

# Отправка уведомления в телеграм-бот

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

## Настройка Telegram

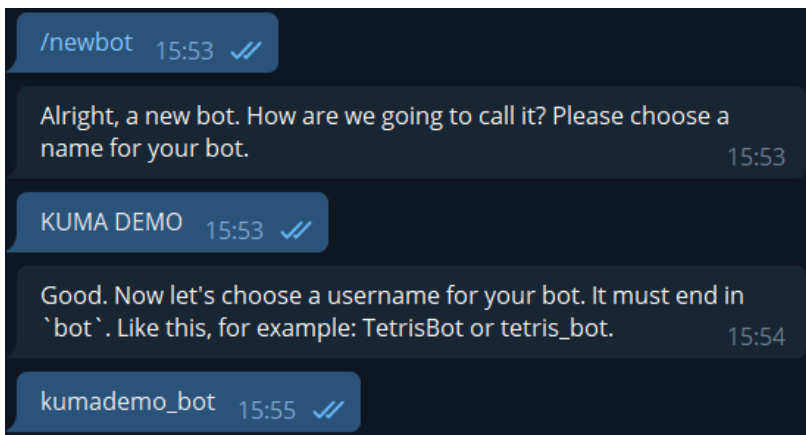
1. В Telegram находим бота: <https://t.me/BotFather>

2. Запускаем командой `/start`

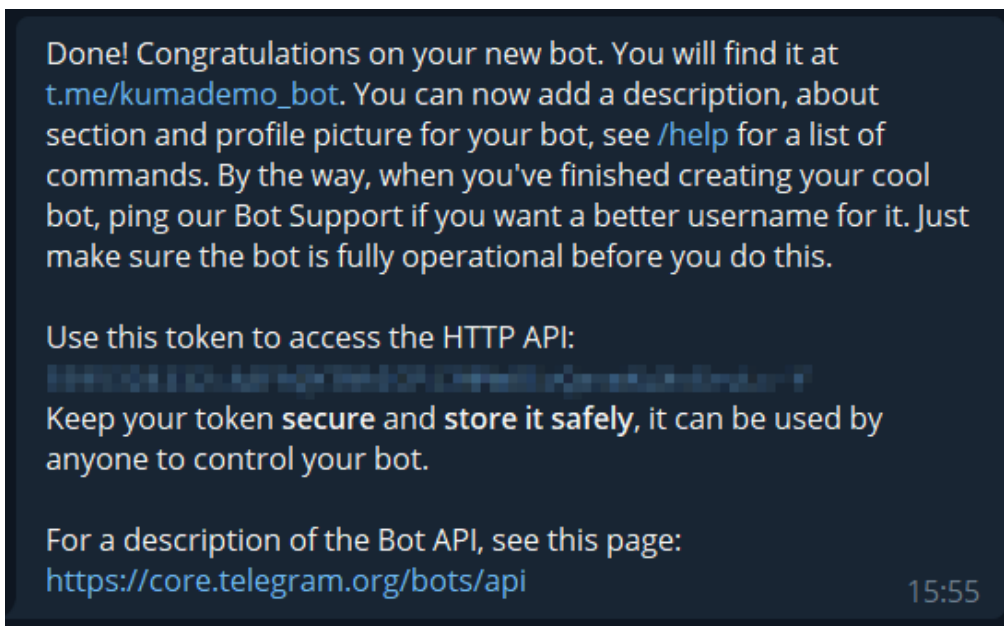


3. Создаем нового бота командой `/newbot`

4. Вводим желаемое имя и логин бота (должен заканчиваться словом `bot`). В данном примере имя бота "KUMA DEMO" и логин бота "kumademo\_bot".



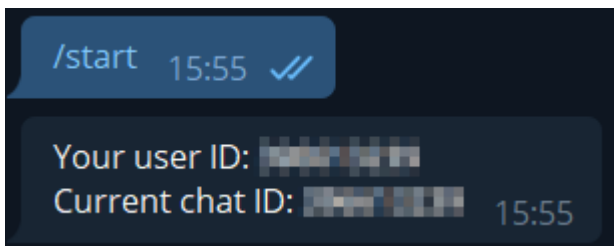
5. По завершении получаем токен для обращения к боту и ссылку на него



6. Если планируется использовать бота в группе, а не в личных сообщениях, то необходимо изменить настройки приватности. Для этого вводим `/mybots`, выбираем своего бота из списка, выбираем Bot Settings, Group Privacy и выбираем Turn off. После этого бот сможет отправлять сообщения в группах.

7. Далее переходим в своего бота по ссылке полученной от BotFather и выполняем команду `/start` для запуска бота.

8. Для отправки сообщения отдельному пользователю необходимо начать диалог с ботом, а также узнать `chat_id` этого пользователя. Чтобы узнать `chat_id` пользователя заходим в бота [https://t.me/getmyid\\_bot](\"https://t.me/getmyid_bot\") и вводим команду `/start`. Полученное значение `chat_id` потребуется в дальнейшем для отправки сообщений ботом

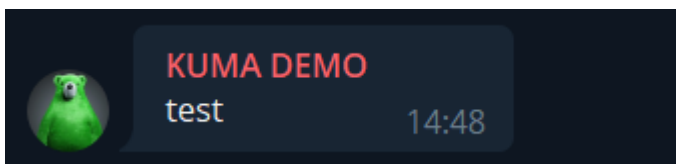


9. Чтобы отправлять сообщение ботом в группу необходимо узнать chat\_id этой группы. Для этого в созданную группу необходимо пригласить бота [https://t.me/getmyid\\_bot](https://t.me/getmyid_bot) и он выведет chat\_id группы (в поле Current chat ID), который потребуется в дальнейшем для отправки уведомлений. После получения id бота нужно удалить из группы.

10. Теперь можно отправить тестовое сообщение боту набрав в строке браузера команду:

```
https://api.telegram.org/bot<token>/sendMessage?chat_id=<chat_id>&text=test
```

Подставив вместо `<token>` и `<chat_id>` значения полученные ранее. В результате в личном чате или группе (в зависимости от выбранного способа) должно появиться уведомление, а в ответе браузера JSON не должен содержать ошибок.



11. После проверки работоспособности бота можно переходить к настройке отправки уведомлений.

## Скрипт уведомления

1. Создайте скрипт уведомления

В простейшем виде скрипт отправки уведомления выглядит следующим образом:

```
#!/bin/bash
set -eu
CHAT_ID=<chat_id из п.8-9 предыдущего раздела>
TG_TOKEN=<token из п.5 предыдущего раздела>
RULE=$1
TEXT="Произошла сработка правила <b>$RULE</b>"
curl --data-urlencode "chat_id=$CHAT_ID" --data-urlencode "text=$TEXT" --data-urlencode "parse_mode=HTML"
https://api.telegram.org/bot$TG_TOKEN/sendMessage
```

В случае, если у сервера коррелятора отсутствует прямой доступ в Интернет, скрипт можно модифицировать добавив адрес прокси-сервера для доступа в Интернет

```
#!/bin/bash
set -eu
CHAT_ID=<chat_id из п.8-9 предыдущего раздела>
TG_TOKEN=<token из п.5 предыдущего раздела>
RULE=$1
TEXT="Произошла сработка правила <b>$RULE</b>"
PROXY=<адрес и порт прокси-сервера>
curl --proxy $PROXY --data-urlencode "chat_id=$CHAT_ID" --data-urlencode "text=$TEXT" --data-urlencode
"parse_mode=HTML" https://api.telegram.org/bot$TG_TOKEN/sendMessage
```

### Еще один вариант скрипта

Еще один вариант скрипта, когда в качестве аргумента передаем следующие поля  
"{{.Timestamp}} | {{.Name}} | {{.DeviceHostName}}":

```
#!/bin/bash

set -eu

CHAT_ID=<chat_id из п.8-9 предыдущего раздела>
TG_TOKEN=<token из п.5 предыдущего раздела>

RULE=$1

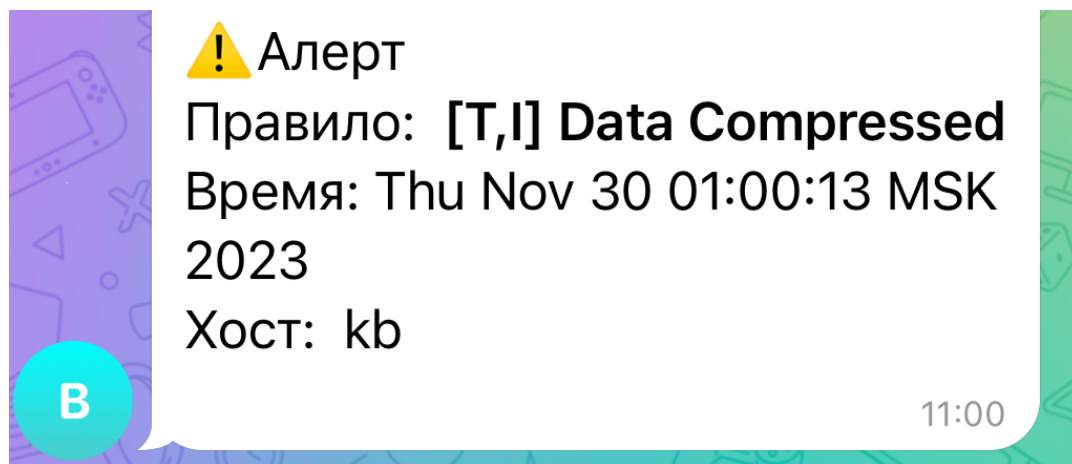
# writing local log of arguments
echo $(date +"%d-%m-%Y %T.%3N") - $RULE >> /opt/kaspersky/kuma/correlator/0b9200ae-d5a9-41ce-
bf7b-c16814ed9524/scripts/bot.log

# escaping spec characters in argument except \s \|
RULE=$(echo $RULE | sed 's/[[][\~`!@#\$%^&*()=+{};:'"'"<>/?-]/\&/g')

#try to beautify alert if arg is like "{{.Timestamp}} | {{.Name}} | {{.DeviceHostName}}"
{
    TIME=$(date -d @"${(echo $RULE | cut -d "|" -f 1)/1000}")
    NAME=$(echo $RULE | cut -d "|" -f 2)
    HOST=$(echo $RULE | cut -d "|" -f 3)
```

```
TEXT="⚠️Алерт %0АПравило: <b>$NAME</b> %0АВремя: $TIME %0АХост: $HOST"
} || {
#else if can't beautify
TEXT="⚠️Алерт %0АПравило: <b>$NAME</b> %0"
}
curl -XPOST
"https://api.telegram.org/bot$TG_TOKEN/sendMessage?chat_id=$CHAT_ID&text=$TEXT&parse_mode=html"
```

Получаем алерт вида:



2. Поместите скрипт в папку коррелятора, уведомления о сработках которого необходимо отправлять через телеграм-бот

Путь для размещения скрипта

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

<id> - идентификатор коррелятора, можно найти в веб-интерфейсе ([ссылка](#))

3. Назначьте пользователя kuma владельцем файла и дайте файлу права на выполнение

```
chown kuma:kuma /opt/kaspersky/kuma/correlator/<id>/scripts/bot.sh
chmod +x /opt/kaspersky/kuma/correlator/<id>/scripts/bot.sh
```

## Настройка KUMA

1. В веб-интерфейсе KUMA перейдите на вкладку **Resources**, выберите **Response** и нажмите на кнопку **Add Response**.

## 2. Задайте параметры правила реагирования

- В поле **Name** укажите имя правила реагирования.
- Укажите тенант.
- В поле **kind** выберите **script**.
- Задайте имя скрипта в поле **Script name**.
- В качестве аргумента укажите **{{.Name}}** - так в качестве аргумента выполнения скрипта будет передаваться имя корреляционного события.

3. Далее перейдите в настройки коррелятора, который будет выполнять реагирование. На вкладке **Response** нажмите **Add** и из выпадающего списка выберите созданное ранее правило реагирования.

**Response**

In response rules you can define what action should be taken if correlator finds a threat. For details see [Online Help](#).

\*Response rule: Bot alert

\*Kind: script

Timeout: 0  
Timeout in seconds

\*Script name: bot.sh

Script arguments: {{.Name}}

Workers: 0

Filter: Create new

☐ Save filter

Conditions: AND + Add condition + Add group + Add filter

## 4. Обновите параметры сервиса коррелятора

## 5. Результат работы скрипта представлен ниже

