Автоматическое добавление активов

Развёртывание скрипта импорта активов

Инструкция описывает установку скрипта реагирования на хост коррелятора и последующую настройку в веб-интерфейсе KUMA.

Что делает скрипт

Скрипт принимает два аргумента — тенант и связку `IP|FQDN` — формирует запрос к API KUMA и импортирует актив (`POST /assets/import`) в указанный тенант.

Часть 1. Подготовка скрипта на хосте коррелятора

kuma_import_asset.sh

```
#!/usr/bin/env bash
#
# Импорт актива в KUMA через POST /api/v3/assets/import
#
# Использование:
#   ./kuma_import_asset.sh <tenantID> "<ip>|<fqdn>"
#
# Второй аргумент — IP и/или FQDN через разделитель '|'. Допустимо:
#   "1.2.3.4|host.example.com"   -> и IP, и FQDN
#   "1.2.3.4"  или  "1.2.3.4|"   -> только IP
#   "|host.example.com"          -> только FQDN
#
set -euo pipefail
```

```
# ============================ КОНСТАНТЫ ============================
KUMA_HOST=""          # адрес системы (KUMA Core)
KUMA_PORT="7223"      # порт REST API
KUMA_TOKEN=""         # токен доступа (Bearer)

#   1 = не проверять сертификат (curl -k)
#   0 = проверять (нужен доверенный CA)
INSECURE=1
# ==================================================================

usage() {
    echo "Использование: $0 <tenantID> \"<ip>|<fqdn>\"" >&2
    exit 1
}

# ---------------------------- Аргументы ----------------------------
tenant="${1:-}"
device="${2:-}"

[[ -z "$tenant" ]] && { echo "Ошибка: не указан тенант (аргумент 1)." >&2; usage; }

IFS='|' read -r ip fqdn <<< "$device"

trim() { local s="$1"; s="${s#"${s%%[![:space:]]*}"}"; s="${s%"${s##*[![:space:]]}"}";
printf '%s' "$s"; }
ip="$(trim "$ip")"
fqdn="$(trim "$fqdn")"

if [[ -z "$ip" && -z "$fqdn" ]]; then
    echo "Ошибка: нужно указать хотя бы IP или FQDN." >&2
    usage
fi

json_escape() {
    local s="$1"
    s="${s//\\/\\\\}"
    s="${s//\"/\\\"}"
    printf '%s' "$s"
```

```
}

parts=()
[[ -n "$ip"   ]] && parts+=("\"ipAddresses\":[\"$(json_escape "$ip")\"]")
[[ -n "$fqdn" ]] && parts+=("\"fqdn\":[\"$(json_escape "$fqdn")\"]")

asset="{$(IFS=,; printf '%s' "${parts[*]}")}"
body="$(printf '{"tenantID":"%s","assets":[%s]}' "$(json_escape "$tenant")" "$asset")"

url="https://${KUMA_HOST}:${KUMA_PORT}/api/v3/assets/import"

curl_opts=(
    -sS
    -w $'\n%{http_code}'
    -X POST "$url"
    -H "Authorization: Bearer ${KUMA_TOKEN}"
    -H "Content-Type: application/json"
    --data "$body"
)
[[ "$INSECURE" == "1" ]] && curl_opts+=(-k)

echo "Отправка в тенант '$tenant' (ip='${ip:-—}', fqdn='${fqdn:-—}')..."

response="$(curl "${curl_opts[@]}")"
http_code="${response##*$'\n'}"
payload="${response%$'\n'*}"

echo "HTTP ${http_code}"
echo "${payload}"

if [[ "$http_code" == "200" ]]; then
    echo "Готово: актив импортирован."
else
    echo "Импорт не выполнен." >&2
    exit 1
fi
```

1.1. Заполнить константы в скрипте

Перед размещением откройте скрипт и задайте значения констант в начале файла:

- `KUMA_HOST` — адрес системы (KUMA Core);
- `KUMA_TOKEN` — токен доступа к API.

Порт (`KUMA_PORT=7223`) уже задан по умолчанию и в стандартной конфигурации менять его не нужно.

> **Требования к токену.** Токен должен принадлежать пользователю, у которого есть права на импорт активов — операцию `POST /assets/import`. Такими правами обладают роли:
> «Главный администратор», «Администратор тенанта», «Аналитик 2 линии», «Аналитик 1 линии».
> Без соответствующих прав API вернёт ошибку доступа (403) и актив не импортируется.

1.2. Определить ID коррелятора

ID коррелятора можно посмотреть в веб-интерфейсе системы в разделе активных сервисов (**Ресурсы→ Активные сервисы**). Он потребуется для пути размещения скрипта.

1.3. Разместить скрипт

Скопируйте скрипт в каталог `scripts` нужного коррелятора, подставив его ID:

```
/opt/kaspersky/kuma/correlator/<id_коррелятора>/scripts/
```

1.4. Назначить владельца

Владельцем скрипта должен быть пользователь и группа `kuma`:

```
chown kuma:kuma /opt/kaspersky/kuma/correlator/<id_коррелятора>/scripts/kuma_import_asset.sh
```

1.5. Выдать права на выполнение

```
chmod +x /opt/kaspersky/kuma/correlator/<id_коррелятора>/scripts/kuma_import_asset.sh
```

После этого скрипт готов к запуску коррелятором.

Часть 2. Настройка в веб-интерфейсе KUMA

2.1. Проверить привязку правил корреляции

Убедитесь, что к коррелятору привязаны следующие правила:

- `[OOTB][Net] Detection unknown assets in destination fields`
- `[OOTB][Net] Detection unknown assets in source fields`
- `[OOTB][Net] Detection unknown assets in device fields`

Если правила привязаны — переходите к шагу 2.2.

Если правила есть в системе, но не привязаны — привяжите их к коррелятору и обновите параметры коррелятора.

Если правил нет в системе — скачайте их из репозитория ресурсов. Нужный пакет: **`[OOTB] KUMA 4.x resources.`** После загрузки привяжите правила к коррелятору и обновите его параметры.

2.2. Импортировать правило реагирования

Импортируйте правило реагирования вручную из файла:

- **Ссылка**: https://box.kaspersky.com/seafhttp/f/c98c50e678794f248be5/?op=view
- **Пароль для импорта**: `Kaspersky@KUMA4.2!`

После импорта привяжите правило реагирования к коррелятору.

2.3. Обновить параметры коррелятора

Обновите параметры коррелятора. После этого связка «правила корреляции + правило реагирования + скрипт» начнёт работать.

Примечание: активный лист

Если правила `[OOTB][Net] Detection unknown assets ...` уже присутствовали в системе и работали ранее, срабатываний может не быть, потому что известные активы уже накоплены в активном листе.

Чтобы срабатывания снова появились, очистите активный лист на корреляторе:

- **Активный лист:** `[OOTB][Net] Detection unknown assets`

После очистки листа коррелятор снова начнёт фиксировать неизвестные активы и запускать скрипт импорта.

Revision #4
Created 2024-02-15 13:05:06 UTC by Koala
Updated 2026-08-04 11:37:48 UTC by Koala