**Инструкция по автоматизированной миграции политики безопасности с доменов MDS R77.30 на домены MDS R80.XX**

Оглавление

[1. Необходимое оборудование и ПО 3](#_Toc10117494)

[2. Подготовительный этап 4](#_Toc10117495)

[3. Наполнение базы данных 10](#_Toc10117496)

[4. Фильтрация политики безопасности и генерация новой политики безопасности для VS 11](#_Toc10117497)

[5. Фильтрация политики безопасности и генерация новой политики безопасности OUT фаерволла 13](#_Toc10117498)

[6. Экспорт сетевых объектов и сервисов на новый MDS 14](#_Toc10117499)

[7. Экспорт правил политики безопасности на новый MDS 15](#_Toc10117500)

[8. Логирование и траблшутинг 16](#_Toc10117501)

## 1. Необходимое оборудование и ПО

1. На АРМ с которого будет производиться запуск скриптов должно быть установлено следующее ПО:

- Последняя версия интерпретатора Python и комплект стандартных библиотек;

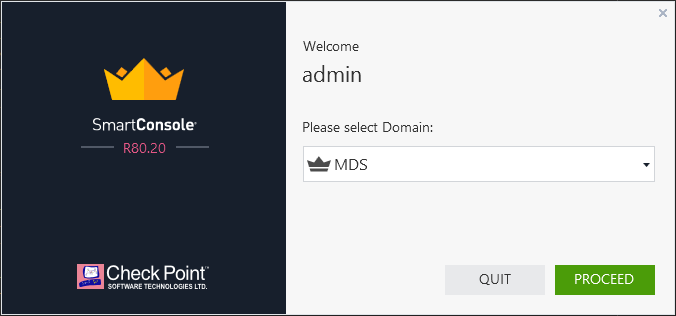
- дополнительно установленная библиотека xlsxwriter

- SmartConsole исходной и целевой версии SMS.

2. АРМ должна иметь сетевой доступ к исходному и целевому SMS.

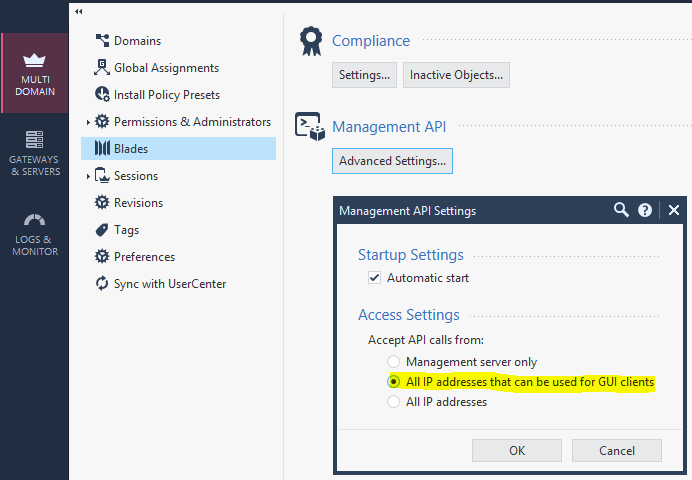
## 2. Подготовительный этап

1. Подключиться к MDS на уровень самого MDS:

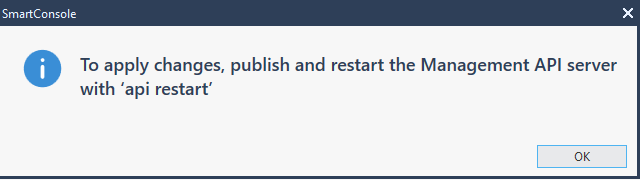


2. Перейти на вкладку Blades -> Management API -> Advanced Settings…

Установить значение All IP address that can be used for GUI clients:



3. Опубликовать изменения и перезапустить API сервер:



# api restart

2019-May-21 14:21:00 - Stopping API...

2019-May-21 14:21:02 - API stopped successfully.

2019-May-21 14:21:02 - Starting API...

. . . . . . . . . . . . .

2019-May-21 14:22:15 - API started successfully.

4. Распаковать архив со скриптом CheckPointParser.zip

5. В директории \data\ создать папку соответствующую имени политики подлежащей переносу. Например \data\Rosneft-Moscow-Tver

6. Экспорт политики и базы объектов для переносимого policy package в формате XML. SK по теме:

<https://supportcenter.checkpoint.com/supportcenter/portal?eventSubmit_doGoviewsolutiondetails=&solutionid=sk64501>

Примеры:

1) Пример запроса – экспорт политики БЕЗ глобальных правил и объектов:

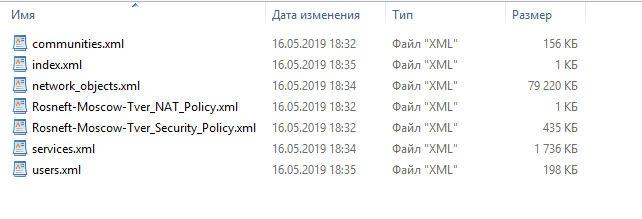
cpdb2web.exe -s 10.16.63.2 -u \*\*\*\*\* –p \*\*\*\*\* -o C:\Users\user\PycharmProjects\CheckPointParser\data\Rosneft-Moscow-Tver -l Rosneft-Moscow-Tver –gr

2) Пример запроса – с экспортом глобальных правил и объектов:

cpdb2web.exe -s 10.16.63.2 -u \*\*\*\*\* –p \*\*\*\*\* -o C:\Users\user\PycharmProjects\CheckPointParser\data\Rosneft-Moscow-Tver -l Rosneft-Moscow-Tver

Результатом работы станет несколько файлов в формате .xml, содержащие сетевые объекта и правила.

Пример:



6. Указать параметры работы скрипта в файле конфигурации \config.py:

**api\_server** = r"" - MDS IP.

**username** = r"" - Имя администратора с правами на запись и публикацию.

**password** = r"" - Пароль администратора.

**domain** = r"" - Имя домена.

**package** = r"" - Имя пакета политики (должен был быть создан предварительно! Например Rosneft-Moscow-Tver)

**layer** = r"" - Имя layer-а, куда будут добавляться правила. Имя layer-а можно узнать из CLI командой. Например:

> show package name Rosneft-Moscow-Tver --format json --root true

{

"uid" : "1a0d9a8a-8337-4e92-a88f-9f97cf860f67",

"name" : "Rosneft-Moscow-Tver",

"type" : "package",

"domain" : {

"uid" : "27961d7b-67ad-4c0d-9cfd-eb2ff4e9b261",

"name" : "Test\_DOMAIN",

"domain-type" : "domain"

},

"access" : true,

"access-layers" : [ {

"uid" : "791fd880-d7f5-47ff-bbab-69b89d1f7c46",

"name" : "Rosneft-Moscow-Tver Network",

"type" : "access-layer",

"domain" : {

"uid" : "27961d7b-67ad-4c0d-9cfd-eb2ff4e9b261",

"name" : "Test\_DOMAIN",

"domain-type" : "domain"

}

} ],

"threat-layers" : [ {

"uid" : "667ef598-782b-4eae-bbf9-fdf71bba9215",

"name" : "IPS",

"type" : "threat-layer",

"domain" : {

"uid" : "27961d7b-67ad-4c0d-9cfd-eb2ff4e9b261",

"name" : "Test\_DOMAIN",

"domain-type" : "domain"

}

}, {

"uid" : "de2a533e-ab91-4645-905d-b5f9b4fdeb95",

"name" : "Rosneft-Moscow-Tver Threat Prevention",

"type" : "threat-layer",

"domain" : {

"uid" : "27961d7b-67ad-4c0d-9cfd-eb2ff4e9b261",

"name" : "Test\_DOMAIN",

"domain-type" : "domain"

}

} ],

"vpn-traditional-mode" : false,

"nat-policy" : true,

"qos" : false,

"qos-policy-type" : "recommended",

"desktop-security" : false,

"threat-prevention" : true,

"installation-targets" : "all",

"comments" : "",

"color" : "black",

"icon" : "Blades/Access",

"tags" : [ ],

"meta-info" : {

"lock" : "unlocked",

"validation-state" : "ok",

"last-modify-time" : {

"posix" : 1558450817974,

"iso-8601" : "2019-05-21T18:00+0300"

},

"last-modifier" : "admin",

"creation-time" : {

"posix" : 1558450817974,

"iso-8601" : "2019-05-21T18:00+0300"

},

"creator" : "admin"

},

"read-only" : false

}

**global\_package** = r"" - Имя пакета глобальной политики (должен был быть создан предварительно! Например TSAUK)

**global\_layer** = r"" - Имя layer-а, куда будут добавляться правила. Имя layer-а можно узнать из CLI командой. Например:

> show package name TSAUK --format json --root true

{

"uid" : "342b0bb0-8ef7-4a62-8d66-68ca69b2e9d3",

"name" : "TSAUK",

"type" : "package",

"domain" : {

"uid" : "1e294ce0-367a-11e3-aa6e-0800200c9a66",

"name" : "Global",

"domain-type" : "global domain"

},

"access" : true,

"access-layers" : [ {

"uid" : "6d13cdd6-a76d-4ea5-91a7-113cbe2758ee",

"name" : "TSAUK Network",

"type" : "access-layer",

"domain" : {

"uid" : "1e294ce0-367a-11e3-aa6e-0800200c9a66",

"name" : "Global",

"domain-type" : "global domain"

}

} ],

"threat-layers" : [ {

"uid" : "05233769-b135-459a-83d3-3eceb21c12c1",

"name" : "IPS",

"type" : "threat-layer",

"domain" : {

"uid" : "1e294ce0-367a-11e3-aa6e-0800200c9a66",

"name" : "Global",

"domain-type" : "global domain"

}

}, {

"uid" : "1477e6f5-b181-4592-84da-317e68a90486",

"name" : "TSAUK Threat Prevention",

"type" : "threat-layer",

"domain" : {

"uid" : "1e294ce0-367a-11e3-aa6e-0800200c9a66",

"name" : "Global",

"domain-type" : "global domain"

}

} ],

"vpn-traditional-mode" : false,

"nat-policy" : true,

"qos" : false,

"qos-policy-type" : "recommended",

"desktop-security" : false,

"threat-prevention" : true,

"installation-targets" : "all",

"comments" : "",

"color" : "black",

"icon" : "Blades/Access",

"tags" : [ ],

"meta-info" : {

"lock" : "unlocked",

"validation-state" : "ok",

"last-modify-time" : {

"posix" : 1570018356599,

"iso-8601" : "2019-10-02T15:12+0300"

},

"last-modifier" : "admin",

"creation-time" : {

"posix" : 1570018356599,

"iso-8601" : "2019-10-02T15:12+0300"

},

"creator" : "admin"

},

"read-only" : false

}

**xml\_dir** = r""- директория с на ранее экспортированными XML файлы

**xml\_policy\_file** = r"" - полный путь к имени XML файла с политикой. Например: D:\customers\AKON\CheckPointParser\data\Rosneft-Moscow-Tver\Rosneft-Moscow-Tver\_Security\_Policy.xml

**filter\_network**=r"" – фильтр модификации политики безопасности для экспорт части политики за VS. Например, если стоит задача создать политику для VS за которую переедут подсети BGP AS: 91.142.135.0/255.255.255.0 и 193.30.236.0/255.255.252.0. То содержимое строки будет перечислением этих подсетей с новой строки, без пустых строк. Допускаются форматы:

91.142.135.0/255.255.255.0,193.30.236.0/255.255.252.0

91.142.135.0/24,193.30.236.0/22

Пример:

filter\_network = r'''91.142.135.0/255.255.255.0

193.30.236.0/255.255.252.0'''

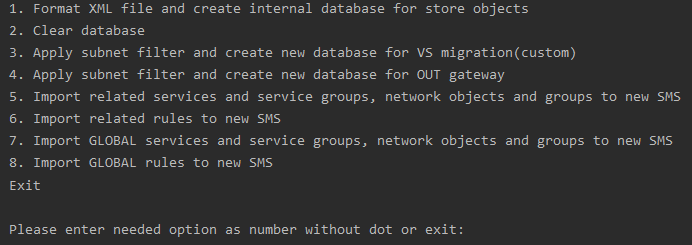
**session** = "" – служебное поле, не изменять!

## 3. Наполнение базы данных

1. Запустить CMD от имени администратора и перейти в папку \parser\, где находится файл скрипта parser\_script.py

2. Запустить скрипт командой: python parser\_script.py

После этого отобразится меню скрипта:

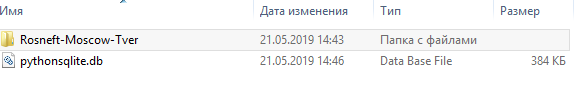


3. Выбрать пункт 1, нажать Enter.

Результатом работы этого пункта скрипта станут:

- На выходе в директории XML файлов будут новые файлы с суффиксом \_new (если файлы с таким суффиксом есть, то они будут перезаписаны):

- Скрипт создаст SQL базу данных в директории /data/ и проиндексирует её:



База данных будет содержать:

- таблицы сервисов, групп сервисов;

- таблицы сетевых объектов, групп сетевых объектов;

- таблицы правил политики.

Несколько важных моментов:

- при повторном использовании скрипта для экспорта другой политики необходимо предварительно удалить базу данных в /data/pythonsqlite.db, выбрав пункт 2 в меню программы «Clear database»

- после завершения работы в директории/logs/ генерируется табличный файл groups\_r77.30.xlsx. Таблица содержит список исходных групповых объектов и количество элементов в каждой из них.

## 4. Фильтрация политики безопасности и генерация новой политики безопасности для VS

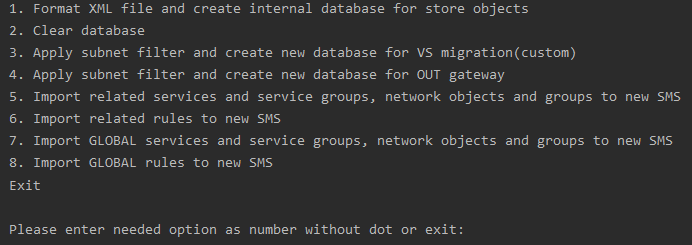
1. Для того чтобы сгенерировать новую политику безопасности для вновь создаваемой VS на основании существующей политики безопасности (скорее всего это будет политика Moscow\_Central\_FW). Необходимо:

2. Определить пул подсетей, который будут находиться за вновь создаваемой VS и указать их в файле конфигурации в параметре filter\_network, как описано выше.

3. Если база данных еще пустая/отсутствует, то наполнить базу данных по методике, как описано в разделе выше «3. Наполнение базы данных». Источником для базы данных должна быть политика, на основании которой будет формироваться новая политика для VS (скорее всего это будет политика Moscow\_Central\_FW)

4. Запустить скрипт командой: python parser\_script.py

После этого отобразится меню скрипта:



5. Выбрать пункт 3, нажать Enter.

6. Результатом работы программы станет новая политика безопасности для VS-ки. При этом существующая (исходная) политика в базе данных будет заменена новой!

7. Логика выбора правил для политики VS при этом выглядит следующим образом:

Допустим стоит задача в том, чтобы сгенерировать политику для VS за которой будут находится подсети:

91.142.135.0/255.255.255.0

193.30.236.0/255.255.252.0

И есть несколько видов правил в исходной политике:

1. Правило с source (полностью или частично) в заданном адресном диапазоне и с destination вне заданного диапазона. Например:

|  |  |
| --- | --- |
| **Source** | **Destination** |
| 91.142.135.45  193.30.236.45  192.168.0.1  10.10.1.1 | 172.16.0.0/12 |

То правило будет перенесено в виде:

|  |  |
| --- | --- |
| **Source** | **Destination** |
| 91.142.135.45  193.30.236.45 | 172.16.0.0/12 |

2. Правило с destination (полностью или частично) в заданном адресном диапазоне и с source вне заданного диапазона. Например:

|  |  |
| --- | --- |
| **Source** | **Destination** |
| 172.16.0.0/12 | 91.142.135.45  193.30.236.45  192.168.0.1  10.10.1.1 |

То правило будет перенесено в виде:

|  |  |
| --- | --- |
| **Source** | **Destination** |
| 172.16.0.0/12 | 91.142.135.45  193.30.236.45 |

3. Правило с source (полностью или частично) в заданном адресном диапазоне и с (полностью или частично) в заданном адресном диапазоне. Например:

|  |  |
| --- | --- |
| **Source** | **Destination** |
| 91.142.135.45  193.30.236.45  192.168.0.1  10.10.1.1 | 91.142.135.45  193.30.236.45  192.168.0.1  10.10.1.1 |

То правило будет перенесено в виде:

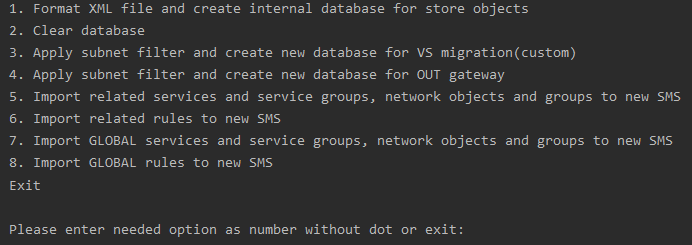
|  |  |
| --- | --- |
| **Source** | **Destination** |
| 91.142.135.45  193.30.236.45  192.168.0.1  10.10.1.1 | 91.142.135.45  193.30.236.45  192.168.0.1  10.10.1.1 |

## 5. Фильтрация политики безопасности и генерация новой политики безопасности OUT фаерволла

1. Генерация политики для OUT фаерволла представляет собой частный случай предыдущего пункта, с той разницей что в выборку попадают только те правила, source которых входит в приватный диапазон RFC1918 (192.168.0.0/255.255.0.0,10.0.0.0/255.0.0.0,172.16.0.0/255.240.0.0), а destination НЕ входит в этот диапазон.

2. Запустить скрипт командой: python parser\_script.py

После этого отобразится меню скрипта:

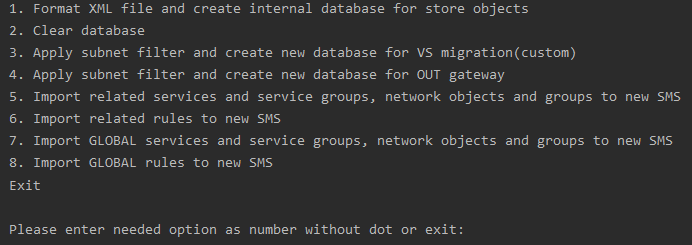


3. Выбрать пункт 4, нажать Enter.

6. Результатом работы программы станет новая политика безопасности для OUT фаерволла. При этом существующая (исходная) политика в базе данных будет заменена новой!

## 6. Экспорт сетевых объектов и сервисов на новый MDS

1. Запустить скрипт командой: python parser\_script.py , выбрать пункт 5, нажать Enter.



2. Если вы подключаетесь к серверу первый раз, то он попросит принять фингерпринт. Необходимо согласиться.

После этого начнется процесс экспорта сетевых объектов и сервисов.

Несколько важных моментов:

- в версии R80.X изменились цвета объектов по сравнению с R77.30. Поэтому те цвета, которые совпадают, будут сохранены. Прочие цвета станут черными.

- глобальные объекты конвертируются в локальные. Из именований удаляются префиксы g\_ и g.

-экспортируются только сервисы, используемые в правилах или вложенные в группы, используемые в правилах. То есть реально использующиеся объекты.

- экспортируются только объекты host\_plain, address\_range, network, network\_group. Прочие объекты – гейты, ежи, коннектры, серверы Check Point НЕ ЭКСПОРТИРУЮТСЯ и создаются как хостовые объекты зеленого цвета с суффиксом F\_ и IP адресом 1.1.1.1

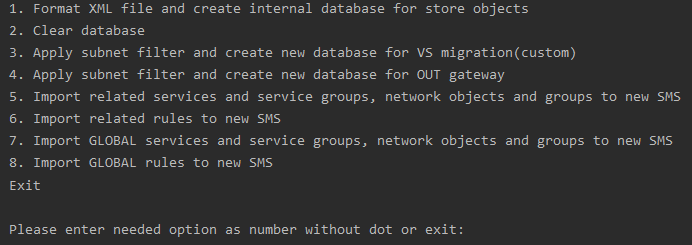
- экспортируются только сетевые объекты, используемые в правилах или вложенные в группы, используемые в правилах. То есть реально использующиеся объекты.

- список сервисов и сетевых объектов, которые присутствуют в правилах, но не были экспортированы на новый MDS по какой-либо причине выгружается в директорию /logs/ в виде XLSX-таблиц с именами service\_error.xlsx и network\_objects\_error.xlsx.

- после завершения работы в директории/logs/ генерируется табличный файл groups\_r80\_local.xlsx. Таблица содержит список экспортированных групповых объектов и количество элементов в каждой из них.

## 7. Экспорт правил политики безопасности на новый MDS

1. Запустить скрипт командой: python parser\_script.py , выбрать пункт 6, нажать Enter.



2. Если вы подключаетесь к серверу первый раз, то он попросит принять фингерпринт. Необходимо согласиться.

После этого начнется процесс экспорта правил политики безопасности.

Несколько важных моментов:

- не подлежат переносу отключенные правила и правила с временными объектами

- все правила создаются с уровнем логирования «Log»

- правила добавляются ниже текущего последнего

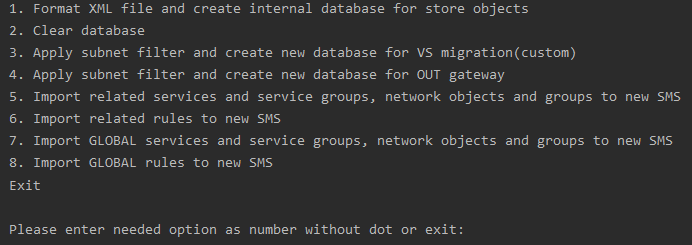
- учитывается отрицание на объектах

- имена правил и комментарии переносятся как они есть

- секции переносятся в той последовательности, в которой они есть.

## 8. Экспорт глобальных сетевых объектов и сервисов на новый MDS

1. Запустить скрипт командой: python parser\_script.py , выбрать пункт 7, нажать Enter.



2. Если вы подключаетесь к серверу первый раз, то он попросит принять фингерпринт. Необходимо согласиться.

После этого начнется процесс экспорта глобальных сетевых объектов и сервисов.

Несколько важных моментов:

- в версии R80.X изменились цвета объектов по сравнению с R77.30. Поэтому те цвета, которые совпадают, будут сохранены. Прочие цвета станут черными.

- для глобальных объектов сохраняется префиксы g\_ и g.

-экспортируются только сервисы, используемые в правилах или вложенные в группы, используемые в правилах. То есть реально использующиеся объекты.

- экспортируются только объекты host\_plain, address\_range, network, network\_group. Прочие объекты – гейты, ежи, коннектры, серверы Check Point НЕ ЭКСПОРТИРУЮТСЯ и создаются как хостовые объекты зеленого цвета с суффиксом F\_ и IP адресом 1.1.1.1

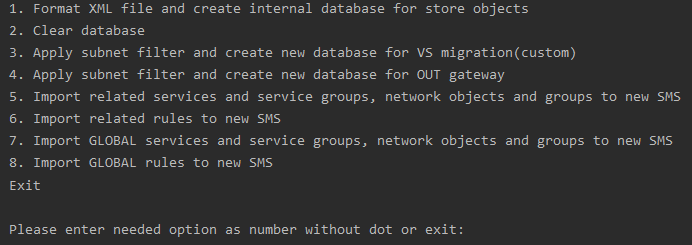
- экспортируются только сетевые объекты, используемые в правилах или вложенные в группы, используемые в правилах. То есть реально использующиеся объекты.

- список сервисов и сетевых объектов, которые присутствуют в правилах, но не были экспортированы на новый MDS по какой-либо причине выгружается в директорию /logs/ в виде XLSX-таблиц с именами service\_error.xlsx и network\_objects\_error.xlsx.

- после завершения работы в директории/logs/ генерируется табличный файл roups\_r80\_global.xlsx. Таблица содержит список экспортированных глобальных групповых объектов и количество элементов в каждой из них.

## 9. Экспорт глобальных правил политики безопасности на новый MDS

1. Запустить скрипт командой: python parser\_script.py , выбрать пункт 8, нажать Enter.



2. Если вы подключаетесь к серверу первый раз, то он попросит принять фингерпринт. Необходимо согласиться.

После этого начнется процесс экспорта правил политики безопасности.

Несколько важных моментов:

- не подлежат переносу отключенные правила и правила с временными объектами

- все правила создаются с уровнем логирования «Log»

- правила добавляются ниже текущего последнего

- учитывается отрицание на объектах

- имена правил и комментарии переносятся как они есть

- секции переносятся в той последовательности, в которой они есть.

## 10. Логирование и траблшутинг

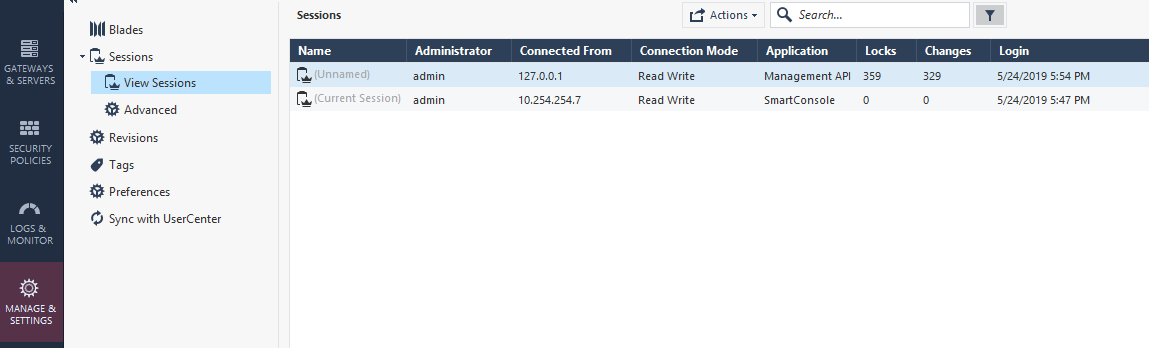
1. Логи работы скрипта пишутся в директорию /logs/parser\_log.log

Каждый раз при запуске скрипта лог перезаписывается.

2. Список сервисов и сетевых объектов, которые присутствуют в правилах, но не были экспортированы на новых MDS по какой-либо причине выгружается в директорию /logs/ в виде XLSX-таблиц с именами service\_error.xlsx и network\_objects\_error.xlsx.

3. При добавление объектов или правил публикация осуществляется один раз – после того как были последовательно добавлены все объекты этого типа. Процесс публикации может занимать несколько минут, если было добавлено много объектов.

Процесс добавления новых объектов и правил можно отслеживать на вкладке “MANAGE & SETTINGS”-> Sessions -> View Session



В этом же меню можно сбросить изменения сессии, если скрипт аварийно завершиться до публикации.

4. Есть не удается подключиться по API к серверу управления, то необходимо проверить статус и логи:

- api status –s (с дебагом)

- /opt/CPsuite-R80/fw1/log/api.elg