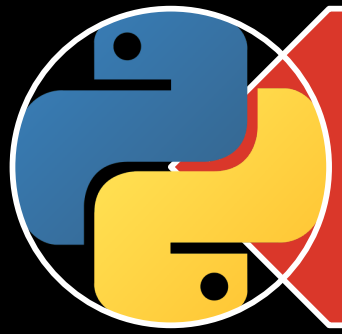
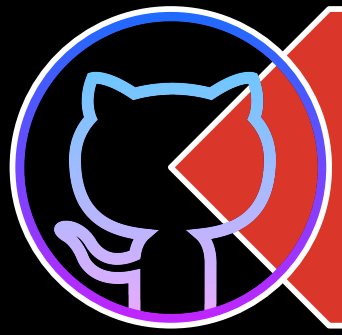
CROWDSTRIKE
FALCONPY
PYTHON 3 SDK FOR THE CROWDSTRIKE FALCON OAUTH2 API

CROWDSTRIKE

WHAT IS FALCONPY?



FalconPy is a Python 3 solution for interacting with the CrowdStrike Falcon OAuth2 API



Community-driven, FalconPy is provided as a completely free and Open Source project

CROWDSTRIKE

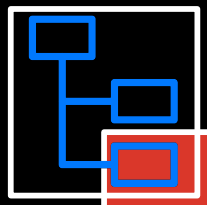
PACKAGE AND REPOSITORY DETAILS

- Package name: crowdstrike-falconpy
- Package URL: https://pypi.org/project/crowdstrike-falconpy/

- Repository name: falconpy
- Repository URL: https://github.com/CrowdStrike/falconpy

CROWDSTRIKE

FALCONPY COMPONENTS



Service Classes

- One class per service collection
- Method defined for every API operation
- Abstracts token interactions
- Token refresh

Uber Class

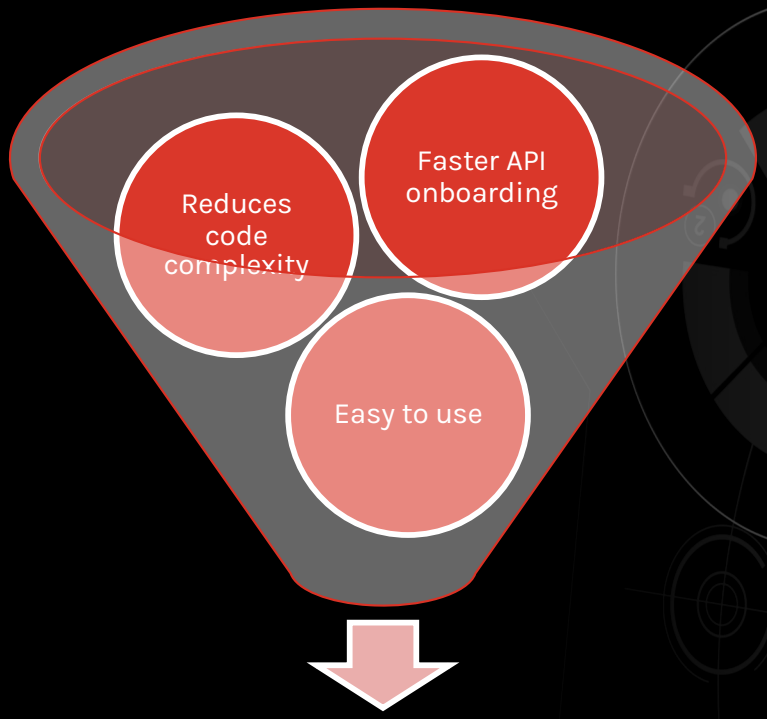
- A single class that provides a harness for the entire API
- Abstracts token interactions
- Token refresh
- Override

Shell

Initial release: 04/21

- Overlays API on a terminal shell
- Save / Load commands and payloads
- Prototype syntax
- Integrated debugger

FALCONPY BENEFITS



Reduces time to deliver your solution

CROWDSTRIKE

USAGE

CROWDSTRIKE

PACKAGE INSTALLATION AND REMOVAL

STANDARD INSTALLATION

$ python3 –m pip install crowdstrike-falconpy

BLEEDING EDGE / TESTING INSTALLATION

$ python3 –m pip install –I https://test.pypi.org/simple crowdstrike-falconpy

REMOVAL

$ python3 –m pip uninstall crowdstrike-falconpy

CROWDSTRIKE

AUTHENTICATING TO THE API

SERVICE CLASS

```
from falconpy import sample_uploads as FalconUploads

falcon = FalconUploads.Sample_Uploads(creds={
        "client_id": falcon_client_id,
        "client_secret": falcon_client_secret
    }
)
```

UBER CLASS

```
from falconpy import api_complete as FalconSDK

falcon = FalconSDK.APIHarness(creds={
        "client_id": falcon_client_id,
        "client_secret": falcon_client_secret
    }
)
```

CROWDSTRIKE

AUTHENTICATING TO THE API SERVICE CLASS VARIATIONS

Credential Authentication

```
from falconpy import sample_uploads as FalconUploads

falcon = FalconUploads.Sample_Uploads(creds={
        "client_id": falcon_client_id,
        "client_secret": falcon_client_secret
    }
)
```

Object Authentication

```
from falconpy import oauth2 as FalconAuth
from falconpy import sample_uploads as FalconUploads

auth = FalconAuth.OAuth2({
        "client_id": falcon_client_id,
        "client_secret": falcon_client_secret
    }
)

falcon = FalconUploads.Sample_Uploads(auth_object=auth)
```

Legacy Authentication

```
from falconpy import sample_uploads as FalconUploads

auth = FalconAuth.OAuth2({
        "client_id": falcon_client_id,
        "client_secret": falcon_client_secret
    }
)
try:
    token = auth.token()['body']['access_token']
except Exception:
    token = False

if token:
    falcon = FalconUploads.Sample_Uploads(access_token=token)
```

CROWDSTRIKE

SERVICE CLASS - SAMPLE UPLOADS API
UPLOADING A SAMPLE, DOWNLOADING IT, THEN DELETING IT

```
FILENAME = "testfile.jpg"
PAYLOAD = open(FILENAME, 'rb').read()
response = falcon.UploadSampleV3(file_name="newfile.jpg", file_data=PAYLOAD)
sha = response["body"]["resources"][0]["sha256"]
response = falcon.GetSampleV3(ids=sha)
open('serviceclass.jpg', 'wb').write(response)
response = falcon.DeleteSampleV3(ids=sha)
print(json.dumps(response, indent=4))
```

CROWDSTRIKE

UBER CLASS - SAMPLE UPLOADS API
UPLOADING A SAMPLE, DOWNLOADING IT, THEN DELETING IT

```
FILENAME = "testfile.jpg"
PAYLOAD = open(FILENAME, 'rb').read()
response = falcon.command('UploadSampleV3',
                          file_name="newfile.jpg",
                          data=PAYLOAD,
                          content_type="application/octet-stream"
                          )
sha = response["body"]["resources"][0]["sha256"]
response = falcon.command("GetSampleV3", ids=sha)
open('uberclass.jpg', 'wb').write(response)
response = falcon.command("DeleteSampleV3", ids=sha)
print(json.dumps(response, indent=4))
```

CROWDSTRIKE

WHAT'S NEXT

CROWDSTRIKE

FALCONPY ROADMAP

Outstanding service classes — Q2 2021

- Classes to cover remaining API service collections

FalconShell functionality expansion — Q3 2021

- Enhanced command, response and batch handling

v1.0 release — Q4 2021

- All base functionality available

* Dates are tentative estimates only

JOIN THE FALCONPY COMMUNITY

- GitHub project: https://github.com/CrowdStrike/falconpy
- GitHub wiki: https://github.com/CrowdStrike/falconpy/wiki
- Bug reports: https://github.com/CrowdStrike/falconpy/issues
- General discussion & requests: https://github.com/CrowdStrike/falconpy/discussions

- Python Package Index page: https://pypi.org/project/crowdstrike-falconpy/

CROWDSTRIKE
QUESTIONS
CROWDSTRIKE