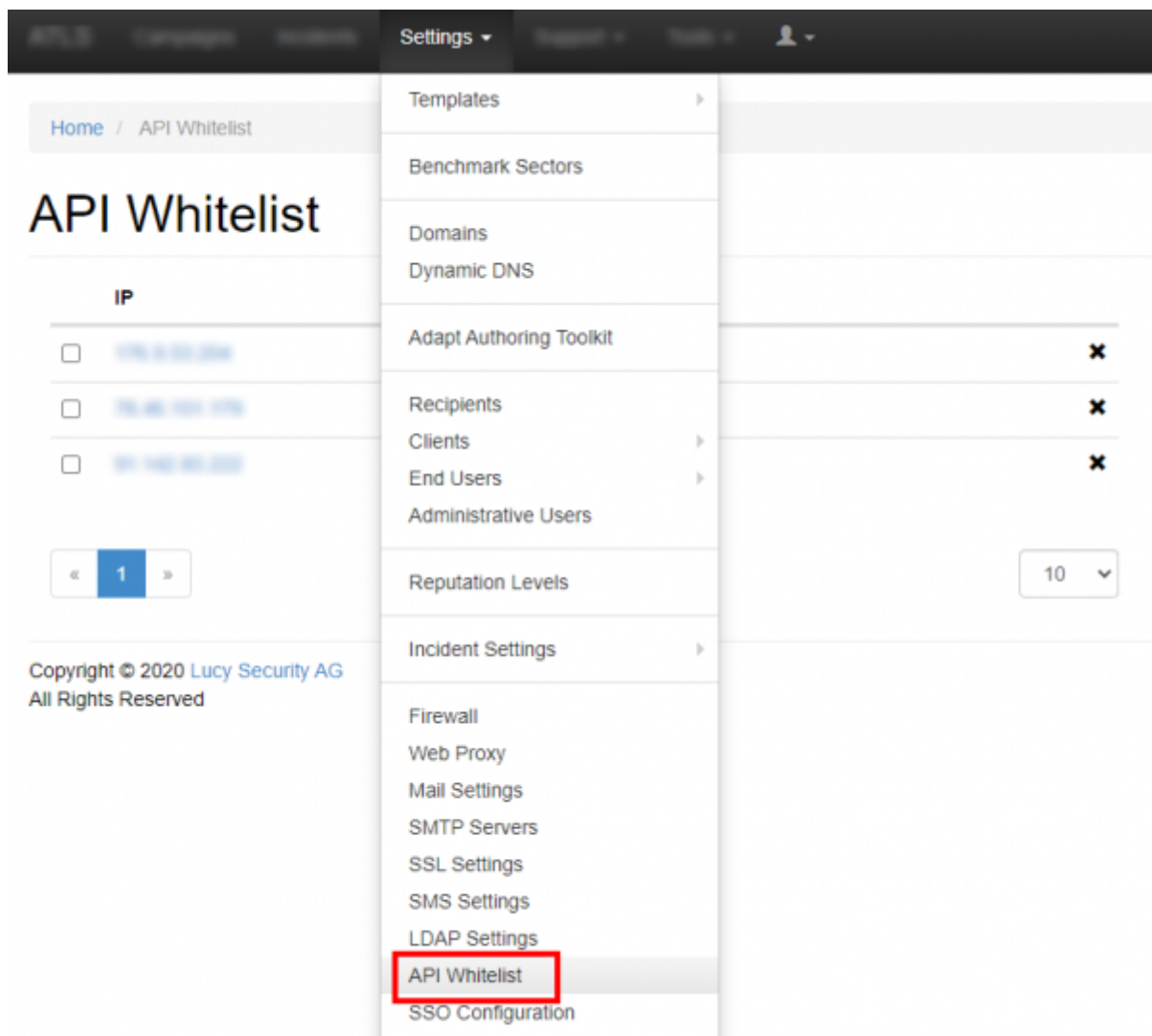# LUCY REST API - Background Info

Lucy API is a RESTful web service. The API does not keep the client's state and all requests can be performed in any order. Within LUCY the API can be activated under settings/API. You will need to specify the remote IP(s) which is allowed to communicate with the API.

Requirements

Some general requirements are:

- The service uses JSON as the only data interchange format, all API responses are in JSON, most request parameters are expected to be in JSON.
- All requests to API should have a "Content-Type" header set to "application/json".
- All requests should be performed strictly over HTTPS protocol. Plain HTTP requests to API will be rejected by server.

Authentication

Lucy API uses JWT (JSON web tokens) for authentication. The token should be obtained by sending an Authentication request before issuing any other requests (see Authentication request description below for details). The obtained authentication token should be specified in the "Authorization" HTTP header with "Bearer" scheme, for all subsequent requests. The only request that doesn't require authentication is the authentication request itself, all other requests require a token. Example:

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI
6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONF
h7HgQ
```

The client should store the token locally and keep it until the token expires. The expiration date and time is specified in the token header. Please check out https://jwt.io/ for JWT token formation description and libraries for various programming languages.

Resources

Resources are used when getting some data from Lucy API or creating/changing some objects in Lucy. Resource structure is always the same and does not depend on the type of action you do - whether you are getting information from Lucy or creating an object, the object representation doesn't change. For example, campaigns - when you get a list of campaigns or get a single campaign, the structure doesn't differ from what the server expects when you are creating a new campaign. The only difference is links - when you are posting/putting something to API, the server does not expect any links (and ignores them if they are present in request). The links are only used for exposing relations in GET requests. For specifying relations with other existing objects in system, please use integer IDs.

Example Usage

First you need to auth:

```
POST /api/auth HTTP/1.1
Host: phish.local
Content-Type: application/json
Cache-Control: no-cache

{"email":"test@test.com","password":"123"}
```

You will get the token in response, use it in all subsequent requests:

{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE1MDQ1OTI4OTksImp0aSI6Ik1QbktRTkxxs
cDdrOHh6YzRxenp3ZlhWd085TjRCMFIZ21UUXpwalFXdVE9IiwiaXNzIjoicGhpc2gubG9jYWwiLCJuYmYiO
jE1MDQ1OTI5MDAsImV4cCI6MTUwNDY3OTMwMCwiZGF0YSI6eyJ1c2VySWQiOjIsInZzZXJOYW1lIjoidGVz
dEB0ZXN0LmNvbSJ9fQ.2B0SafZcpF-
kyN0RqscfAthojX0iaEtcCegAYfZeG6BuBVF3pUxnYgclYqpLUGj57WwEPTdapSc1dqgbwW6l2w"}

Then create a group for example:

```
PUT /api/recipient-groups/ HTTP/1.1
Host: phish.local
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE1MDQ1OTI4OTksImp0aSI6Ik1Qbkt
RTkxscDdrOHh6YzRxenp3ZlhWd085TjRCMFRIZ21UUXpwwalFXdVE9IiwiaXNzIjoicGhpc2gubG9
jYWwiLCJuYmYiOjE1MDQ1OTI5MDAsImV4cCI6MTUwNDY3OTMwMCwiZGF0YSI6eyJ1c2VySWQiOjI
sInVzZXJOYW1lIjoidGVzdEB0ZXN0LmNvbSJ9fQ.2B0SafZcpF-
kyN0RqscfAthojX0iaEtcCegAYfZeG6BuBVF3pUxnYgclYqpLUGj57WwEPTdapSc1dqgbwW6l2w
Content-Type: application/json
Cache-Control: no-cache


{"name":"test recipient group"}
```

You will get something like this in response:

```
{"recipient-group":{"id":262,"name":"test recipient
group","usb_attack":false,"links":  [{"rel":"self","href":"/api/recipient-
groups/262"}]}}
```

Then you may create a recipient in that group:

```
PUT /api/recipient-groups/262/recipients HTTP/1.1
Host: phish.local
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE1MDQ1OTI4OTksImp0aSI6Ik1Qbkt
RTkxscDdrOHh6YzRxenp3ZlhWd085TjRCMFRIZ21UUXpwwalFXdVE9IiwiaXNzIjoicGhpc2gubG9
jYWwiLCJuYmYiOjE1MDQ1OTI5MDAsImV4cCI6MTUwNDY3OTMwMCwiZGF0YSI6eyJ1c2VySWQiOjI
sInVzZXJOYW1lIjoidGVzdEB0ZXN0LmNvbSJ9fQ.2B0SafZcpF-
kyN0RqscfAthojX0iaEtcCegAYfZeG6BuBVF3pUxnYgclYqpLUGj57WwEPTdapSc1dqgbwW6l2w
Content-Type: application/json
Cache-Control: no-cache

{"email":"oliver@test.com","name":"Oliver Test"}
```

You will get this response:

```
{"recipient":{"email":"oliver@test.com","phone":null,"name":"Oliver
Test","staff_type":null,"location":null,"division":null,"comment":null,"link
":null,"language_id":null,"recipient_group_id":262,"id":"158928","last_teste
d":null,"links":[{"rel":"self","href":"/api/recipients/158928"},
{"rel":"recipient_group","href":"/api/recipient-groups/262"}]}}
```

Detailed Documentation

A more detailed documentation can be found here.

From:

https://wiki.lucysecurity.com/ - **LUCY**

Permanent link:

https://wiki.lucysecurity.com/doku.php?id=api&rev=1505835619

Last update: **2019/07/25 12:51**