Инструкция по эксплуатации Servicepipe FlowCollector

Листов: 19

2026 г

Оглавление

1 Введение

1.1. Общие положения

Настоящая инструкция по эксплуатации определяет порядок использования программного обеспечения FlowCollector, требования к пользователям системы, а также основные термины и обозначения, применяемые в документе.

Инструкция предназначена для администраторов и специалистов, осуществляющих сопровождение и эксплуатацию системы анализа потокового трафика.

1.2. Назначение программного обеспечения

Программное обеспечение FlowCollector предназначено для сбора, агрегирования и анализа потокового сетевого трафика с целью выявления DDoS-атак и иных сетевых аномалий.

При выявлении аномалии программное обеспечение может инициировать меры реагирования, включая формирование BGP-анонсов для перенаправления трафика в защитный контур.

1.3. Требования к квалификации персонала

Эксплуатация экземпляра программного обеспечения требует:

- знания и навыки администрирования ОС Linux и СУБД;
- знания и навыки работы с командной строкой;
- понимание работы сетевых протоколов;
- базовые знания языка SQL.

1.4. Доступ к экземпляру программного обеспечения

Экземпляр программного обеспечения FlowCollector, развернутый для проведения экспертизы, размещен на удаленном сервере. Доступ к экземпляру осуществляется по протоколу SSH. Команда для подключения:

```
ssh -p 22217 admin@185.169.152.74
```

Пароль: Jmn%lk1pz@mrl4(Mn1JA@

2 Обеспечение функционирования программного обеспечения

2.1 Общие сведения

Программное обеспечение FlowCollector функционирует в виде системной службы analyzer, управляемой средствами системы инициализации операционной системы Ubuntu.

Запуск, остановка и контроль состояния программы осуществляются посредством управления данной службой.

2.2 Запуск программы

Для запуска службы выполнить команду:

```
sudo systemctl start analyzer
```

После выполнения команды рекомендуется проверить состояние службы:

```
sudo systemctl status analyzer
```

Служба считается запущенной, если её состояние определяется как active (running).

2.3 Остановка программы

Для остановки службы выполнить команду:

```
sudo systemctl stop analyzer
```

Проверка состояния службы осуществляется командой:

```
sudo systemctl status analyzer
```

Служба считается остановленной, если её состояние определяется как inactive.

2.4 Перезапуск программы

Для перезапуска службы выполнить команду:

```
sudo systemctl restart analyzer
```

3 Управление объектами мониторинга

3.1 Общие положения

В системе FlowCollector объект мониторинга определяется конфигурационным файлом формата YAML.

Объект представляет собой набор:

- IP-префиксов получателей трафика;
- правил подсчёта и анализа трафика;
- пороговых значений;
- векторов анализа атак.

Каждый объект настраивается индивидуально с использованием отдельного конфигурационного файла.

3.2 Размещение конфигурационных файлов

Конфигурационные файлы объектов размещаются в каталоге:

```
/opt/spfc/etc/mo/
```

Допускается использование вложенной структуры директорий для логической группировки конфигураций.

```
/opt/spfc/etc/mo/web/

/opt/spfc/etc/mo/filial1/service1.yaml

/opt/spfc/etc/mo/filial2/service2.yaml
```

3.3 Создание конфигурационного файла

Для создания нового объекта необходимо создать YAML-файл в каталоге:

```
/opt/spfc/etc/mo/.
```

Пример:

```
sudo nano /opt/spfc/etc/mo/new-object.yaml
```

Имя файла выбирается произвольно и должно соответствовать имени создаваемого объекта.

3.4 Структура конфигурационного файла

Конфигурационный файл объекта содержит обязательные параметры верхнего уровня и блок правил обработки трафика.

3.4.1 Параметр name

Параметр name определяет строковое имя объекта мониторинга. Имя чувствительно к регистру и должно быть уникальным в рамках системы. Пример:

```
name: new-object
```

3.4.2 Параметр cidrs

Параметр cidrs определяет перечень IP-префиксов, относящихся к объекту мониторинга. Маска 0.0.0.0/0 не допускается. Пример:

```
cidrs:
  - 1.1.1.0/24
  - 1.1.2.0/24
```

3.4.3 Структура правила

Блок rules содержит набор правил мониторинга и анализа трафика. Каждое правило задаётся отдельным элементом списка и определяет параметры агрегации, анализа и пороговой детекции.

3.4.3.1 Параметр type

Параметр type определяет область агрегации трафика при выполнении анализа.

Допустимые значения:

- global — агрегированный подсчёт по всему объекту;
- local — подсчёт по каждому IP-адресу (/32) внутри заданных префиксов.

Допускается одновременное использование нескольких значений.

3.4.3.2 Параметр units

Параметр units определяет единицы измерения анализируемого трафика.

Допустимые значения:

- bytes — объём переданных данных;
- packets— количество пакетов.

Допускается одновременное использование нескольких единиц измерения.

3.4.3.3 Параметр vectors

Параметр vectors определяет тип анализируемого трафика либо характер предполагаемой атаки

Допустимые значения:

- total-traffic — любой трафик IPv4 и IPv6.
- dns-flood — TCP или UDP-трафик к порту назначения 53.
- ip-fragment-flood — IP-пакеты с установленным признаком фрагментации.
- icmp-flood — трафик протокола ICMP (IPv4).
- tcp-flood — любой TCP-трафик.
- tcp-cwr-flood — TCP-пакеты с установленным флагом CWR.
- tcp-ece-flood — TCP-пакеты с установленным флагом ECE.
- tcp-urg-flood — TCP-пакеты с установленным флагом URG.
- tcp-ack-flood — TCP-пакеты с установленным флагом ACK.
- tcp-psh-flood — TCP-пакеты с установленным флагом PSH.
- tcp-rst-flood — TCP-пакеты с установленным флагом RST.
- tcp-syn-flood — TCP-пакеты с установленным флагом SYN.
- tcp-fin-flood — TCP-пакеты с установленным флагом FIN.
- udp-flood — любой UDP-трафик.
- invalid-protocol-flood — IP-пакеты с недопустимым значением поля «Protocol» (0).
- chargen-amp — UDP-трафик с портом источника 19.
- dns-amp — UDP-трафик с портом источника 53.
- ntp-amp — UDP-трафик с портом источника 123.

- snmp-amp — UDP-трафик с портом источника 161.
- snmptrap-amp — UDP-трафик с портом источника 162.
- ldap-amp — UDP-трафик с портом источника 389.
- mssql-amp — UDP-трафик с портом источника 1434.
- ibm-cics-amp — UDP-трафик с портом источника 1435.
- ssdp-amp — UDP-трафик с портом источника 1900.
- apple-remote-desktop-amp — UDP-трафик с портом источника 3283.
- ws-discovery-amp — UDP-трафик с портом источника 3702.
- memcached-amp — UDP-трафик с портом источника 11211.
- udp-zero-payload-flood — UDP-трафик без полезной нагрузки.
- udp-big-packets-flood — UDP-трафик с увеличенным размером пакетов.

Допускается указание нескольких векторов в рамках одного правила.

3.4.3.4 Параметры пороговых значений

Пороговые параметры определяют условия фиксации аномалии. Поддерживаются следующие типы порогов:

- limit-threshold — фиксированное значение в выбранных единицах измерения;
- limit-diff — порог превышения значения за заданный интервал;
- limit-reldiff — относительное изменение (множитель роста).

Допускается одновременное использование нескольких типов порогов.

Превышение любого заданного порогового значения приводит к регистрации аномалии.

3.5 Пример конфигурации объекта

Ниже приведён пример конфигурационного файла объекта мониторинга web-services, предназначенного для анализа трафика, адресованного к префиксу 1.1.1.0/24.

```
name: web-services

cidrs:
  - 1.1.1.0/24

rules:

  - type:
      - global
    units:
      - bytes
    limit-threshold: 2000000000

  - type:
      - local
    units:
      - bytes
    limit-threshold: 500000000

  - type:
      - local
      - global
    units:
      - bytes
```

```
    vectors:
      - dns-flood
    limit-threshold: 100000000
```

Объект web-services предназначен для контроля входящего трафика к указанному префиксу. В конфигурации заданы три правила анализа.

Первое правило устанавливает статический порог для суммарного объёма трафика по всему объекту. В случае превышения значения 2 000 000 000 байт фиксируется аномалия на уровне префикса.

Второе правило выполняет подсчёт трафика по каждому IP-адресу внутри префикса и регистрирует аномалию при превышении значения 500 000 000 байт для отдельного адреса.

Третье правило предназначено для выявления трафика, соответствующего вектору dns-flood. Анализ осуществляется как на уровне всего объекта, так и на уровне отдельных IP-адресов. Превышение установленного порога рассматривается как аномалия соответствующего типа.

Представленный пример демонстрирует комбинированную настройку порогов, обеспечивающую контроль как суммарной нагрузки на объект, так и активности в отношении конкретных узлов и прикладных протоколов.

3.6 Активация объекта

После создания конфигурационного файла объект необходимо активировать.

1. Создать символическую ссылку в каталоге активированных объектов:

```
sudo ln -s /opt/spfc/etc/mo/<file>
/opt/spfc/etc/mo.enabled/<file>
```

2. Применить конфигурацию запуском службы анализатора:

```
sudo systemctl start analyzer
```

3. Проверить состояние службы:

```
sudo systemctl status analyzer
```

Объект считается активированным, если служба анализатора находится в состоянии active (running) и в выводе отсутствуют сообщения об ошибках обработки конфигурации.

3.7 Поведение системы после активации

После активации объекта:

- сервис начинает учитывать трафик, соответствующий указанным CIDR;
- выполняется подсчёт показателей согласно заданным правилам;
- при превышении пороговых значений фиксируется событие аномалии;
- информация сохраняется и становится доступной для последующего анализа.

4. Контроль и анализ детектированных аномалий

4.1 Просмотр журнала службы

Для контроля работы системы и выявления детектированных аномалий выполнить просмотр журнала службы analyzer:

```
sudo journalctl -xefu analyzer
```

Команда обеспечивает вывод записей журнала в режиме реального времени.

4.2 Идентификация аномалии в журнале

При срабатывании правила детекции в журнале формируется запись о начале действия. Пример записи:

```
Start action: ID=7be59492-4b31-4e17-84ec-eee4bd7151d7 group_id=0
managed_object="object1" subnet=NONE ip=1.2.3.1 vector=total-
traffic units=packets limit=threshold limit_value=1.000000Kp
start_value=0.000000 detect_value=17.523100Kp value=17.523100Kp
flow_spec_rules=
```

Параметр ID является уникальным идентификатором аномалии. При прекращении действия формируется запись:

```
Stop action: ID=7be59492-4b31-4e17-84ec-eee4bd7151d7 group_id=0
managed_object="object1" subnet=NONE ip=1.2.3.1 vector=total-
traffic units=packets limit=threshold limit_value=1.000000Kp
value=0.009743
```

Дополнительно фиксируется обработка отчёта:

```
ReportsJob: Post: 7be59492-4b31-4e17-84ec-eee4bd7151d7: Ok
```

Сообщение «Ok» подтверждает успешное формирование отчёта.

4.3 Получение детализированного отчёта

Для получения расширенной информации об аномалии выполнить запрос к базе данных событий с использованием идентификатора события:

```
db.reports.find({ id: "7be59492-4b31-4e17-84ec-eee4bd7151d7" })
```

В результате возвращается JSON-документ, содержащий агрегированные данные по событию, включая распределение трафика по протоколам, портам, странам, континентам и длинам пакетов.

4.4 Просмотр метрик объекта в базе данных

Для анализа метрик по векторам для объекта мониторинга, а также для его IP-адресов и подсетей, указанных в параметре cidrs, допускается использовать запросы к базе данных с метриками. Хранение метрик осуществляется в таблице default.graphite.

Пример запроса для получения значений по байтам за последний час:

```
SELECT
    toDateTime(Time) AS time,
    Path,
    Value
FROM default.graphite
WHERE Time >= toUInt32(now()) - 3600
  AND position(Path, '.object1.') > 0
  AND endsWith(Path, '.if_rx_octets')
ORDER BY Path, Time;
```

Пример запроса для получения значений по пакетам за последний час:

```
SELECT
    toDateTime(Time) AS time,
    Path,
    Value
FROM default.graphite
WHERE Time >= toUInt32(now()) - 3600
  AND position(Path, '.object1.') > 0
  AND endsWith(Path, '.if_rx_packets')
ORDER BY Path, Time;
```

Составили

Наименование организации, предприятия	**Должность исполнителя**	**Фамилия, имя, отчество**	**Подпись**	**Дата**

Согласовано

Наименование организации, предприятия	**Должность исполнителя**	**Фамилия, имя, отчество**	**Подпись**	**Дата**

Лист регистрации изменений

№ версии док-та	Дата изменения	Автор изменений	Изменения