Title: Remove fields from logs Description: GDPR, log anonymizer, reduce log volume

Output filter: remove-fields

This plugin removes fields before sending them to the output destination.

In the context of data protection regulations like GDPR you might need to mask data fields, especially when you hand over log data to third parties. This plugin can replace all occurrences of the original field values with `maskValuesString`. To mask the field values any other field, simply add a list of field names in `maskValuesInFields` (see example below).

Configuration

Add the following 'outputFilter' section to the Logagent configuration file. Note that you can use the plugin with multiple configurations for different event sources.

```
# tail web server logs
input:
  files:
    - '/var/log/*/access_log'

# log agent parses web server logs out of the box ...
# output filter to remove client_ip and user field in web server logs
outputFilter:
  remove-fields:
    module: remove-fields
    # JS regular expression to match log source name
    matchSource: !!js/regexp .*
    fields:
      - user
      - client_ip
      # json-path expressions are supported for nested fields
      # See: https://jsonpath.com/ online evaluator
      # - /request/header
      # - ../body
```

Masking strings matching removed fields' values

Additionally, use the `maskValuesString` directive to mask strings from removed fields' values in fields specified in the `maskValuesInFields` directive:

```
# tail web server logs
input:
  files:
```

```
    - '/var/log/*/access_log'

# log agent parses web server logs out of the box ...
# output filter to remove client_ip and user field in web server logs
outputFilter:
  remove-fields:
    module: remove-fields
    # JS regular expression to match log source name
    matchSource: !!js/regexp .*
    fields:
      - user
      - client_ip
      # json-path expressions are supported for nested fields
      # See: https://jsonpath.com/ online evaluator
      # - /request/header
      # - ../body
    # List of fields where the values from removed fields should be replaced with maskValues
    maskValuesInFields:
      - message
      - path
    # String to replace masked values from removed fields
    maskValuesString: "ANONYMIZED-DATA"
```

For example:

Assume an event where the a credit card number appears twice:

```
{
   message: "Credit Card No 12345",
   credit_card: "12345"
}
```

A simple removal of credit_card field would result in:

```
{
   message: "Credit Card No 12345",
}
```

The problem here is that even thought the `credit_card` field was removed, the credit card number from the `credit_card` field renamed exposed in the `message` field. To solve this problem we use `maskValuesInFields` along with `maskValuesString` directives:

```
maskValuesInFields
  - message
```

the result is

```
{
   message: "Credit Card No ANONYMIZED-DATA",
```

```
}
```

The value 12345 from the field `credit_card` was replaced in the `message` field, thus preventing the transmission of sensitive data.

Run Logagent with your config:

```
logagent --config logagent-example-config.yml -n httpd --yaml
```

The output does not contain the fields `client_ip` and `user`. Optionally the `user` is replaced with "ANONYMIZED-DATA" in the `message` and `path` field:

```
logSource:    httpd
_type:        access_common
remote_id:    -
method:       GET
path:         /user/ANONYMIZED-DATA
message:      GET /user/ANONYMIZED-DATA
http_version: HTTP/1.1
status_code:  304
size:         0
@timestamp:   Thu Apr 26 2018 22:02:26 GMT+0200 (CEST)
```