API Quick User Guide

Unknown Cyber, Inc

Table of Content

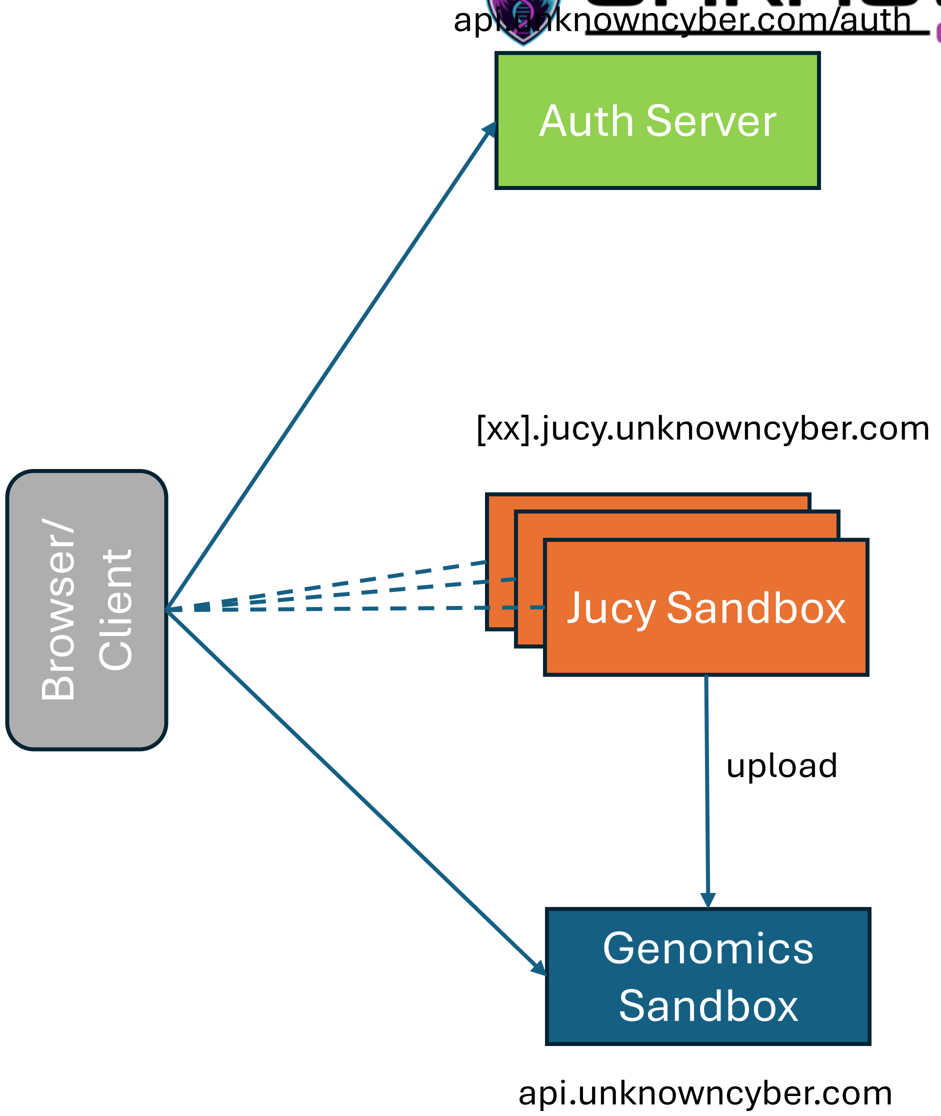
- Scope of this Document
- UC Architecture
- Authentication Schemes
- JWT Authentication Basics
- Authentication via API
- Upload to Jucy Sandbox via API
- Upload to Genomics Sandbox

Scope of this Document

- This is a quick guide to integrate with Unknown Cyber (UC) using the API.
- It presumes knowledge about what UC does.
- As a “quick guide” it is not intended to be a tutorial

UC Architecture

- UC services are accessed through three independent servers:
  - Authentication Server
  - Jucy Sandboxes
  - Genomics Sandbox
- Currently there are multiple Jucy Sandboxes.
  - One for each customer organization.
  - One for community/open users.
  - Each sandbox has its own domain.
  - This is likely to change within a quarter.
- Files uploaded to Jucy Sandbox are also sent to Genomics Sandbox.
- UC's dashboard integrates data from the two systems in the browser.

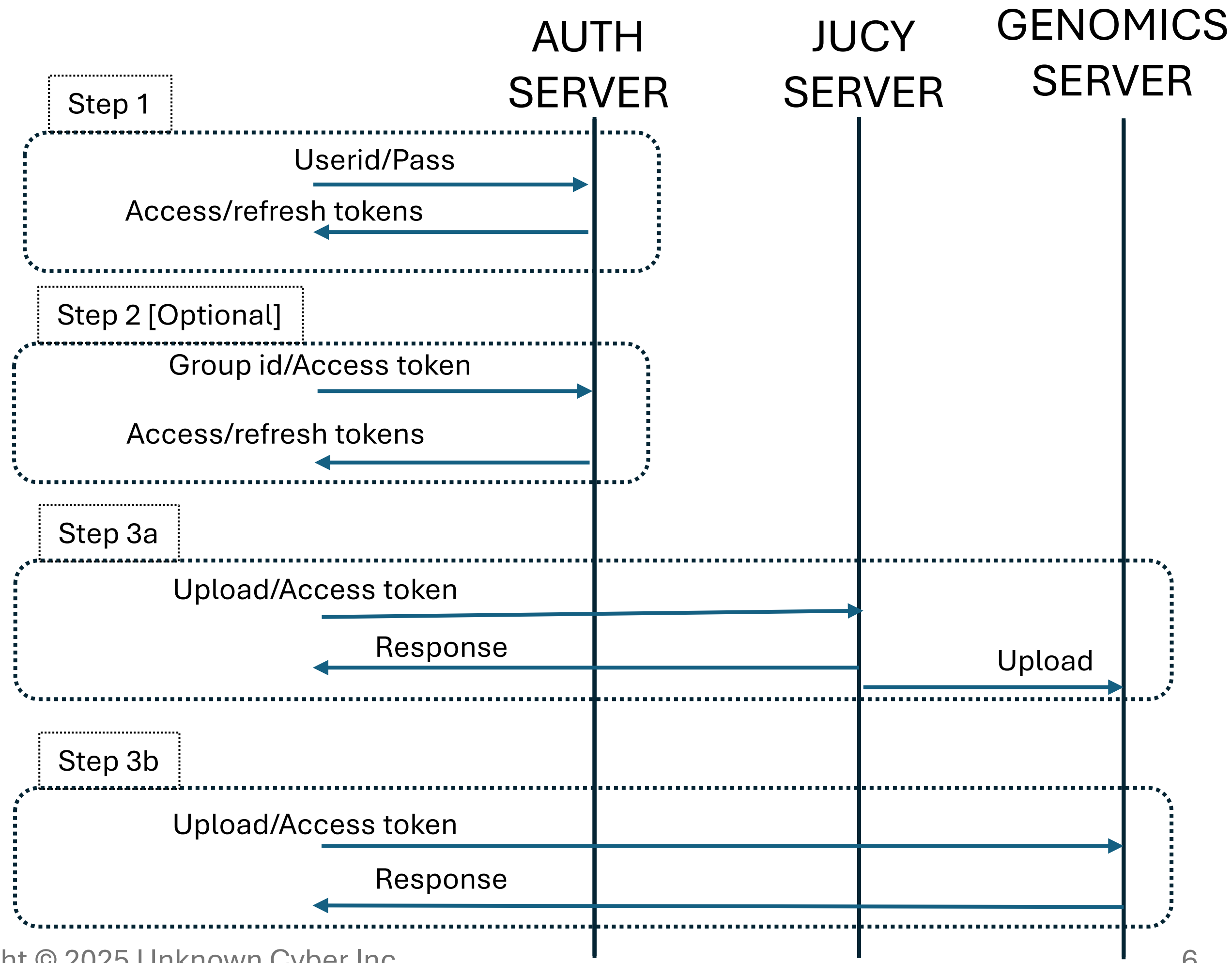


Authentication Schemes

- Jucy Sandbox supports only JWT (JSON Web Token) Authentication.
- Genomics Sandbox supports basic (userid/password), API, and JWT Authentication.
- Since JWT Authentication provides a better framework for integration with enterprise SSO, that is the recommended to use for enterprise applications.
- UC expects to deprecate other authentication schemes for enterprise users and integrate with a customer organizations's authentication server.

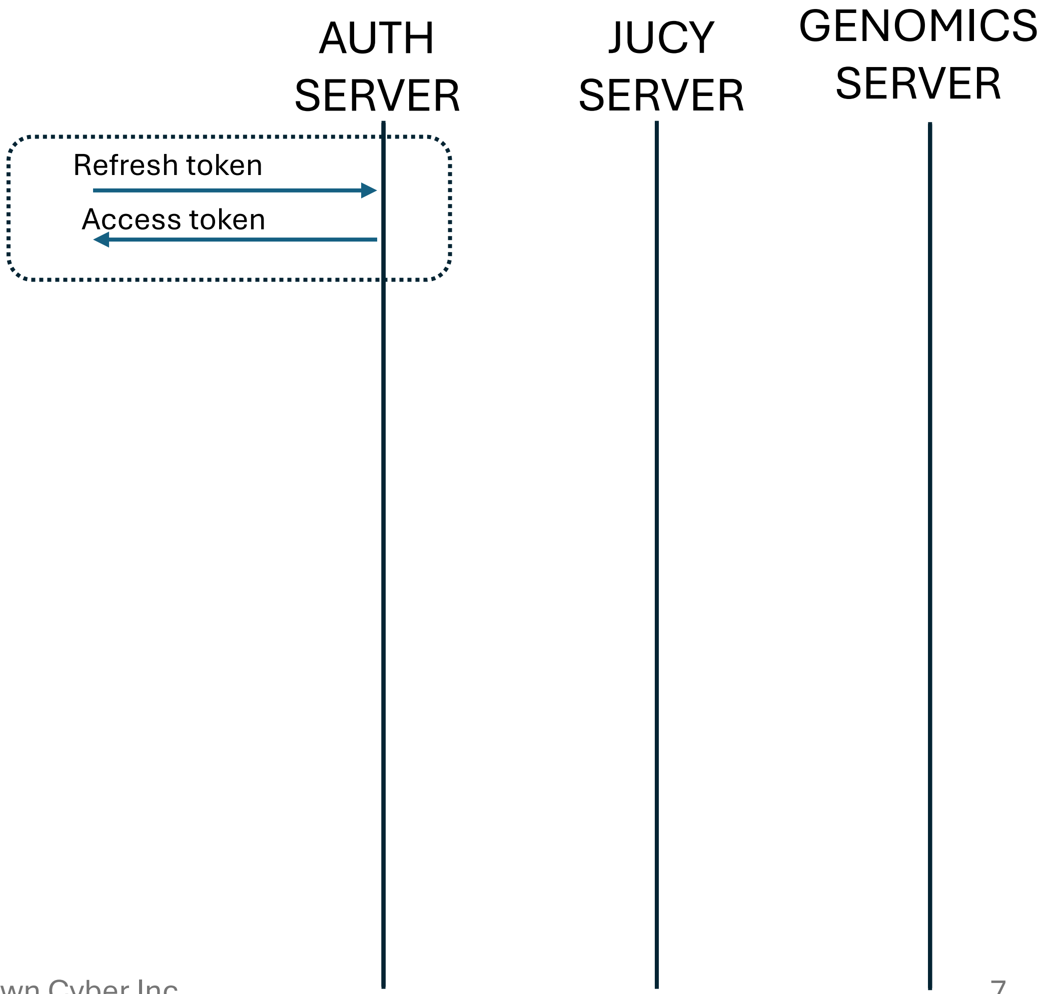
JWT Authentication Basics

- There are three steps to using JWT Authentication.
- Step 1:
  - Authenticate with Auth Server using Basic Authentication and get JWT Access and Refresh Tokens
- Step 2 [Optional]
  - Use JWT Access Token of the user to get JWT Access and Refresh Tokens for a Group.
- Step 3:
  - Use JWT Access Token to upload a file, whether to Jucy, Genomics, or both.
  - Files uploaded to Jucy are also uploaded to Genomics by Jucy.



JWT Authentication Token Lifetime

- Lifetime of Access Tokens?
  - 5 minutes.
- What to do after five minutes?
  - Acquire new access token using the refresh token
- Lifetime of Refresh Tokens?
  - 24 hours
- What to do after 24 hours?
  - Get new Access/Refresh token using Basic Authentication



Authentication API

Authenticate and get Access/Refresh Tokens

```
curl -X POST $MAGIC_API/auth/tokens/jwt/issue \
    -d username=$MAGIC_USER \
    -d password=$MAGIC_PASS
```

Extract tokens from JSON response

```
response = json.load(response)
if response['success']:
   access_token = response['resource']['access’]
   refresh_token = response['resource']['refresh']
```

Get Group Access/Refresh Token

```
curl -H "Authorization: Bearer $MAGIC_USER_ACCESS_TOKEN" \
    -H 'Content-Type: multipart/form-data' \
    -F "id=$MAGIC_GROUP_ID" \
    -X 'POST' "$MAGIC_API/auth/groups/login/"
```

Refresh Access Token

```
curl -H 'Content-Type: multipart/form-data' \
    -F "refresh=$MAGIC_REFRESH_TOKEN" \
    -X 'POST' "$MAGIC_API/auth/tokens/jwt/refresh/"
```

Upload to Jucy Sandbox via API

```
curl -H "Authorization: Bearer $MAGIC_ACCESS_TOKEN" \
    -F "file=@$1" \             # file name and data
    -F "tags_tasks=$TAGS" \     # TAGS (comma separated)
    -F "password=$PASSWORD" \ # Zip decryption password
    -F "timeout=120 " \         # execution time in seconds
    -F "route=internet" \       # Connect to internet
    -F "machine=win10" \        # Operating System
    -X POST "$JUCY_API/apiv2/tasks/create/file/"
```

Response (JSON Object)

```
{ "error":false         # true for error
  "errors":[],          # error messages of ‘error’ is true
  "data":{
       "task_ids":[355],  # task id for accessing status and analysis
       "message":"Task ID 355 has been submitted“
  },
 "url":["https://jucy.unknowncyber.com/submit/status/355/"]
}
```

Upload to Genomics Sandbox (1/2)

```
curl -H "Authorization: Bearer $MAGIC_ACCESS_TOKEN" \
    -F "filename=$1" \
    -F "filedata=@$1" \
    -F "tags=$TAGS" \
    -F "notes=$NOTES" \
    -F "password=$PASSWORD" \
    -X POST "$MAGIC_API/v2/files?$query_params"
```

Query Parameters

```
skip_unpack=false       # Make it true to disable unpacking
extract=true            # make it false to not extract a file to the top level
recursive=true          # make it false to extract only one level deep files
                        # else the extraction is done recursively
retain_wrapper=true     # make it false if you do not want to keep the wrapper
                        # in your collection (it is always kept in the system)
```

Upload to Genomics Sandbox (2/2)

Response

```
{   "success": true,            # false, if not successful
    "status": 200,
    "message": "OK",
    "errors": [],
    "links": {},
    "resources": [{
            "sha1": ..., "md5": ..., "sha256“: ..., "sha512":...,
            "filename": ...,
            "filesize": ...,
            "mimetype": "application/octet-stream",
            "status": 200,
            "message": "File uploaded",
            "job_id": ...,
            "_self": "http://api.unknowncyber.com/v2/files/ ...",

        }]}
```