Cyber Security Investigations with Jupyter Notebooks

Ian Hellen

Principal Software Development Engineer

Microsoft Threat Intelligence Center

@ianhellen @ianhelle

Microsoft
Threat
Intelligence
Center

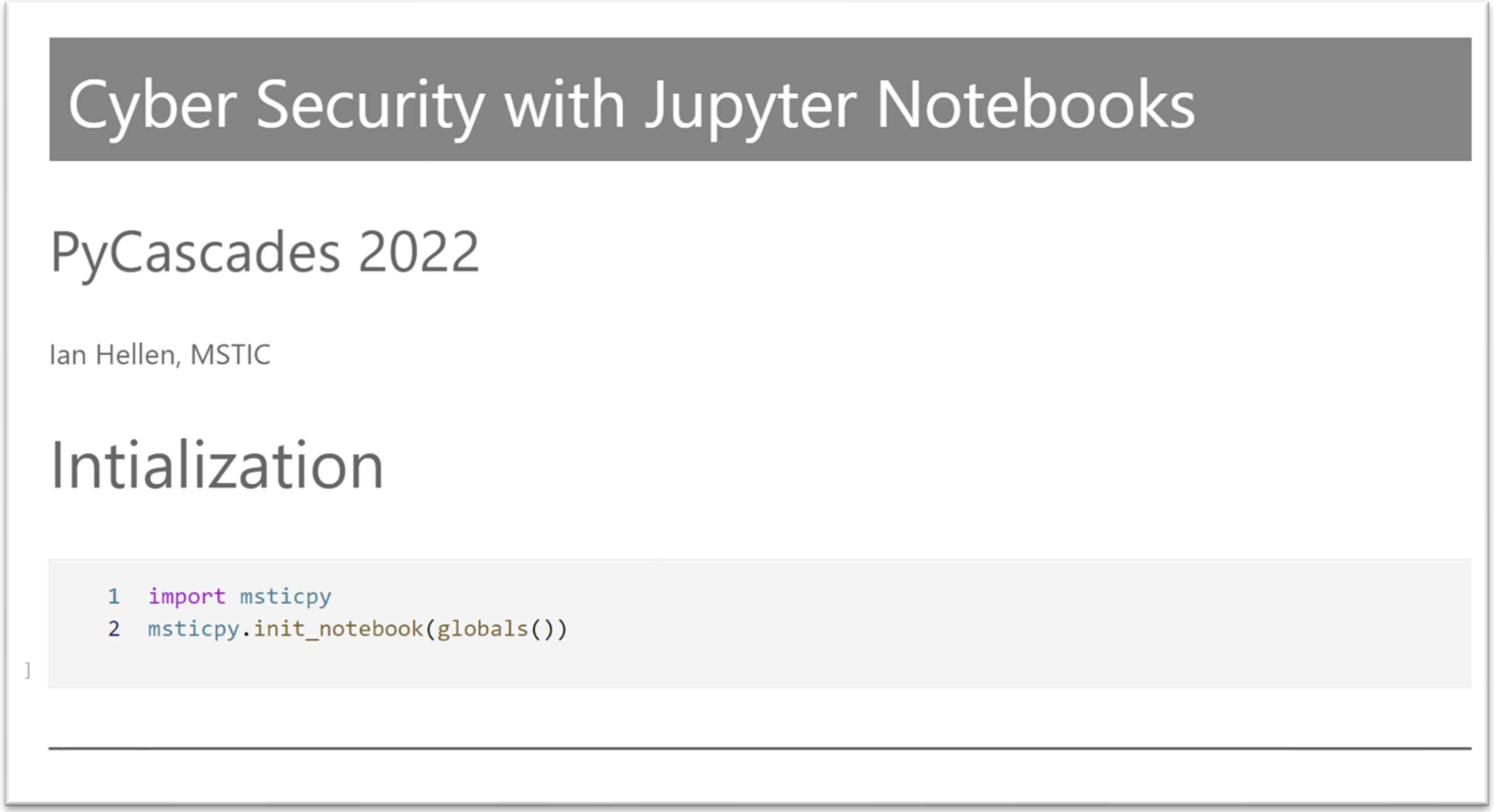
Jupyter Notebooks, InfoSec & MSTICPy

Overview

- Background
- Getting data
- Enriching data
- Visualizing threats
- Putting it together

- Looking at architecture and Python coding techniques we used to make our package

Companion notebook



Cyber Security with Jupyter Notebooks

PyCascades 2022

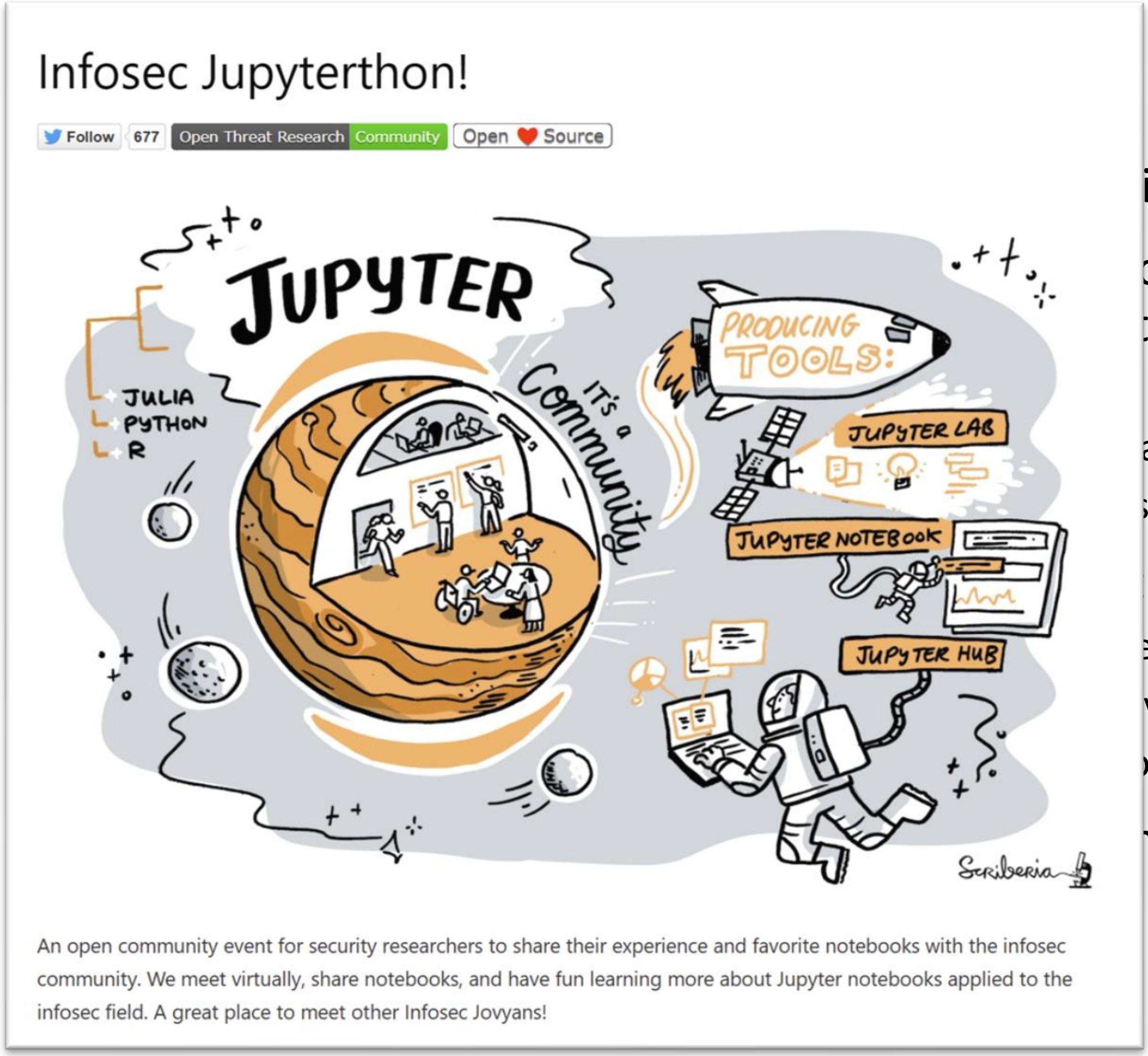
Ian Hellen, MSTIC

Intialization

```
import msticpy
msticpy.init_notebook(globals())
```

https://github.com/ianhelle/pycascades2022

What is great about Jupyter for Infosec?



Infosec Jupyterthon!

Follow 677 | Open Threat Research Community | Open ❤ Source

An open community event for security researchers to share their experience and favorite notebooks with the infosec community. We meet virtually, share notebooks, and have fun learning more about Jupyter notebooks applied to the infosec field. A great place to meet other Infosec Jovyans!

ific

of proprietary SOC
raints

your own workflow

ations

s

ext/code/visualization

ving progress

le document format

notebooks

https://infosecjupyterthon.com

Where there are ~~gaps~~ opportunities

Many Jupyter and Python features shaped by origins in Scientific and Data Science worlds

I want to break us out of the proprietary world of SIEMs but Jupyter/Python seems hard

InfoSec Analyst

Data
- Many log sources
- Template queries
- Consistent access pattern
- Specialized data - TI

Analytics
- Data is predominantly text (not numbers)

Viz
- Bar/line charts not too useful
- Domain-specific visualizations

High bar
- Learning curve
- Coding
- Example notebooks

SIEM: Security Information and Event Management
https://en.wikipedia.org/wiki/Security_information_and_event_management

MSTICPy

A toolset for cyber security investigators/hunters

Open source

Jupyter notebooks but also Python apps and scripts

Main components:

- Data access & queries
- Enrichment
- Visualizations
- Analysis

MSTICPy GitHub
https://github.com/microsoft/msticpy

MSTICPy Docs
https://msticpy.readthedocs.io

Microsoft
Threat
Intelligence
Center

Data Providers

(Config-driven func creation)

Without big data, you are blind and deaf and in the middle of a freeway.

Geoffrey Moore

In God we trust, all others bring data.

W Edwards Deming

Data providers

Analysts need ***lots*** of data, often from ***lots*** of different places

- Extensible data provider drivers masking:
  - Different access methods
  - Different query languages
  - Different authentication requirements
- Parameterized queries
- Returns pandas DataFrame

Drivers: MSSentinel, MSDefender, Splunk, MSGraph, Sumologic, OTRFData, LocalData, …

Our solution

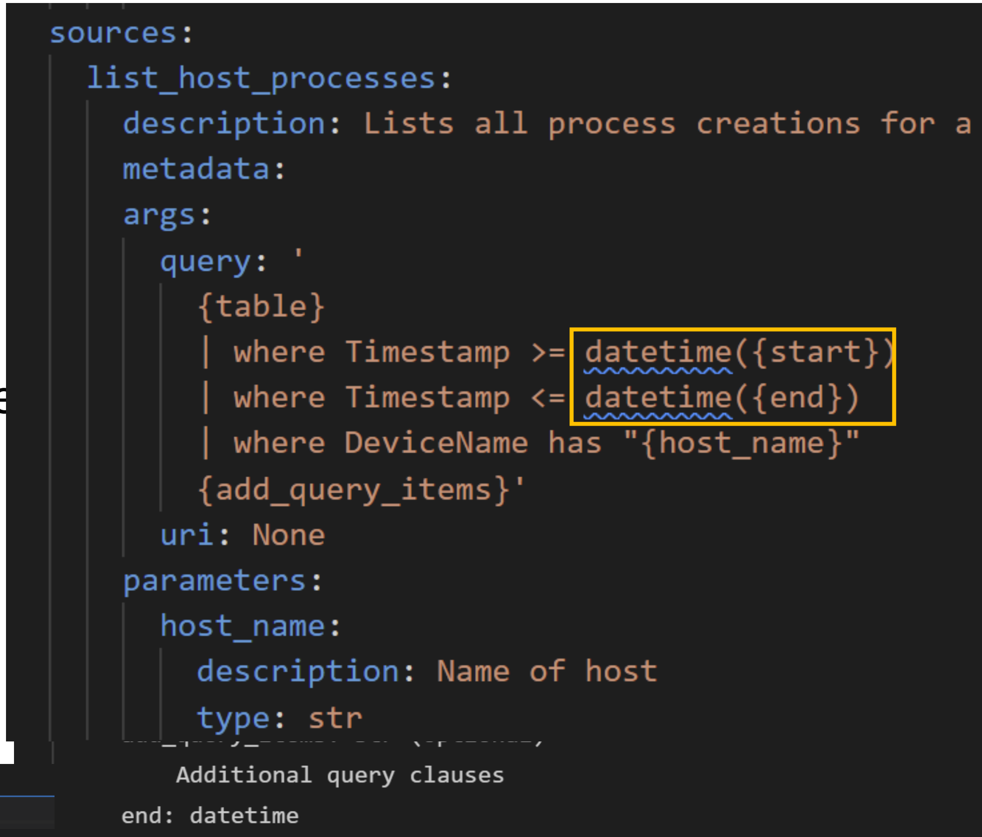
```
qp = QueryProvider(driver_name)
qp.connect(connection_string)
qp.list_queries()

# run a query
qp.query_name(params…)
```

Based on Intake - https://intake.readthedocs.io

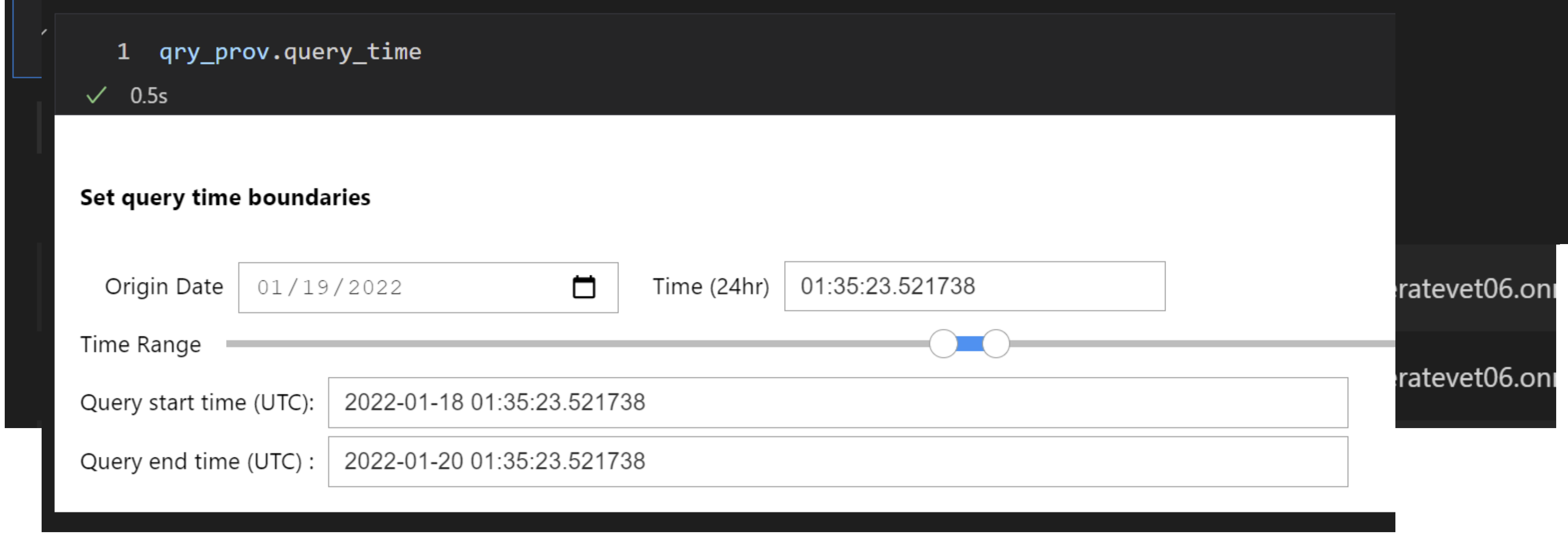
Data providers requirements

- Built-in queries - declarative (YAML)
- Invoke as Python functions
- Informative doc strings
- Query time management



```
sources:
  list_host_processes:
    description: Lists all process creations for a
    metadata:
    args:
      query: '
        {table}
        | where Timestamp >= datetime({start})
        | where Timestamp <= datetime({end})
        | where DeviceName has "{host_name}"
        {add_query_items}'
      uri: None
    parameters:
      host_name:
        description: Name of host
        type: str
```

```
        Additional query clauses
    end: datetime
```



```
qry_prov.query_time
```

✓ 0.5s

Set query time boundaries

Origin Date: 01/19/2022

Time (24hr): 01:35:23.521738

Time Range

Query start time (UTC): 2022-01-18 01:35:23.521738

Query end time (UTC) : 2022-01-20 01:35:23.521738

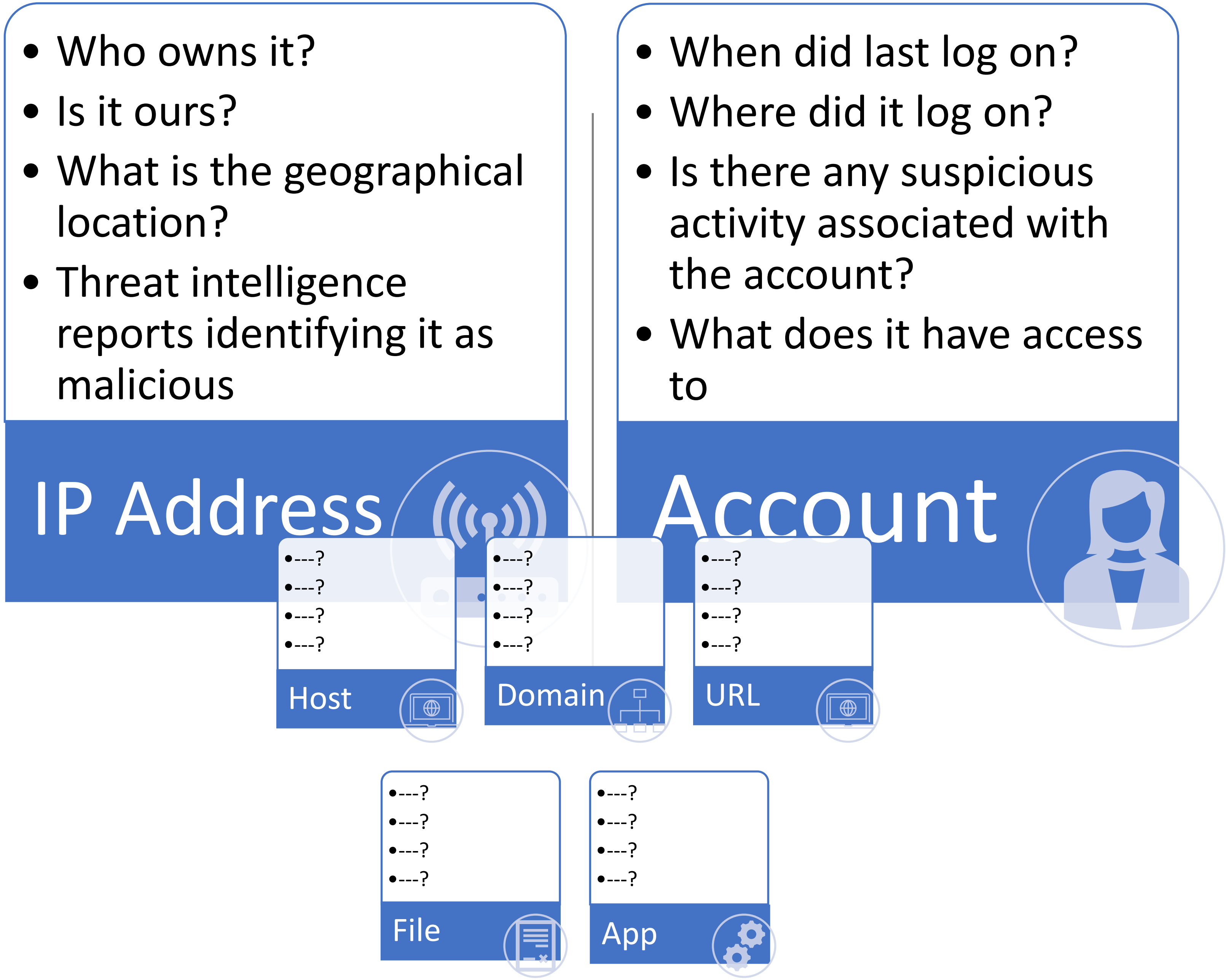
Demo

Config-driven functions

Enrichment (wrapping functions)

> *For me context is the key - from that comes the understanding of everything.*
>
> Kenneth Noland

Entities and context examples



- Who owns it?
- Is it ours?
- What is the geographical location?
- Threat intelligence reports identifying it as malicious

IP Address

- When did last log on?
- Where did it log on?
- Is there any suspicious activity associated with the account?
- What does it have access to

Account

- ---?
- ---?
- ---?
- ---?

Host

- ---?
- ---?
- ---?
- ---?

Domain

- ---?
- ---?
- ---?
- ---?

URL

- ---?
- ---?
- ---?
- ---?

File

- ---?
- ---?
- ---?
- ---?

App

Ideal enrichment functions

InfoSec analysts need a variety of contextual (enrichment) information

Check-list

☑ Contextual data is specific to each “entity”

☑ Needs to be easy to discover contextual data methods

☑ They should work in (roughly) the same way

- Common parameters
- Common return types

Our initial approach

We built a bunch of enrichment functions:

- Geo-location of IPs
- Domain/IP ownership
- Threat intelligence
- others

Some extensible – so some consistency.

Check-list

☒ Specific to each “entity”

☒ Needs to be easy to discover

☒ Should work in the same way

Fails

☹ No special association between the enrichment function and what you wanted to enrich

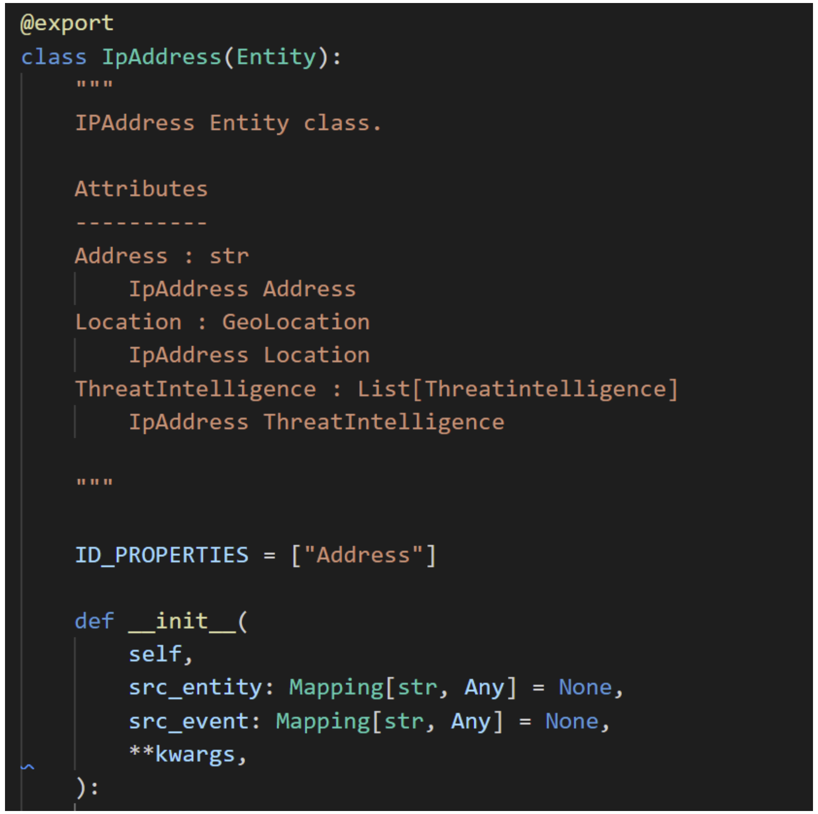
☹ You had to know what to import from where

☹ Every provider interface was a little bit different:

- input parameters
  - Names
  - Types
- output format

Entity-centric *pivot* functions

Example entities			
Account	Alert	AzureResource	CloudApplication
Dns	**File**	**Host**	IoTDevice
IpAddress	Mailbox	Malware	NetworkConnection
Process	RegistryKey	RegistryValue	**Url**



```
@export
class IpAddress(Entity):
    """
    IPAddress Entity class.

    Attributes
    ----------
    Address : str
        IpAddress Address
    Location : GeoLocation
        IpAddress Location
    ThreatIntelligence : List[Threatintelligence]
        IpAddress ThreatIntelligence

    """

    ID_PROPERTIES = ["Address"]

    def __init__(
        self,
        src_entity: Mapping[str, Any] = None,
        src_event: Mapping[str, Any] = None,
        **kwargs,
    ):
```

But we'd already written the enrichment code

- Wrap enrichment functions for consistent interface
- Dynamically attach to entities

Demo

Pivot mechanism

Visualization

A soul never thinks without a picture

Aristotle

Visualization Requirements

InfoSec analysts need a domain-specific visualizations.

Criteria

- Time-based charts are critical
- Some specific visualization types:
  - Process trees
  - Graphs
- Interactivity
  - zooming, hover for more info
- Discoverability

Visualization in MSTICPy

No silver bullet – fair bit of coding

- Standardized on Bokeh
- Support generic input
  - Only need a timestamp
- Pandas accessors

Visualizations

- Event timeline
- Event duration
- Matrix (interaction)
- Time series
- Process tree

Inspirations & Credits

Bokeh Periodic table

https://docs.bokeh.org/en/latest/docs/gallery/periodic.html

Myrthings – CatScatter

https://github.com/myrthings/catscatter

Demo

Visualization

Composability

Everything should work together!

- Ideally one function output can be the input for another
- Chained pipelines for repeatable analysis

Standing on the shoulders of giant (pandas)

Make everything pandas-centric

- Functions should always accept DataFrames as input
- Return DataFrame by default
- Use pandas accessors

Inspirations

Pandas https://pandas.pydata.org/docs/development/extending.html

Hvplot (and others) https://hvplot.holoviz.org/

Demo

Composable functions

Conclusion

- What seems like a quick project can grow on you!
- Organic growth is not always organized growth
- Understand the special requirements of your domain
- Plan for:
  - Discoverable functionality
  - Consistent experience (inputs/outputs)
  - Building the right visualizations
- Python lets you bend the very laws of space and time!

References

What	Where
The notebook	https://github.com/ianhelle/pycascades2022
MSTICPy GitHub repo	https://github.com/microsoft/msticpy ⭐ (leave us a star if you like it!)
MSTICPy Docs	https://msticpy.readthedocs.io
Simple feature notebooks	https://github.com/microsoft/msticpy/tree/main/docs/notebooks
Scenario notebooks	https://github.com/Azure/Azure-Sentinel-Notebooks
Contact	@ianhellen @ianhelle @msticpy msticpy@microsoft.com