База знаний

Настройка syslog-ng для передачи логов в SIEM-систему

Экспортировано 02/18/2025

Оглавление

Окружение

Astra Linux Special Edition 1.7[1]

Вопрос

Как настроить syslog-ng для передачи логов в SIEM-систему?

Ответ

ⓘ На всех узлах, которые будут работать по кастомным правилам, создана директория `/etc/syslog-ng/siem` и подключена следующей строчкой в конфигурационном файле `/etc/syslog-ng/syslog-ng.conf`:

```
@include "/etc/syslog-ng/siem/*.conf"
```

В общем случае для передачи логов необходимо создать в директории `/etc/syslog-ng/siem` конфигурационный файл `<имя_файла>.conf`, содержащий в себе следующие основные блоки:

- Источник логов (source)
  В блоке `source` указывается, откуда syslog-ng следует забирать логи (сокет, файл, сеть, stdout приложения и т. д.).
  Формат:

```
source <identifier> {
```

1 https://wiki.astralinux.ru/pages/viewpage.action?pageId=137563438

```
    source-driver(params);
    source-driver(params);
    ...
};
```

где `<identifier>` — уникальное, произвольное имя, формируемое согласно следующих требований:

- Имя идентификатора должно четко описывать его назначение: следует использовать названия, которые точно отражают роль источника, фильтра или цели.
- Чтобы легко различать типы объектов, имена идентификаторов необходимо начинать определенным образом:
  - `s_` — для источников (source);
  - `d_` — для целей (destination);
  - `f_` — для фильтров (filter);
  - `r_` — для правил перезаписи (rewrite);
  - `l_` — для путей логирования (log).

- Цель (destination)
  В блоке `destination` указывается, куда логи следует отправить (сокет, файл, сеть, stdin приложения и т. д.).
  Формат:

```
destination <identifier> {
    destination-driver(params);
    destination-driver(params);
    ...
};
```

- Фильтр логов (filter)
  Блок `filter` используется для фильтрации логов по содержимому.
  Формат:

```
filter <identifier> {
    <filter_type> ("<filter_expression>");
};
```

Примеры

Пример 1: Фильтрация по хосту

Фильтрация сообщений, полученных с хоста `example.com`:

```
filter f_host_filter {
    host("example.com");
};
```

Пример 2: Фильтрация по содержимому сообщения

Фильтрация сообщений, содержащих слово `error`:

```
filter f_message_filter {
    match("error" value("MESSAGE")); # Фильтруем сообщения, содержащие слово
"error"
};
```

Пример 3: Комбинированная фильтрация с операторами AND и OR

Для создания более сложных условий фильтрации можно комбинировать несколько условий логическими операторами:

- AND (И) — сообщение должно удовлетворять всем условиям.
  Фильтрация сообщений, одновременно и полученных от хоста `example.com`, и содержащих слово `error`:

  ```
  filter f_and_filter {
      host("example.com") and match("error" value("MESSAGE"));
  };
  ```

- OR (ИЛИ) — сообщение должно соответствовать хотя бы одному из условий.
  Фильтрация сообщений, полученных и от хоста `example1.com`, и от `example2.com`:

  ```
  filter f_or_filter {
      host("example1.com") or host("example2.com");
  };
  ```

Пример 4: Фильтрация с использованием NOT

Исключение сообщений, полученных от хоста `example.com`:

```
filter f_not_filter {
    not host("example.com");
};
```

Пример 5: Фильтрация с подстановочными символами (wildcards)

Фильтрация сообщений, полученных с хостов, имена которых начинаются с `myhost`:

```
filter f_wildcard_filter {
    host("myhost*" type(glob)); # Фильтруем хосты, имена которых начинаются с
"myhost"
};
```

`type(glob)` — отключает регулярные выражения, оставляя только подстановочные знаки.

- Правила перезаписи логов (rewrite)
  Блок `rewrite` позволяет изменять содержимое сообщений, приводя их к нужному виду, в том числе с помощью регулярных выражений.
  Формат:

```
rewrite <name_of_the_rule> {
    subst("<string or regular expression to find>", "<replacement string>",
value(<field name>), flags());
};
```

Примеры

Пример 1: Замена части сообщения

Используя директиву `subst()`, можно заменить любую часть текста в сообщении:

```
rewrite r_rewrite_subst {
    subst("старый_текст", "новый_текст", value("MESSAGE"), flags("global"));
};
```

- `"старый_текст"` — заменяемый текст;
- `"новый_текст"` — новый текст, который будет вставлен вместо старого;
- `value("MESSAGE")` — замена применяется к содержимому сообщения;
- `flags("global")` — замена будет применена ко всем вхождениям в сообщении, а не только к первому.

Пример 2: Последовательная замена

Выполнение нескольких замен подряд.

Сначала строка `IP` меняется на `IP-Address`, а затем строка `Address` на `Addresses`:

```
rewrite r_rewrite_multi_subst {
    subst("IP", "IP-Address", value("MESSAGE"));
    subst("Address", "Addresses", value("MESSAGE"));
};
```

Пример 3: Добавление тега

Для маркировки и дальнейшей фильтрации логов к сообщениям можно добавить тег:

```
rewrite r_add_tag {
    set-tag("MyCustomTag");
};
```

Пример 4: Удаление части сообщения

Удалить часть сообщения можно с помощью директивы `subst()`.

Первое вхождение строки `10.0.0.1` будет удалено из сообщения:

```
rewrite r_remove_ip {
    subst("10.0.0.1", "", value("MESSAGE"));
};
```

Чтобы удалить все повторы искомой части сообщения, а не только первое вхождение, необходимо использовать `flags("global")`:

```
rewrite r_remove_ip {
    subst("10.0.0.1", "", value("MESSAGE"), flags("global"));
};
```

Пример 5: Комплексное использование rewrite

```
rewrite r_clean_logs {
    subst("10\\.0\\.0\\.[0-9]{1,3}", "[HIDDEN-IP]", value("MESSAGE"), flags("gl
obal"));
    set-tag("SensitiveData");
    set("ModifiedProgram", value("PROGRAM"));
};
```

- `subst("10\\.0\\.0\\.[0-9]{1,3}", "[HIDDEN-IP]", value("MESSAGE"), flags("global"))` — замена всех IP-адресов, начинающиеся с `10.0.0.`, на строку "`[HIDDEN-IP]`". Двойной слэш нужен для экранирования точки в регулярном выражении.
- `set-tag("SensitiveData")` — добавление тега `SensitiveData` для дальнейшей фильтрации.
- `set("ModifiedProgram", value("PROGRAM"))` — изменение имени программы на `ModifiedProgram`.

- Путь логирования (log)
  Блок `log` определяет цепочку действий — от источника, через фильтры и правила перезаписи, к цели.
  Формат:

```
log {
    source(s1); source(s2); ...
    optional_element(filter1|parser1|rewrite1);
    optional_element(filter2|parser2|rewrite2);
    ...
    destination(d1); destination(d2); ...
    flags(flag1[, flag2...]);
};
```

Примеры конфигураций

Таким образом, примеры конфигураций могут иметь вид:

- на сервере-отправителе (КД):
  - Файл `/etc/syslog-ng/siem/destination.conf`:

    ```
    destination d_audit {
        syslog("audit.ald.pro" port(514) template("${MESSAGE} ${TAGS} \n"));
    };
    ```

    где:

- `"audit.ald.pro"` — syslog-ng считывает все файлы с директории, поэтому в файле указывается сервер аудита, чтобы избежать дальнейших повторений для различных правил;
- `port(514)` — логи отправляются на удаленный сервер, используя порт 514;
- `template("${MESSAGE} ${TAGS} \n")` — шаблон определяет, как логи будут выглядеть при отправке: в данном случае отправляется само сообщение с тегами, а в конце добавляется символ переноса строки.

- Файл `/etc/syslog-ng/siem/output-named.conf`:

```
source s_local_bind {
    file("/var/log/named/bind_log.log" follow-freq(1) flags(no-parse));
};

log {
    source(s_local_bind);
    rewrite {
        set-tag("tag-bind-query");
    };
    destination(d_audit);
};
```

где:

- Источник логов `s_local_bind`:
  - `"/var/log/named/bind_log.log"` — syslog-ng читает файл с логами BIND, расположенный по адресу `/var/log/named/bind_log.log`;
  - `follow-freq(1)` — syslog-ng проверяет файл на изменения раз в одну секунду;
  - `flags(no-parse)` — все данные передаются в неизмененном виде.
- Путь логирования `log {…}`:
  - `source(s_local_bind) … destination(d_audit)` — логи идут от источника `s_local_bind` к цели `d_audit`;
  - `set-tag("tag-bind-query")` — для идентификации и фильтрации к передаваемым логам с помощью `rewrite` добавляется тег `"tag-bind-query"`.

- на сервере-получателе (сервер журналирования):

> ⓘ Источником логов для сервера-получателя является `s_net`, описанный в `/etc/syslog-ng/syslog-ng.conf`.

- Файл, отвечающий за получение, обработку и запись логов `/etc/syslog-ng/siem/destination-named.conf`:

```
destination d_bind_logs {
    file("/var/log/aldpro/bind_query.log" template("${MESSAGE}\n"));
};

filter f_bind_logs {
```

```
    message("tag-bind-query");
};

filter f_exclude_ips {
    not (
        message("10.123.0.1[2-8]") or
        message("127.0.0.1") or
        message("mon.ald.pro")
    );
};

log {
    source(s_net);
    filter(f_bind_logs);
    filter(f_exclude_ips);
    rewrite {
        subst(" tag-bind-query,.source.s_local_bind", "", value("MESSAGE")
);
    };
    destination(d_bind_logs);
};
```

где:

- Цель `d_bind_logs`:
  - `"/var/log/aldpro/bind_query.log"` — логи записываются в файл `/var/log/aldpro/bind_query.log`;
  - `template("${MESSAGE}\n")` — записывается неизменённое сообщение.
- Фильтры логов:
  - `f_bind_logs` — фильтр пропускает только сообщения, которые содержат тег `"tag-bind-query".`
  - `f_exclude_ips` — фильтр игнорирует сообщения (ИЛИ):
    - от IP-адресов `10.123.0.1[2-8]`;
    - от IP-адреса `127.0.0.1`;
    - содержащие "`mon.ald.pro`".
- Перезапись `rewrite {…}`:
  - `value("MESSAGE")` — перезапись производится в теле сообщения;
  - `subst(" tag-bind-query,.source.s_local_bind", ""…)` — из сообщения удаляется строка "` tag-bind-query,.source.s_local_bind`".

Полезная информация

1. Подробная документация доступна по ссылке https://syslog-ng.github.io/admin-guide/README.
2. Настройка ротации журналов обязательна.
3. Выполнять следующие команды необходимо после каждого изменения файлов конфигурации:

```
syslog-ng -s # Проверка синтаксиса всех конфигурационных файлов
systemctl restart syslog-ng
```