлайн с помощью готовых команд и скриптов, доступных «из коробки»

Проверка соответствия ресурсов по клику, расписанию или офлайн с помощью мощного встроенного сканера конфигураций

Гибкое и простое создание политик безопасности любой сложности



couch.ru

кауч

Приходите к нам на стенд...

...или на сайт couch.ru

Вот с кем мы уже работаем

Райффайзен Банк
ПСБ
БСПБ
МОСОБЛБАНК
АТБ Азиатско-Тихоокеанский Банк
МЕТКОМБАНК
axenix
СБЕР НПФ
СБЕР РЕШЕНИЯ
MR GROUP
СТОЛОТО
ERG