Introducing the new Splunk SOAR SDK

DEV1495

.conf25 splunk>

Introducing the new Splunk SOAR SDK

Scott Odle

Senior Software Engineer | Splunk SOAR

.conf25 splunk>

Forward-looking statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

splunk> .conf25

SOAR Connectors (“Apps”)

Connect Splunk SOAR to a third-party service

~350 connectors available on Splunkbase

>50% built by Splunk

Others built by community members

Connector pain points

Example: CrowdStrike connector

52

supported actions

Each is fundamentally a REST API call, adapted to work within SOAR.

Lots of boilerplate and repeated code.

5,000

lines of Python

Relies on closed-source `phantom` library, so no coding assistance or type checking.

18,000

lines of JSON

Metadata for the connector, for each action, and for each of its inputs and outputs.

We edit this by hand!

Big feature updates are difficult

2024: adding actions to manage Indicators of Attack

+10

actions

- Manage RuleGroups
- Manage Rules
- Generate valid Rule parameters

+600

lines of Python

+2,500

lines of JSON

7 near-identical copies of the same output data structure

6 weeks

of work for one dev

It should not be this hard.

Building a better SDK

The features we want

All Python

No JSON

Open source

Great docs

Works with IntelliSense and CoPilot

Reusable input and output types

No SOAR server needed to build or test an app

Building a better SDK

The foundations we built on

splunk-soar-sdk

Start of development: October 2024
First beta release on PyPI: April 2025
1.0.0 GA Release: August 2025

Pydantic models for assets, inputs, and outputs

uv for managing app dependencies

Typer for a beautiful command-line interface

Building a better SDK

Available today!

```
uv tool install splunk-soar-sdk
```

https://pypi.python.org/project/splunk-soar-sdk

Works on Mac or Linux

Trying it again, with the SDK

Getting started

Requirements:

- A Mac or Linux machine
- uv
- Python 3.9 and 3.13 installed via uv

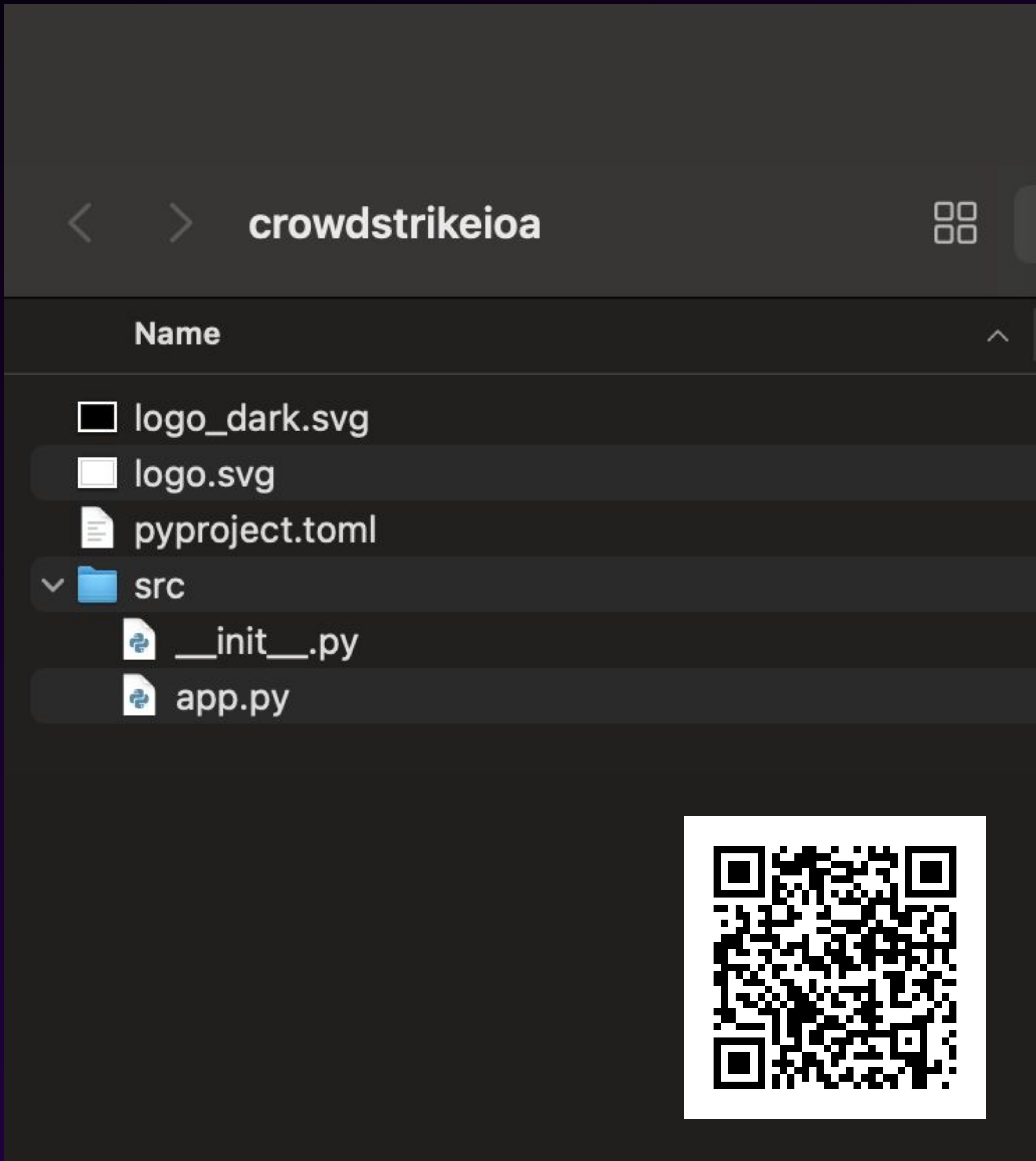
Install the SDK globally:

```
uv tool install splunk-soar-sdk
```

Start a new project:

```
soarapps init --app-dir crowdstrikeioa
```

Open your editor and follow along:
https://github.com/phantomcyber/sdk-crowdstrike-example



crowdstrikeioa

```
Name
logo_dark.svg
logo.svg
pyproject.toml
src
    __init__.py
    app.py
```

Asset Configuration

Now a Pydantic model

```
class Asset(BaseAsset):
    base_url: str = AssetField(default="https://api.crowdstrike.com")
    client_id: str
    client_secret: str = AssetField(sensitive=True)
```

Asset Configuration

Add a convenience method to get a CrowdStrike client

```
class Asset(BaseAsset):
   base_url: str = AssetField(default="https://api.crowdstrike.com")
   client_id: str
   client_secret: str = AssetField(sensitive=True)

   def get_client(self) -> CustomIOA:
       return CustomIOA(
           client_id=self.client_id, client_secret=self.client_secret,
           base_url=self.base_url, pythonic=True,
       )
```

Our first action

Listing IOA rule groups

```python
@app.action()
def list_rule_groups(params: ListGroupsParameters, asset: Asset) -> ListGroupsOutput:
   """List IOA rule groups, with an optional filter."""
   client = asset.get_client()
   result = client.query_rule_groups_full(
       filter=params.fql_query, offset=offset, limit=limit
   )
   return ListGroupsOutput(rule_groups=result.data)
```

Action Inputs and Outputs

Pydantic again - and we can create reusable objects!

```
class ListGroupsParameters(Params):
   fql_query: Optional[str] = Param(description="FQL query to filter groups")

class ListGroupsOutput(ActionOutput):
   rule_groups: list[IoaGroup]

class IoaGroup(ActionOutput):
   id: str = OutputField(cef_types=["crowdstrike ioa rule group id"])
   name: str
   description: str
```

Testing and building our app

We can run our action from the CLI, without installing SOAR:

```
python src/app.py action list-rule-groups -a crowdstrike_asset.json
```

When we're ready to build, we can do that from the CLI:

```
soarapps package build -o crowdstrike.tgz
```

App package includes all dependency wheels

- Retrieved from Python CDN, instead of building from source
- Faster builds
- Allows us to support x86 and ARM CPUs easily

Rewritten in the SDK

~~6 weeks~~

1 week

of work

~~600~~

488

lines of Python

Rewritten in the SDK

~~2,500~~ 0 lines of JSON

Migrate your existing app today

```
soarapps convert myapp
```

Automatically migrates your:

- App name, description, logos
- Asset model
- Action names and descriptions
- Action parameters
- Action outputs

Everything but the action logic!

Development roadmap

Now

- Basic apps and actions
- Ingestion
- Webhooks
- Custom views

- `soarapps init`
- `soarapps convert`

- Coroutines (async/await)
- Code splitting

Next

- Unit testing framework
- Tighter integration between platform and SDK

Later

- Use SDK in the App Wizard
- Upgrade to Pydantic 2.x

Try the SDK

```
uv tool install splunk-soar-sdk
```

PyPI: https://pypi.org/project/splunk-soar-sdk

CrowdStrike IOA app sample: https://github.com/phantomcyber/sdk-crowdstrike-example

HUGE THANKS to the dozens of Splunkers
who have contributed to the SDK

Questions? Find me on the show floor