
sticks Documentation

Release 0.1

Eurogiciel

May 13, 2015

1	Introduction	1
2	Installation	3
2.1	Sticks installation and configuration	3
3	Architecture	7
4	API References	9
4.1	Sticks REST API (root)	9
4.2	Sticks REST API (v1)	11
5	Indices and tables	15
	HTTP Routing Table	17

Introduction

Sticks is a Security As A Service project aimed at integrating security tools inside Openstack.

Installation

2.1 Sticks installation and configuration

2.1.1 Install from source

There is no release of Sticks as of now, the installation can be done from the git repository.

Retrieve and install Sticks :

```
git clone git://git.openstack.org/stackforge/sticks
cd sticks
python setup.py install
```

This procedure installs the `sticks` python library and a few executables:

- `sticks-api`: API service

Install a sample configuration file :

```
mkdir /etc/sticks
cp etc/sticks/sticks.conf.sample /etc/sticks/sticks.conf
```

2.1.2 Configure Sticks

Edit `/etc/sticks/sticks.conf` to configure Sticks.

The following shows the basic configuration items:

```
[DEFAULT]
verbose = True
log_dir = /var/log/sticks

rabbit_host = RABBIT_HOST
rabbit_userid = openstack
rabbit_password = RABBIT_PASSWORD

# Class of tracking plugin, ie redmine, trac, etc.
#tracking_plugin=

# Name of sticks role (default: sticks)
#sticks_role_name=sticks

[auth]
```

```
username = sticks
password = STICKS_PASSWORD
tenant = service
region = RegionOne
url = http://localhost:5000/v2.0

[keystone_authtoken]
username = sticks
password = STICKS_PASSWORD
project_name = service
region = RegionOne
auth_url = http://localhost:5000/v2.0
auth_plugin = password

[database]
connection = mysql://sticks:STICKS_DBPASS@localhost/sticks

DEFAULT]
```

2.1.3 Setup the database and storage backend

MySQL/MariaDB is the recommended database engine. To setup the database, use the `mysql` client:

```
mysql -uroot -p << EOF
CREATE DATABASE sticks;
GRANT ALL PRIVILEGES ON sticks.* TO 'sticks'@'localhost' IDENTIFIED BY 'STICKS_DBPASS';
EOF
```

Run the database synchronisation scripts:

```
sticks-dbsync upgrade
```

Init the storage backend:

```
sticks-storage-init
```

2.1.4 Setup Keystone

Sticks uses Keystone for authentication.

To integrate Sticks to Keystone, run the following commands (as OpenStack administrator):

```
keystone user-create --name sticks --pass STICKS_PASS
keystone user-role-add --user sticks --role admin --tenant service
```

Create the Helpdesk service and its endpoints:

```
keystone service-create --name Sticks --type helpdesk
keystone endpoint-create --service-id SECURITY_SERVICE_ID \
    --publicurl http://localhost:8888 \
    --adminurl http://localhost:8888 \
    --internalurl http://localhost:8888
```

2.1.5 Start Sticks

Start the API service :


```
sticks-api --config-file /etc/sticks/sticks.conf
```

Architecture

API References

4.1 Sticks REST API (root)

GET /

Return the version list

Return type list([APIVersion](#))

type [APILink](#)

API link description.

Data samples:

Json

```
{
  "href": "http://127.0.0.1:8888/v1",
  "rel": "self",
  "type": "text/html"
}
```

XML

```
<value>
  <type>text/html</type>
  <rel>self</rel>
  <href>http://127.0.0.1:8888/v1</href>
</value>
```

href

Type unicode

URL of the link.

rel

Type unicode

Relationship with this link.

type

Type unicode

Type of link.

type APIMediaType

Media type description.

Data samples:

Json

```
{
  "base": "application/json",
  "type": "application/vnd.openstack.sticks-v1+json"
}
```

XML

```
<value>
  <base>application/json</base>
  <type>application/vnd.openstack.sticks-v1+json</type>
</value>
```

base

Type unicode

Base type of this media type.

type

Type unicode

Type of this media type.

type APIVersion

API Version description.

Data samples:

Json

```
{
  "id": "v1",
  "links": [
    {
      "href": "http://127.0.0.1:8888/v1",
      "rel": "self",
      "type": "text/html"
    }
  ],
  "media_types": [
    {
      "base": "application/json",
      "type": "application/vnd.openstack.sticks-v1+json"
    }
  ],
  "status": "STABLE",
  "updated": "2014-08-11T16:00:00Z"
}
```

XML

```
<value>
  <id>v1</id>
  <status>STABLE</status>
  <updated>2014-08-11T16:00:00Z</updated>
```

```
<links>
  <item>
    <type>text/html</type>
    <rel>self</rel>
    <href>http://127.0.0.1:8888/v1</href>
  </item>
</links>
<media_types>
  <item>
    <base>application/json</base>
    <type>application/vnd.openstack.sticks-v1+json</type>
  </item>
</media_types>
</value>
```

id

Type unicode

ID of the version.

links

Type list(APILink)

List of links to API resources.

media_types

Type list(APIMediaType)

Types accepted by this API.

status

Type Enum(EXPERIMENTAL, STABLE)

Status of the version.

updated

Type unicode

Last update in iso8601 format.

4.2 Sticks REST API (v1)

4.2.1 Tickets

GET /v1/tickets

Return all tickets :type project: unicode

Return type TicketResourceCollection

GET /v1/tickets

Return ticket :type ticket_id: unicode

Return type TicketResource

POST /v1/tickets

Create a ticket :type data: TicketResource

Return type `TicketResource`

PUT `/v1/tickets`

Modify a ticket

DELETE `/v1/tickets`

Delete a ticket

type `TicketResource`

Type describing a ticket.

Data samples:

Json

```
{
  "project": "test_project",
  "title": "Ticket incident"
}
```

XML

```
<value>
  <title>Ticket incident</title>
  <project>test_project</project>
</value>
```

id

Type `unicode`

Id of the ticket.

project

Type `unicode`

Associated project of the ticket.

start_date

Type `date`

Start date.

status

Type `unicode`

Status.

title

Type `unicode`

Title of the ticket.

type `TicketResourceCollection`

A list of Tickets.

Data samples:

Json

```
{ }
```

XML


```
<value />
```

Indices and tables

- *genindex*
- *search*

/

GET /, 9

/v1

GET /v1/tickets, 11

POST /v1/tickets, 11

PUT /v1/tickets, 12

DELETE /v1/tickets, 12

A

APILink (webservice type), 9

APIMediaType (webservice type), 9

APIVersion (webservice type), 10

B

base (sticks.api.root.APIMediaType attribute), 10

H

href (sticks.api.root.APILink attribute), 9

I

id (sticks.api.root.APIVersion attribute), 11

id (sticks.api.v1.datamodels.ticket.TicketResource attribute), 12

L

links (sticks.api.root.APIVersion attribute), 11

M

media_types (sticks.api.root.APIVersion attribute), 11

P

project (sticks.api.v1.datamodels.ticket.TicketResource attribute), 12

R

rel (sticks.api.root.APILink attribute), 9

S

start_date (sticks.api.v1.datamodels.ticket.TicketResource attribute), 12

status (sticks.api.root.APIVersion attribute), 11

status (sticks.api.v1.datamodels.ticket.TicketResource attribute), 12

T

TicketResource (webservice type), 12

TicketResourceCollection (webservice type), 12

title (sticks.api.v1.datamodels.ticket.TicketResource attribute), 12

type (sticks.api.root.APILink attribute), 9

type (sticks.api.root.APIMediaType attribute), 10

U

updated (sticks.api.root.APIVersion attribute), 11