

Блокировка адресов при помощи Cisco ASA Firewall на основе сработок алертов

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

Данный скрипт добавлен пользователем KUMA Community и предоставляется AS IS без каких-либо гарантий и ответственности.

В нашем случае мы при помощи правила корреляции, на основе логов веб-сервера Apache, блокируем входящий трафик от внешних адресов.

Для настройки реагирования средствами Cisco ASA (блокировки IP адресов) (пример, при подключении по SSH) необходимо:

1. Авторизоваться на ASA с привилегиями, позволяющими переходить в режим конфигурирования, отправить команду `conf t`
2. Создать объект, в который наш скрипт будет добавлять адреса (черный список), назовём его BLACKLIST. Создаём командой `object network BLACKLIST`
3. На ASA создаём access-list, который будет блокировать входящие сетевые соединения из интернета от IP находящихся в нашем объекте. Пример:

```
access-list INTERNET extended deny ip object-group BLACKLIST any
```

Настройка на стороне KUMA

1. В группирующем поле **правила корреляции** должны находиться целевые поля, которые используются в правилах реагирования, в нашем примере это **sourceAddress**

Общие	Селекторы	Действия
*Название	<input type="text" value="Test_IP"/>	
*Тенант	<input type="text" value="MAIN"/>	
*Тип	<input type="text" value="standard"/>	
*Группирующие поля	<input type="button" value="+ Добавить поле"/> SourceAddress <input type="button" value="x"/> <input type="button" value="x Сбросить"/>	
Уникальные поля	<input type="button" value="+ Добавить поле"/>	
Частота срабатываний	<input type="text"/> <input type="button" value="?"/>	
*Время жизни контейнера, сек.	<input type="text"/>	
Политика хранения базовых событий	<input type="text"/>	
Уровень важности	<input type="text" value="Низкий"/>	
Сортировать по	<input type="text"/>	
Описание	<input type="text"/>	

- В "селекторы" и "действия" задаём необходимые в данном кейсе параметры. Обязательно добавить обогащение событие EventOutcome (как указано на скриншоте), это ключ (триггер) для следующего этапа по запуску правила реагирования.

Общие Селекторы Действия

Действия

- > На первом срабатывании правила
- > На последующих срабатываниях правила
- ▼ На каждом срабатывании правила
- Отправить событие на дальнейшую обработку
- Отправить событие снова в коррелятор
- Не создавать алерт

Обогащение

⋮ Обогащение №1

- *Тип источника данных
- *Целевое поле
- Константа ?
- *Отладка

3. Размещаем скрипт (находится в [папке](#) пресейл-пака) предварительно заменив ключевые данные в скрипте, а именно: ip asa, login, password с правами, необходимыми для добавления в объект BLACK (см. выше, этап настройки ASA)
4. Скрипт помещаем на сервере(-ах) по пути `/opt/kaspersky/kuma/correlator/<id>/scripts/` и предоставляем права пользователю kuma, чтобы служба имела достаточные права для запуска скрипта, командами:
`chown kuma:kuma /opt/kaspersky/kuma/correlator/<id>/scripts/asa.py`
`chmod +x /opt/kaspersky/kuma/correlator/<id>/scripts/asa.py`

5. **Данный этап индивидуален и зависит конкретно от Вашего экземпляра ОС, возможно будут дополнительные ошибки при запуске скрипта, для проверки скрипт можно запускать вручную с сервера и проверять**

работоспособность

Также на сервер коррелятора в pip3 необходимо до установить следующие библиотеки, для возможности запуска python3.*

threading
paramiko
sys
argparse
subprocess

команда:

```
pip3 install threading paramiko sys argparse subprocess
```

6. Создаём правило реагирования, которое будет непосредственно запускать скрипт на коллекторе (шаг 4)

[Правила реагирования](#) >

Редактирование правила реагирования

The screenshot shows a configuration form for a rule named 'ASA Block'. The form includes the following fields and options:

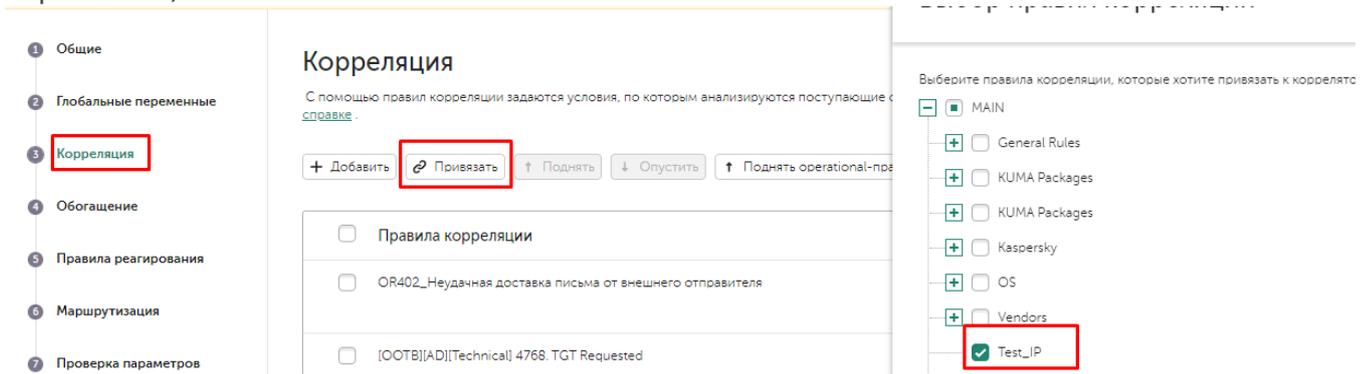
- *Название:** ASA Block
- *Тенант:** MAIN
- *Тип:** Запуск скрипта
- Время ожидания:** 0 (Время ожидания в секундах)
- *Название скрипта:** asa.py
- Аргументы скрипта:** --ip {{.SourceAddress}}
- Рабочие процессы:** 0
- Описание:** Описание
- Фильтр:** Создать
- Сохранить фильтр
- Условия:** И, + Добавить условие, + Добавить группу, + Добавить фильтр

At the bottom, a red box highlights a condition rule: `Если поле события EventOutcome = константа BLOCK`.

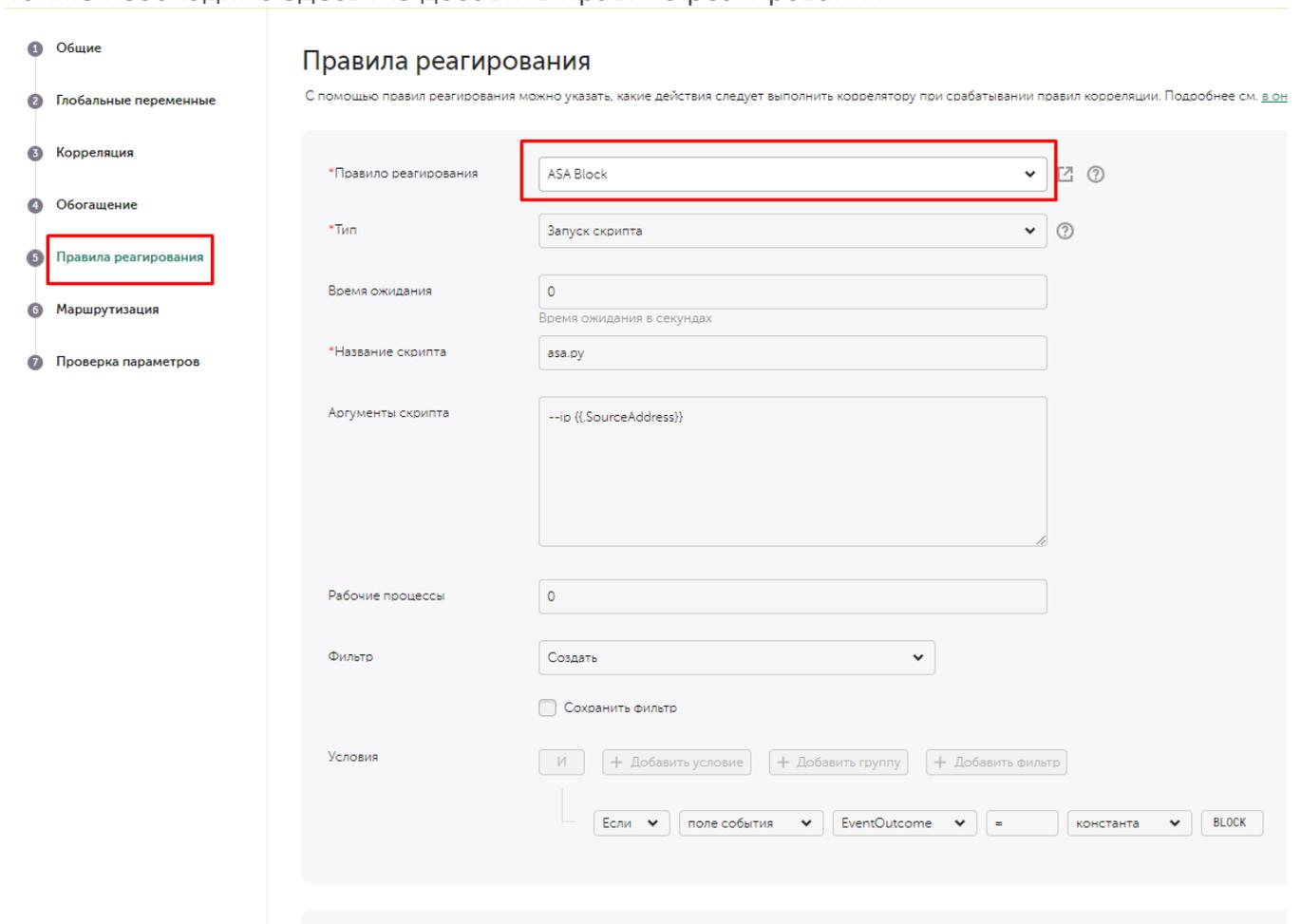
ключевое, задаём Название скрипта, который мы перенесли на сервер коррелятора (шаг 4), также аргументы скрипта, как на скриншоте и ключевое поле EventOutcome = BLOCK (добавляется при срабатывании алерта, при помощи обогащения) (указано

как пример, можно задать списком и другими полями).

7. Осталось привязать новое правило корреляции. Привязываем к коррелятору. Ресурсы -> Корреляторы -> Выбираем наш -> Корреляция, привязать (на скриншоте)



8. Также необходимо здесь же добавить правило реагирования



9. Готово, сохраняем и рестартуем коррелятор (ы)

Revision #9

Created 8 December 2023 17:34:20 by Anton

Updated 19 July 2024 14:37:44 by Koala