Event Stream Management API

Marius Scurtescu, Google

IETF100 Singapore

November 2017

Overview

- changes since IETF99
- API at a glance
  - get stream config
  - update stream config
  - get stream status
  - add/remove subject
  - verification
- future work
- open questions
- Q & A

Changes Since IETF99

- current draft: draft-scurtescu-secevent-event-stream-mgmt-api-00
- previous draft: draft-scurtescu-secevent-simple-control-plane-00
- added stream configuration update
- split out stream status
- verification event definition moved from delivery spec
- added security considerations

API: get stream config - Request

```
GET /set/stream HTTP/1.1
Host: transmitter.example.com
Authorization: Bearer eyJ0b2tlbiI6ImV4YW1wbGUifQo=
```

API: get stream config - Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8

{
  "aud": "http://www.example.com",
  "delivery": {
    "delivery_method": "urn:example:secevent:delivery:http_post",
    "url": "https://receiver.example.com/events"
  },
  "events": [
    "urn:example:secevent:events:type_1",
    "urn:example:secevent:events:type_2",
    "urn:example:secevent:events:type_3"
  ],
}
```

API: get stream config - Errors

Code	Description
401	authorization failed or it is missing
403	the Event Receiver is not allowed to read the stream configuration
404	there is no Event Stream configured for this Event Receiver

API: update stream config - Request

```
POST /set/stream HTTP/1.1
Host: transmitter.example.com
Authorization: Bearer eyJ0b2tlbiI6ImV4YW1wbGUifQo=
Content-Type: application/json; charset=UTF-8

{
  "aud": "http://www.example.com",
  "delivery": {
    "delivery_method": "urn:example:secevent:delivery:http_post",
    "url": "https://receiver.example.com/events"
  },
  "events": [
    "urn:example:secevent:events:type_1",
    "urn:example:secevent:events:type_2",
    "urn:example:secevent:events:type_3"
  ]
}
```

API: get stream status

```
GET /set/stream/status HTTP/1.1
Host: transmitter.example.com
Authorization: Bearer eyJ0b2tlbiI6ImV4YW1wbGUifQo=
VContent-Type: application/json; charset=UTF-8


HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "status": "enabled"
}
```

API: add subject - Request

```
POST /set/subjects:add HTTP/1.1
Host: transmitter.example.com
Authorization: Bearer eyJ0b2tlbiI6ImV4YW1wbGUifQo=
Content-Type: application/json; charset=UTF-8

{
  "email": "example.user@example.com"
}
```

Note: profiling specs MUST define how subjects are identified, "email" and below "phone_number" are provided only as examples.

API: add subject - Errors

Code	Description
400	the request body cannot be parsed or the request is otherwise invalid
401	authorization failed or it is missing
403	the Event Receiver is not allowed to add this particular subject
404	the subject is not recognized by the Event Transmitter; the Event Transmitter may chose to stay silent in this case and respond with 200
429	the Event Receiver is sending too many requests in a gvien amount of time

API: remove subject - Request

```
POST /set/subjects:remove HTTP/1.1
Host: transmitter.example.com
Authorization: Bearer eyJ0b2tlbiI6ImV4YW1wbGUifQo=
Content-Type: application/json; charset=UTF-8

{
  "phone_number": "+1 206 555 0123"
}
```

API: verification event - SET


```
{
  "jti": "123456",
  "iss": "https://transmitter.example.com",
  "aud": "receiver.example.com",
  "iat": "1493856000",
  "events": [
    "urn:ietf:params:secevent:event-type:core:verification" : {
      "state": "VGhpcyBpcyBhbiBleGFtcGxlIHN0YXRlIHZhbHVlLgo=",
    },
  ],
}
```

API: verification event - Trigger

```
POST /set/verify HTTP/1.1
Host: transmitter.example.com
Authorization: Bearer eyJ0b2tlbiI6ImV4YW1wbGUifQo=
Content-Type: application/json; charset=UTF-8

{
  "state": "VGhpcyBpcyBhbiBleGFtcGxlIHN0YXRlIHZhbHVlLgo="
}
```

Future Work: supported events

- transmitters declare supported events in stream configuration (read-only)
- receivers set supported events in stream configuration (read-write)
  - no wildcards, explicit list
- stream config to show potential list of delivered events (read-only)

Future Work: supported events - Example

```
{
  "aud": "636C69656E745F6964",
  "delivery": {...},
  "receiver_event_types_requested": [
    "http://schemas.openid.net/secevent/risc/event-type/account-disabled",
    "http://schemas.openid.net/secevent/risc/event-type/account-enabled",
    "http://schemas.openid.net/secevent/risc/event-type/identifier-changed",
  ],
  "transmitter_event_types_supported": [
    "http://schemas.openid.net/secevent/risc/event-type/account-disabled",
    "http://schemas.openid.net/secevent/risc/event-type/account-enabled",
    "http://schemas.openid.net/secevent/risc/event-type/account-deleted",
  ],
  "event_types_provided": [
    "http://schemas.openid.net/secevent/risc/event-type/account-disabled",
    "http://schemas.openid.net/secevent/risc/event-type/account-enabled",
  ],
}
```

Future Work: subject definition framework

- to allow profiles to define subject schemes

Open Question: get vs update

- how to distinguish read-only configuration attributes from read-write

Open Question: Discovery Document

- is this spec the right place to define a discovery document?

```
https://idp.example.com/.well-known/secevent-configuration

{
  "issuer": "https://idp.example.com/",
  "stream_endpoint": "https://idp.example.com/se/stream",
  "subject_add_endpoint": "https://idp.example.com/se/subject:add",
  "subject_remove_endpoint": "https://idp.example.com/se/subject:rm",
  "status_endpoint": "https://idp.example.com/se/status",
  "verification_endpoint": "https://idp.example.com/se/verify",
  "jwks_uri": "https://idp.example.com/se/jwks.json",
}
```

Q & A