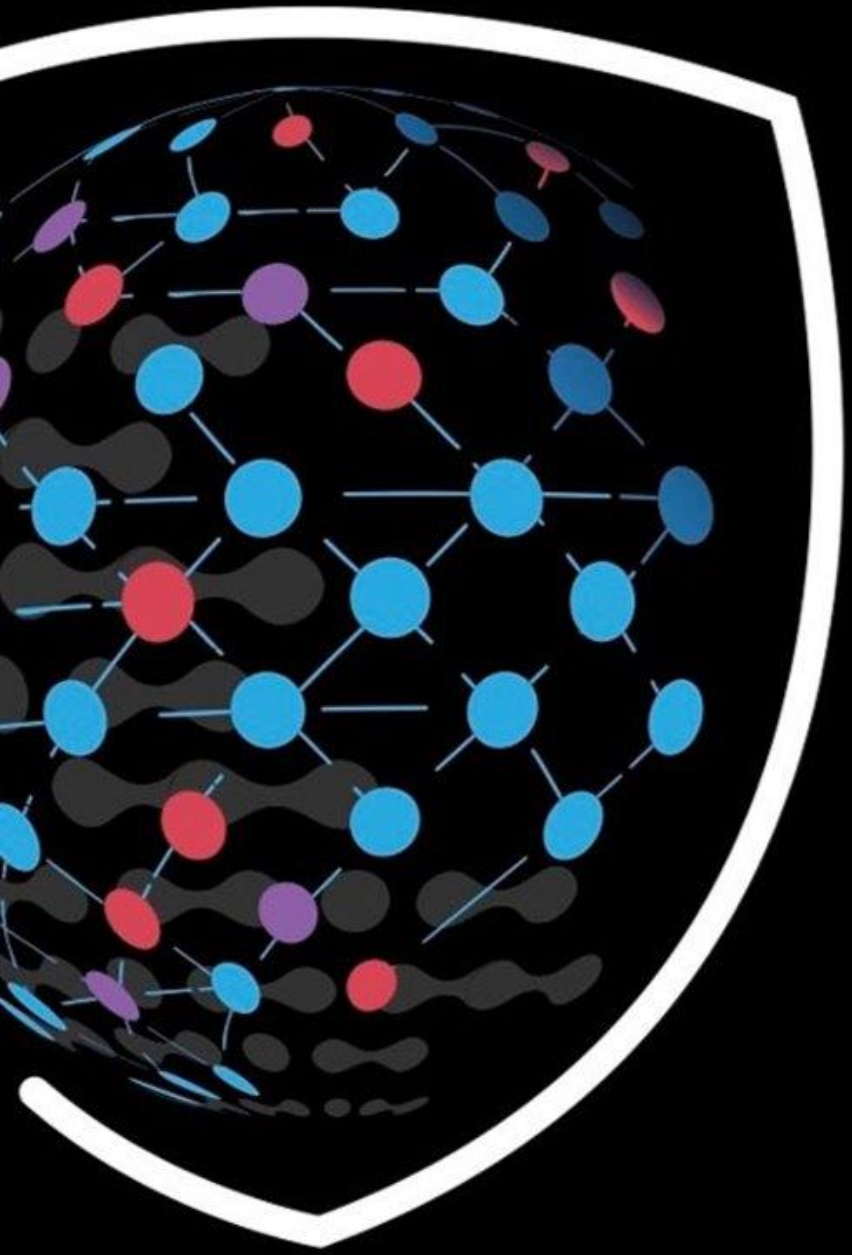


OPEN THREAT RESEARCH

EMPOWERING THE INFOSEC COMMUNITY

Análisis de Datos en Seguridad Defensiva via **Jupyter Notebooks**

Quiénes Somos?

Amantes de la Colaboración y Fuente Abierta

Roberto Rodriguez 🇵🇪
@Cyb3rWard0g

- Microsoft Threat Intelligence Center (MSTIC)

Jose Rodriguez 🇵🇪
@Cyb3rPandaH

- MITRE - ATT&CK

- Colaboración Abierta ♡
- Threat Hunter Playbook @HunterPlaybook
- Mordor @Mordor_Project
- OSSEM @OSSEM_Project
- Blacksmith & more..

https://github.com/OTRF

Agenda

1) Introducción a Jupyter Notebooks
   - Opciones de instalación
   - El proyecto Binder
2) Introducción a Pandas
   - Importing the Library
3) El proceso de Análisis de Datos
4) Necesitamos data?... Mordor
   - Descargando sets de datos
   - De datos a Dataframe
5) Algunos ejemplos de técnicas para Análisis de Datos

Espera...
Whát? 🤔

¿Qué haremos?

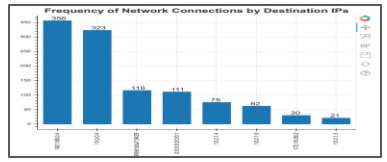
La Estructura del Workshop:

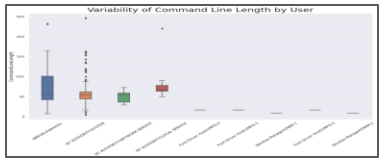
Plataformas para Análisis

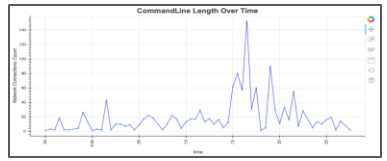
Set de Datos **MORDOR**

La Librería Pandas

Técnicas de Análisis







Pre - Requisitos

Todo lo que necesitas esta aqui:

https://otrf.github.io/workshop-ekoparty-bluespace-2020

Infosec Jupyter Book

Introducción a Jupyter Notebooks

¿Qué son Jupyter Notebooks?

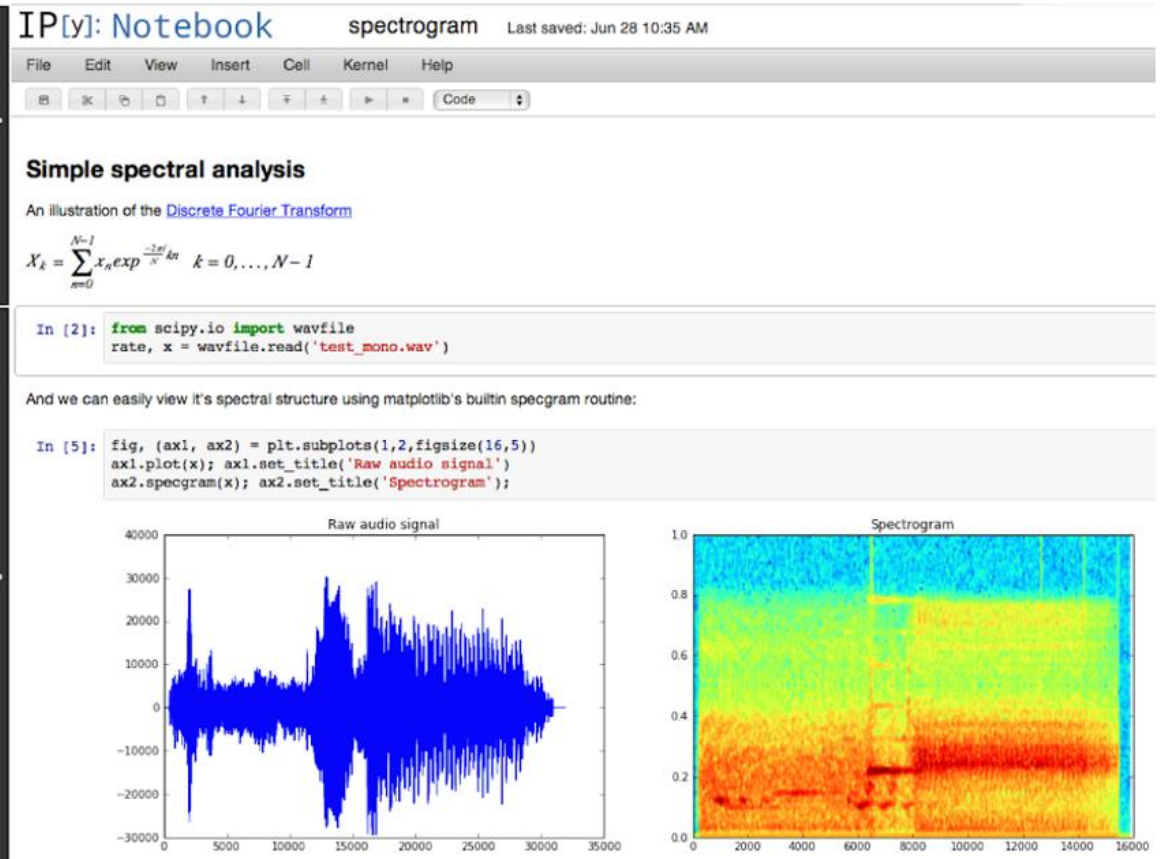
- Son documentos que podemos accesar a través de una interface web. Nos permite gestionar y almacenar:
  - **Input:** Código (por ejemplo Python)
  - **Output**: Resultados de Código ejecutado
- Excelente para contar la historia de la investigacion desarrollada.

https://jupyter.org/

Python Interpreter -> IPython -> Jupyter

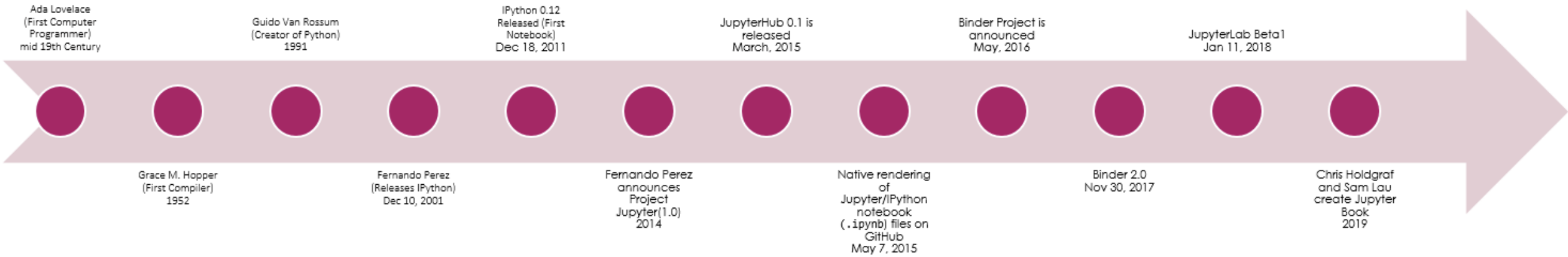
```
Robertos-MacBook-Pro:~ wardog$ python3
Python 3.7.2 (default, Feb 12 2019, 08:16:38)
[Clang 10.0.0 (clang-1000.11.45.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hola Python!!')
Hola Python!!
>>> 12 * 2
24
>>>
```

```
Robertos-MacBook-Pro:GitHub wardog$ ipython
Python 2.7.10 (default, Oct  6 2017, 22:29:07)
Type "copyright", "credits" or "license" for more information.

IPython 5.7.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: print('Hola IPython!!')
Hola IPython!!

In [2]: 12 * 2
Out[2]: 24

In [3]:
```



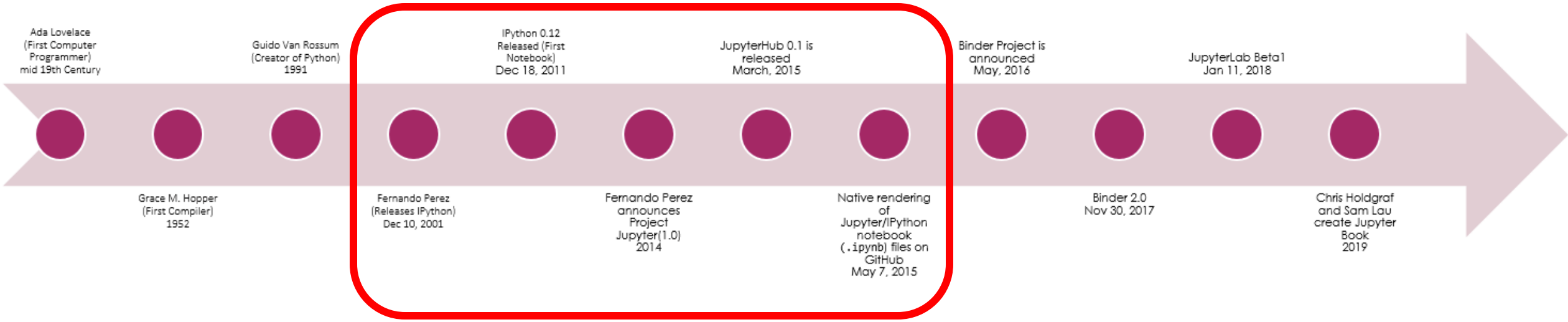
https://jupyter.org/

IPython -> Jupyter



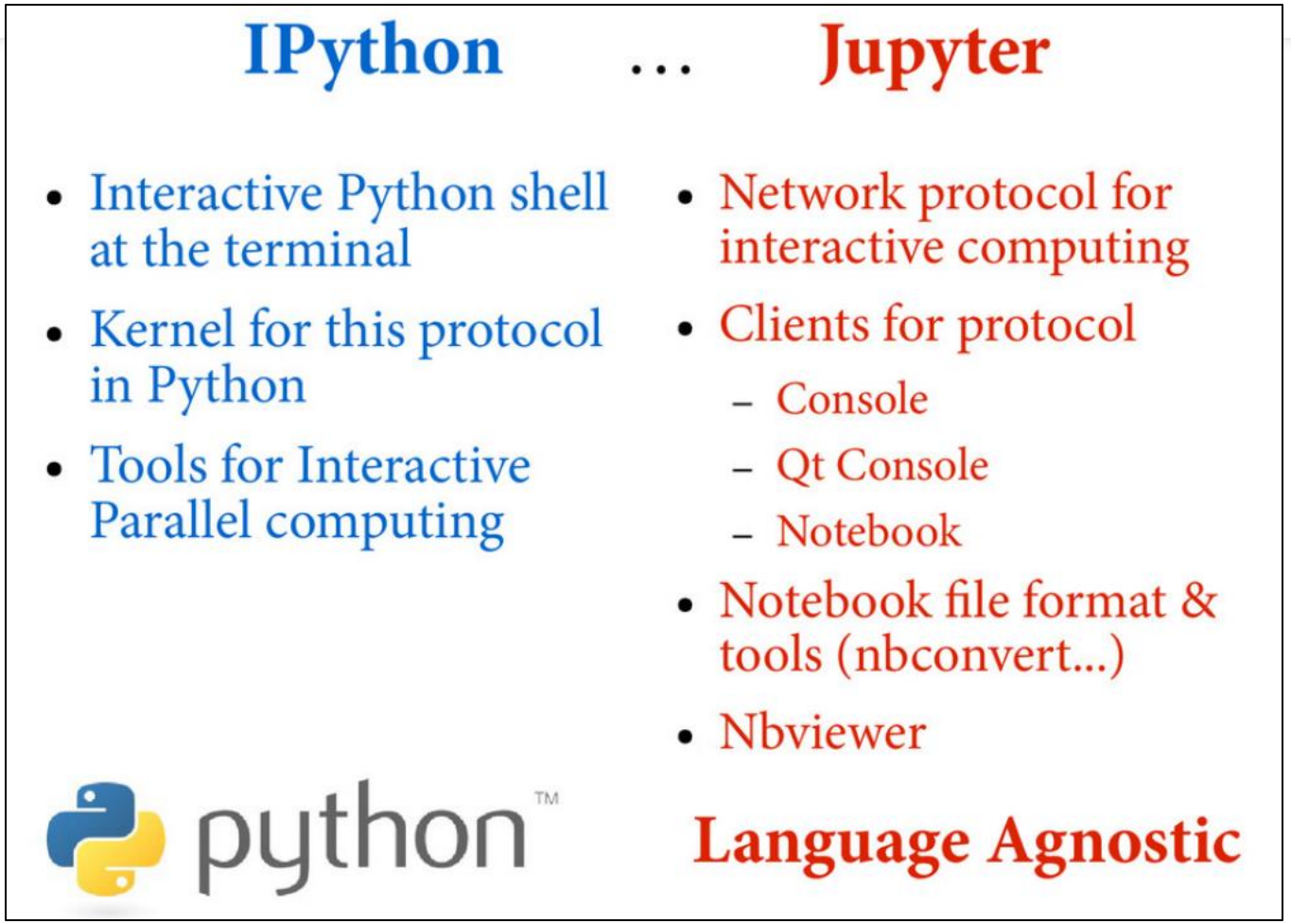
https://infosecjupyterthon.com/notebooks/1_what_about_notebooks.html

IPython -> Jupyter



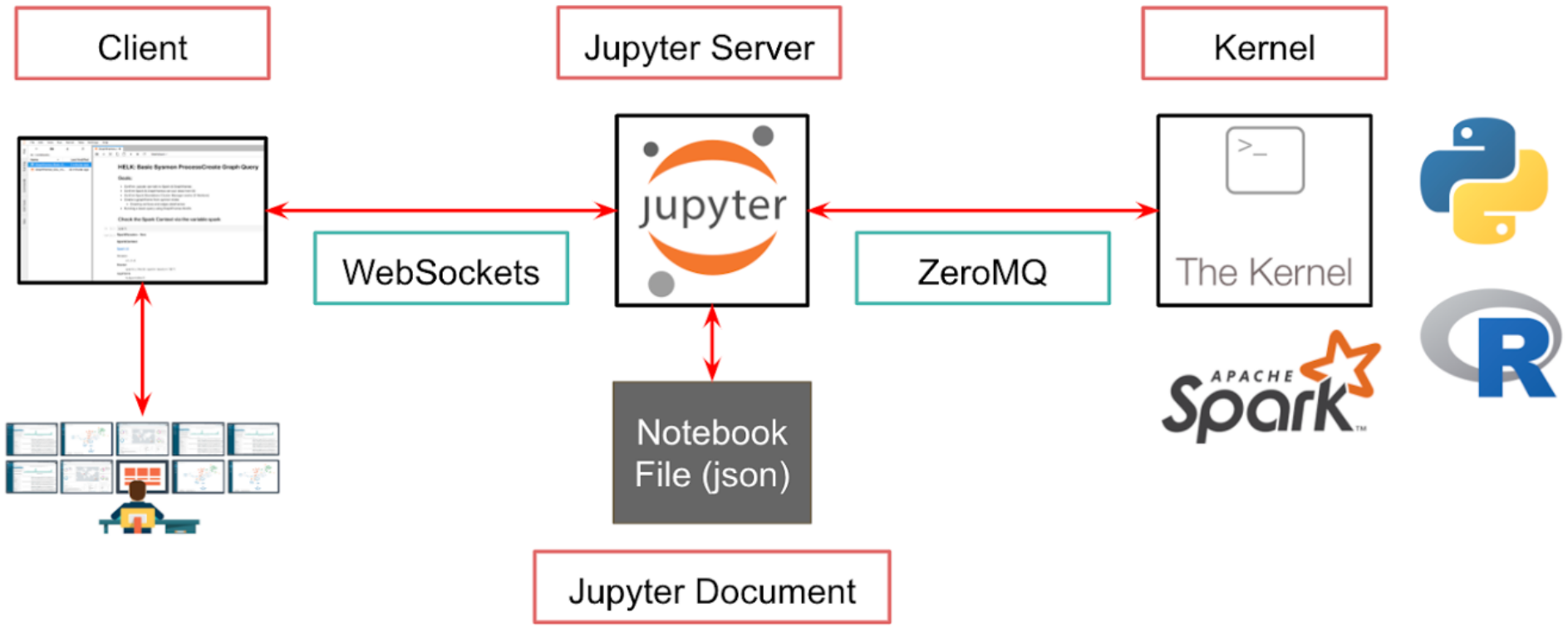
https://infosecjupyterthon.com/notebooks/1_what_about_notebooks.html

IPython -> Jupyter



https://speakerdeck.com/fperez/project-jupyter?slide=5

La arquitectura básica de Jupyter Notebooks



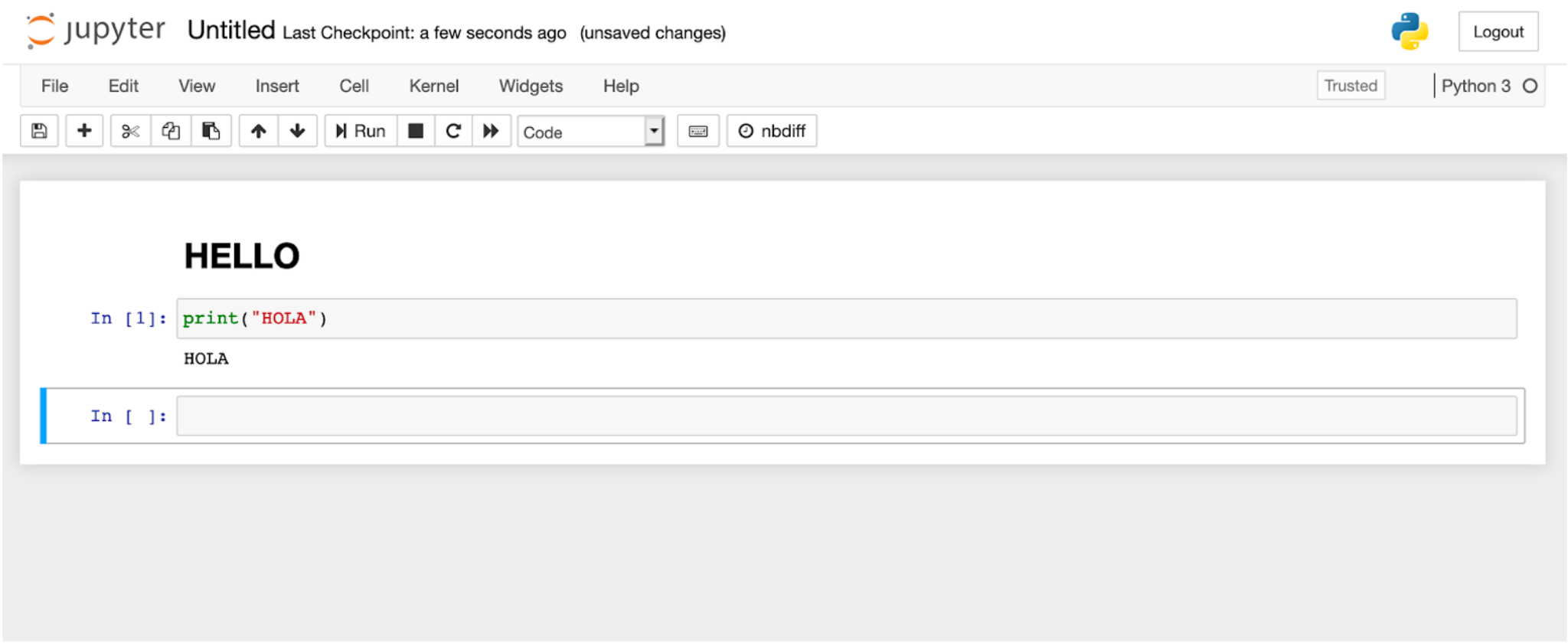
https://jupyter.org/

Algunos usos de Jupyter Notebooks

- Limpieza y transformación de data
- Visualización de data
- Modelamiento estadístico de datos
- Aplicaciones de machine learning y mas..

https://jupyter.org/

Cómo se ve un Jupyter Notebook?



https://jupyter.org/

Cómo podemos instalarlo localmente?

- La opción **Manual**
- Usando **Docker CE**
- Dando click con **Binder**

Implementándolo manualmente:

Prerequisite: Python

Asi Jupyter pueda correr diferentes tipos de lenguage, Python is un requerimiento (Python 3.3+, or Python 2.7) para instalar la aplication de Jupyter Notebooks.

Using Conda : conda install -c conda-forge notebook

Using PIP : pip install notebook

Una vez que Jupyter Notebook este instalado, puedes correr lo siguiente en tu terminal para poder initializar el servidor: **jupyter notebook**

Implementándolo con **Docker CE**:

Prerequisite: Docker CE

Te recommiendo que primero instales docker desktop y la version "Community Edition" . Despues de eso vas a poder bajar y correr una imagen de docker que hemos creamos para este workshop con todo el material que vamos a usar por las siguientes 2 horas.

Tambien existen imagenes de docker "Ready-To-Go":

- Jupyter Docker Stacks: https://github.com/jupyter/docker-stacks
- Jupyter Docker Base Image: https://hub.docker.com/r/jupyter/base-notebook/

docker run -p 8888:8888 jupyter/minimal-notebook:3b1f4f5e6cc1

Implementándolo con **Docker CE**:

docker image pull cyb3rward0g/ekoparty-blue-2020**:0.1**

docker run --rm -it -p 8888:8888 cyb3rward0g/ekoparty-blue-2020**:0.1**

Implementándolo con binder :

- El proyecto Binder es una comunidad abierta que hace posible la creación ambientes interactivos y facil de compartir con la comunidad.
- El producto principal de esta comunidad es el BinderHub. Lo cual es un proyecto que maneja, monitorea y ejecuta ambientes definidos por usarios en la comunidad y guardados en respositorios de binder (Ejemplo: GitHub)
- Para quien: Desarroladores, Profesores, Personas que quieren compartir su research y las que se quieren comunidad por medio de data

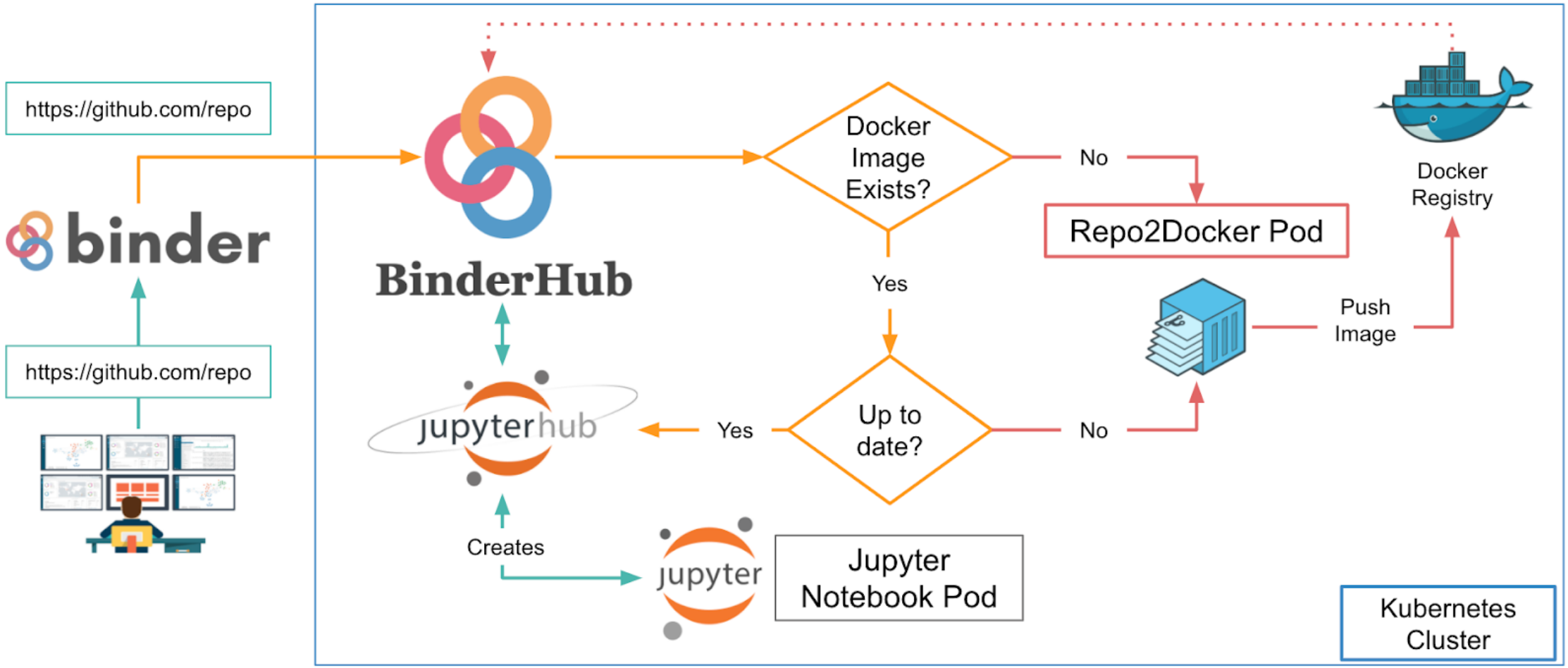
binderHub

BinderHub conecta bastantes servicios para proveer un ambiente en la nube.

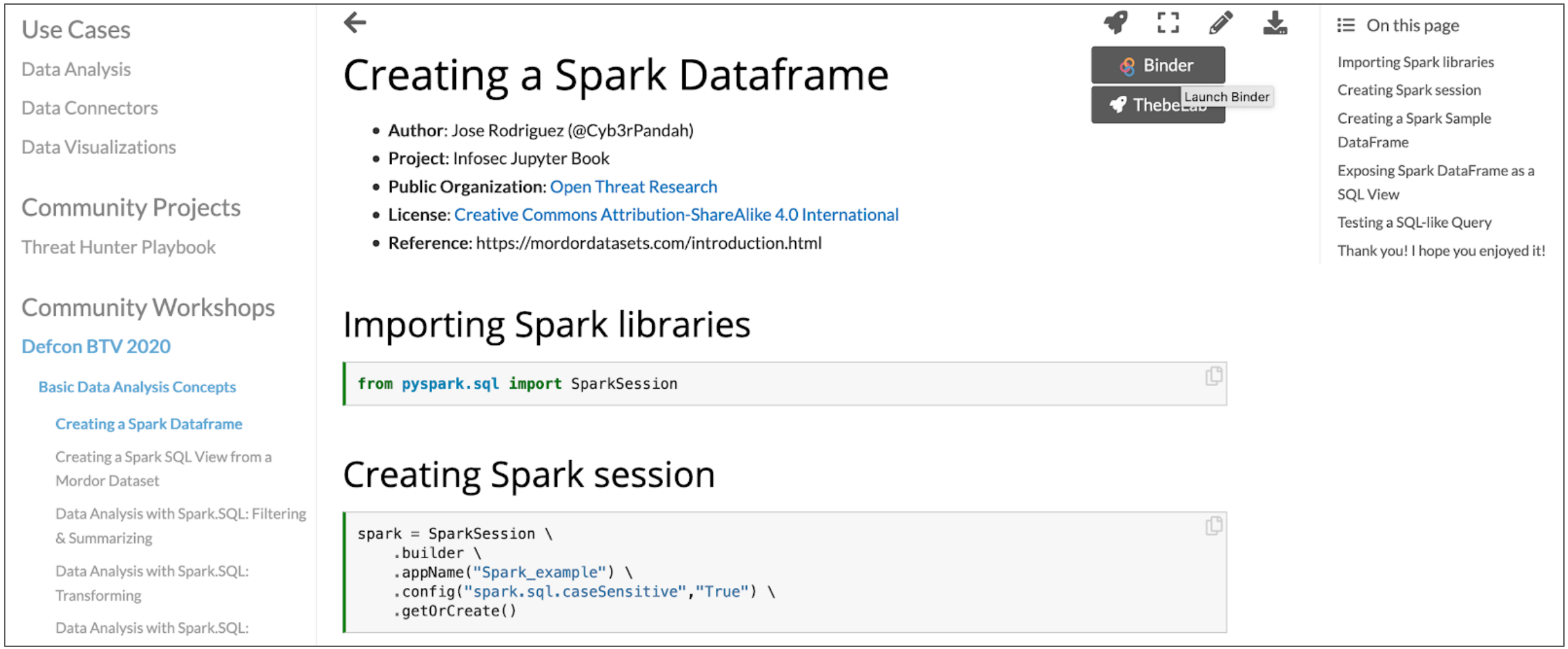
Utiliza los siguientes servicions:

- Un **proveedor de servicios en la nube** como Google Cloud, Microsoft Azure, Amazon EC2, and others
- **Kubernetes** para administrar los recursos en la nube
- **Helm** para configurar y administrar Kubernetes
- **Docker** para usar contenedores que estandarizan ambientes computacionales
- Una **BinderHub UI** que los usuarios pueden accessar para especificar repositorios Git que desean crear
- **BinderHub** para generar imagenes de Docker usando la URL de un repositorio GIT
- **A Docker registry** (por ejemplo gcr.io) que alberga las imagenes de los contenedores
- **JupyterHub** para implementar contenedores temporales para usuarios

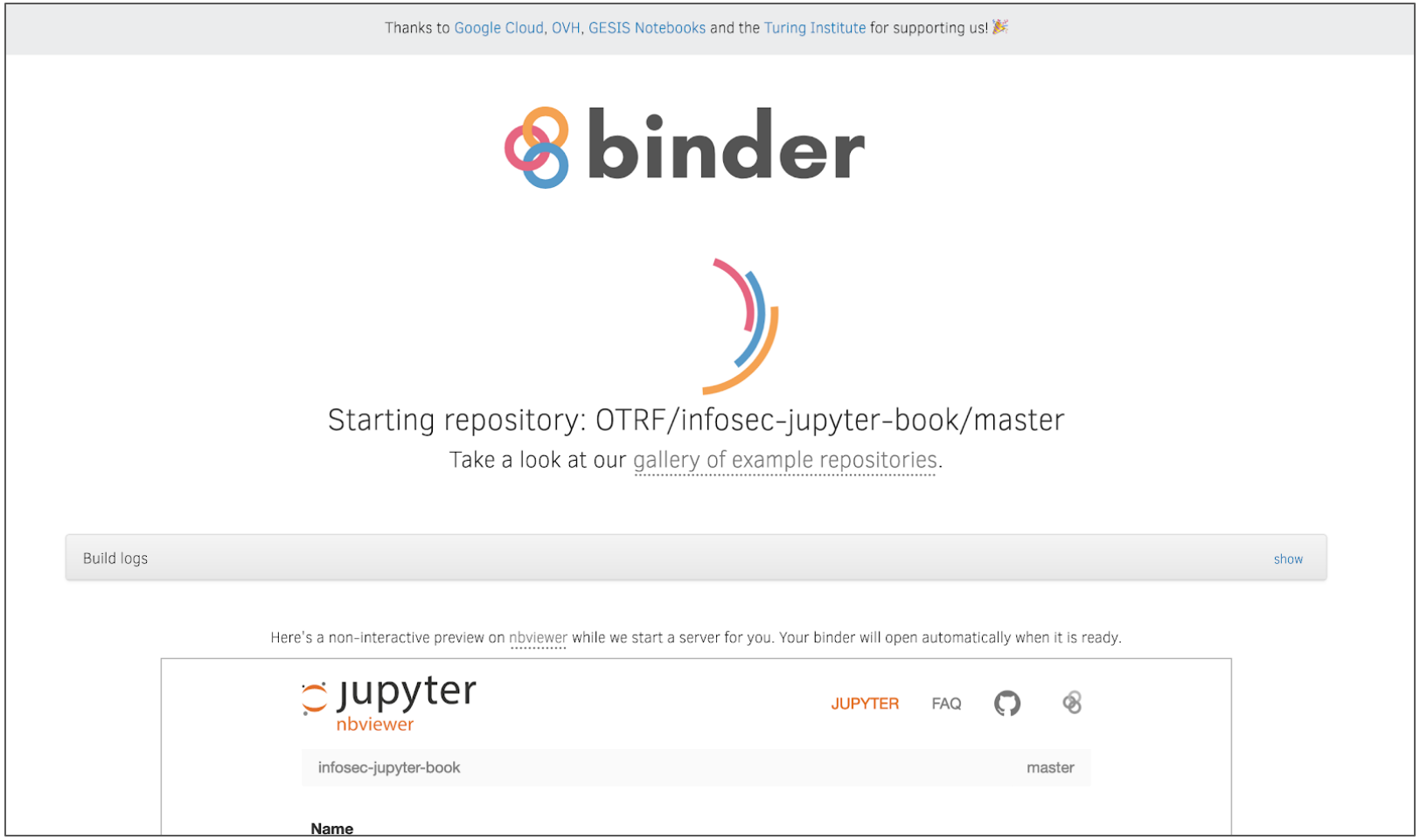
El diseño de binder



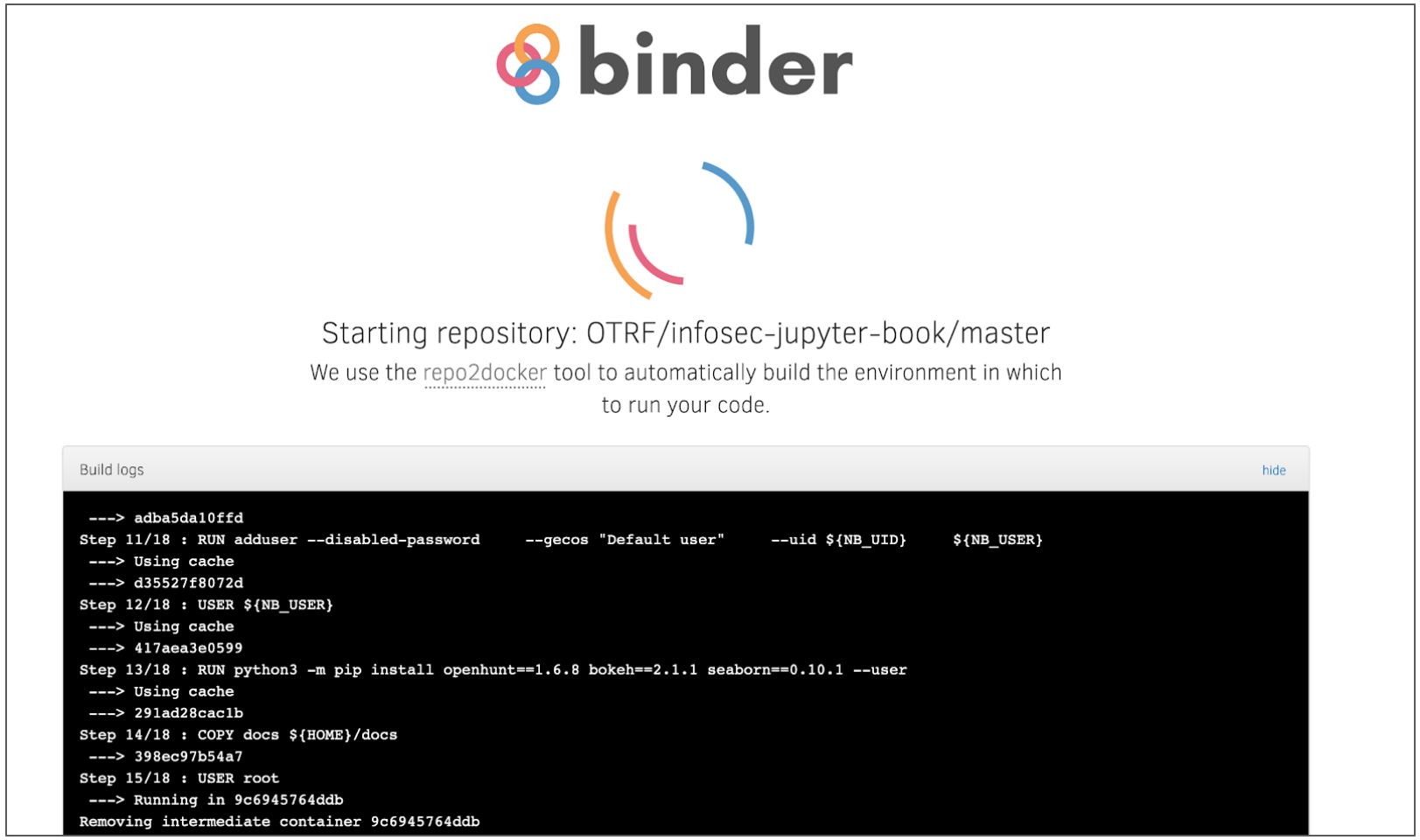
Usando binder



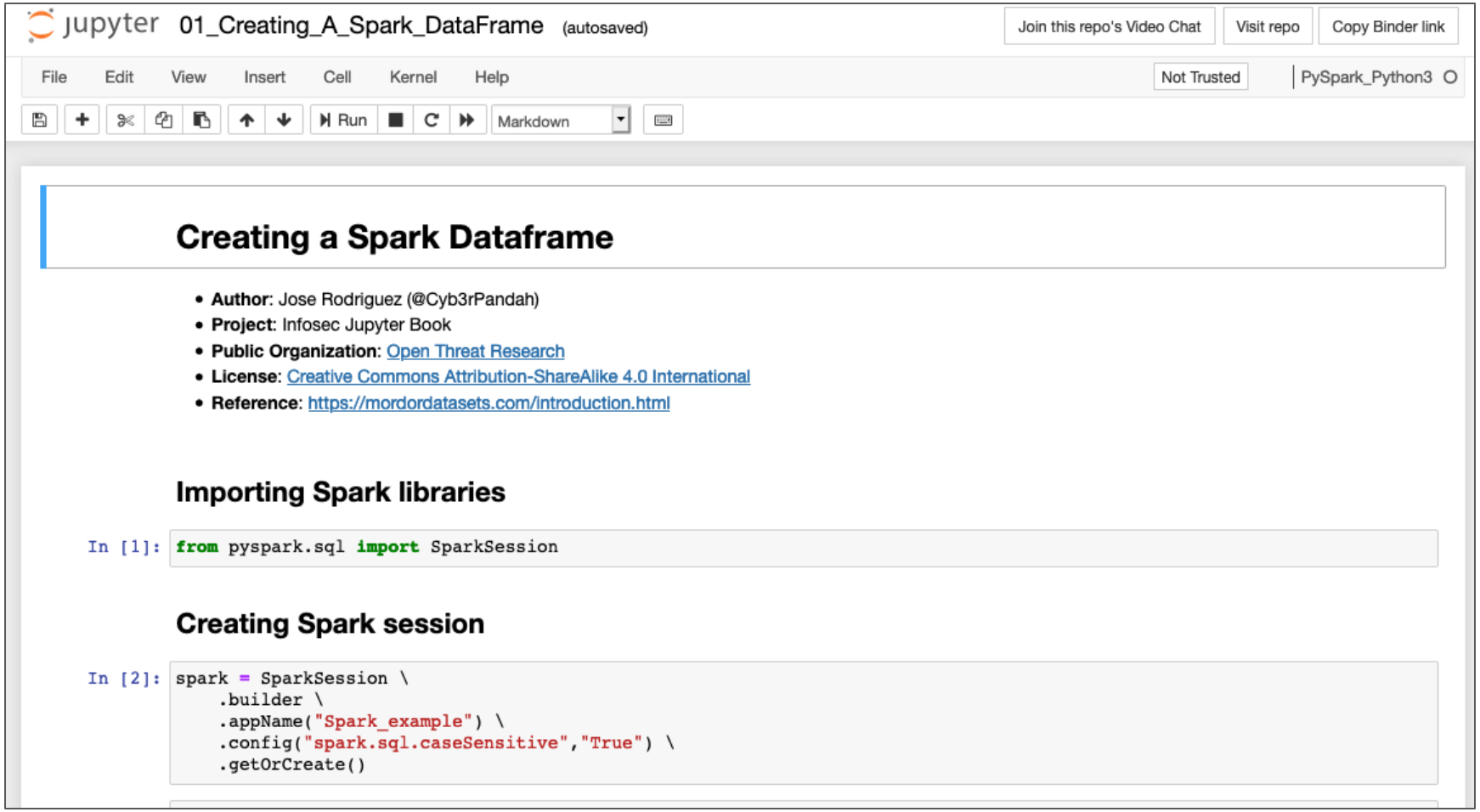
Usando binder



Usando binder



Usando binder



Introducción a Pandas

Una librería del lenguaje Python

- Pandas provee **estructuras de datos** que han sido diseñadas para que el trabajo con data relacionada o categorizada sea de una forma eficiente, fácil e intuitiva.
- A su vez, Pandas depende de la librería **Numpy** y sus conceptos de arrays/arreglos multidimensionales para ejecutar operaciones matemáticas de una manera eficiente.

https://pandas.pydata.org/docs/getting_started/overview.html

Un Array?

- **Estructuras de Python similares a una lista de Python pero limitadas en los tipos de objectos que pueden ser guardados en el mismo array.**

```
import array
```

```
array_one = array.array('i',[1,2,3,4])
type(array_one)
```

```
type(array_one[0])
```

```
int
```

Type code	C Type	Python Type	Minimum size in bytes
'b'	signed char	int	1
'B'	unsigned char	int	1
'u'	Py_UNICODE	Unicode character	2
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int	int	2
'I'	unsigned int	int	2
'l'	signed long	int	4
'L'	unsigned long	int	4
'f'	float	float	4
'd'	double	float	8

https://docs.python.org/3.0/library/array.html

Quieres Seguir Los Demos? (Notebook #1)

https://otrf.github.io/workshop-ekoparty-bluespace-2020/conceptos-basicos/1_intro_numpy_arrays.html

NumPy N-Dimensional Array (ndarray)?

- El ndarray es un contenedor multidimensional de objetos del mismo tipo de data.
- Extiende el concepto de ejecucion de funciones matematicas en un array y permite la ejecución de tasks complejas en un array entero sin crear un Python For loop (Operaciones vectoriales eficientes).
- Numpy arrays, a su vez, usa menos memoria que una lista

```
import numpy as np
np.__version__
```

```
'1.19.2'
```

```
list_one = [1,2,3,4,5]
```

```
numpy_array = np.array(list_one)
type(numpy_array)
```

```
numpy.ndarray
```

```
numpy_array
```

```
array([1, 2, 3, 4, 5])
```

https://pandas.pydata.org/docs/getting_started/overview.html

NumPy N-Dimensional Array (ndarray)?

- El ndarray es un contenedor multidimensional de objetos del mismo tipo de data.
- Extiende el concepto de ejecucion de funciones matematicas en un array y permite la ejecución de tasks complejas en un array entero sin crear un Python For loop (Operaciones vectoriales eficientes).
- Numpy arrays, a su vez, usa menos memoria que una lista

```
list_two = [1,2,3,4,5]
# The following will throw an error:
list_two + 2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most
<ipython-input-8-03923fe34c76> in <module>
      1 list_two = [1,2,3,4,5]
      2 # The following will throw an error:
----> 3 list_two + 2

TypeError: can only concatenate list (not "int") to list
```

- Performing a loop to add **2** to every integer in the list

https://pandas.pydata.org/docs/getting_started/overview.html

NumPy N-Dimensional Array (ndarray)?

- El ndarray es un contenedor multidimensional de objetos del mismo tipo de data.
- Extiende el concepto de ejecucion de funciones matematicas en un array y permite la ejecución de tasks complejas en un array entero sin crear un Python For loop (Operaciones vectoriales eficientes).
- Numpy arrays, a su vez, usa menos memoria que una lista

```
for index, item in enumerate(list_two):
    list_two[index] = item + 2
list_two
```

```
[3, 4, 5, 6, 7]
```

https://pandas.pydata.org/docs/getting_started/overview.html

NumPy N-Dimensional Array (ndarray)?

- El ndarray es un contenedor multidimensional de objetos del mismo tipo de data.
- Extiende el concepto de ejecucion de funciones matematicas en un array y permite la ejecución de tasks complejas en un array entero sin crear un Python For loop (Operaciones vectoriales eficientes).
- Numpy arrays, a su vez, usa menos memoria que una lista

```
numpy_array
```

```
array([1, 2, 3, 4, 5])
```

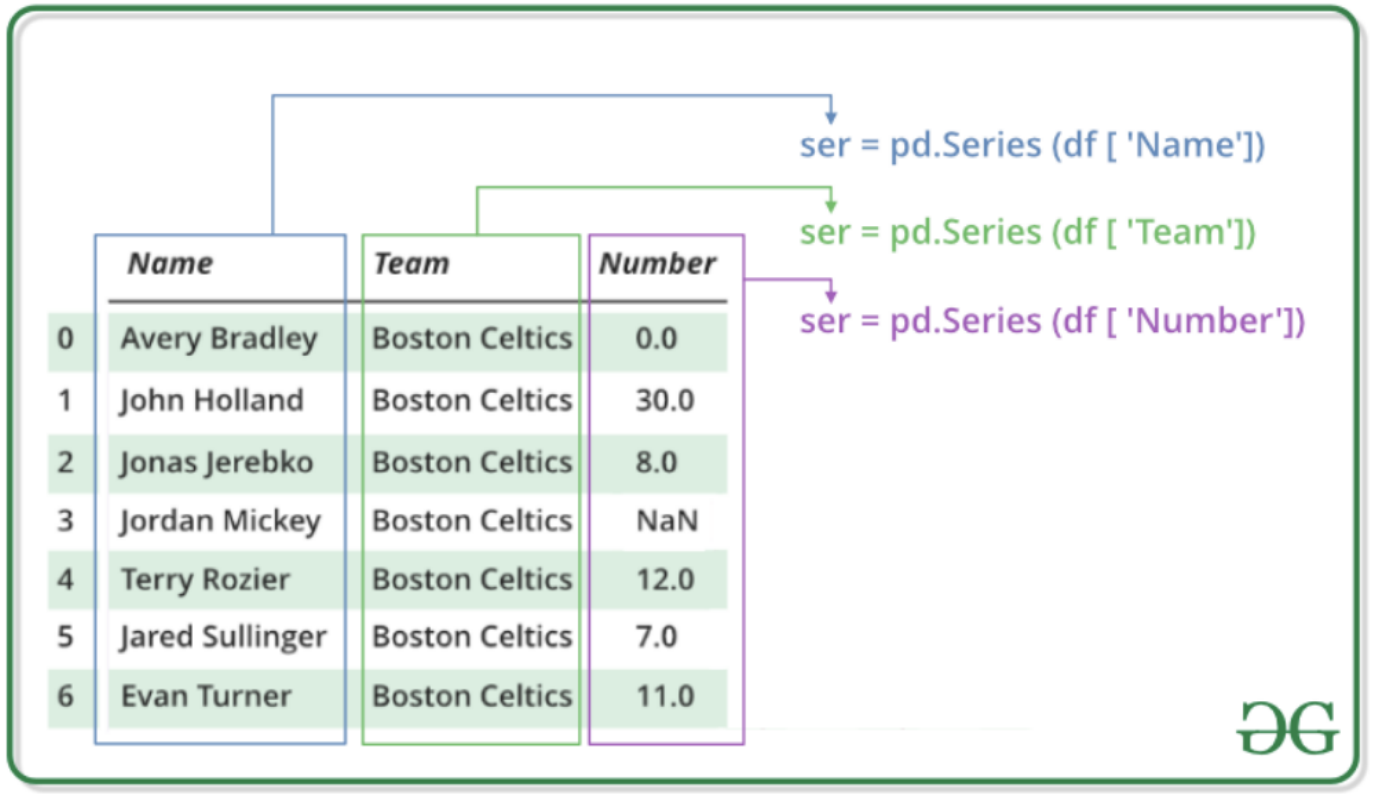
```
numpy_array + 2
```

```
array([3, 4, 5, 6, 7])
```

https://pandas.pydata.org/docs/getting_started/overview.html

Estructuras de data de Pandas

- Las dos principales estructuras de datos que utiliza Pandas son **Series** (1 dimensión) y **Dataframe** (2 dimensiones).
- Estas estructuras permiten a un usuario gestionar data en diversos casos de uso.

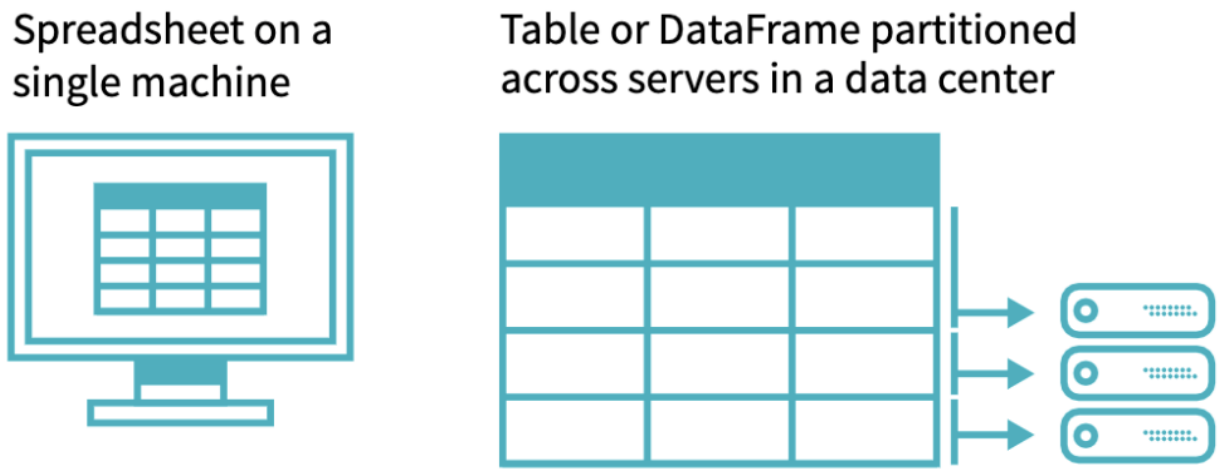


	Name	Team	Number
0	Avery Bradley	Boston Celtics	0.0
1	John Holland	Boston Celtics	30.0
2	Jonas Jerebko	Boston Celtics	8.0
3	Jordan Mickey	Boston Celtics	NaN
4	Terry Rozier	Boston Celtics	12.0
5	Jared Sullinger	Boston Celtics	7.0
6	Evan Turner	Boston Celtics	11.0

https://pandas.pydata.org/docs/getting_started/overview.html

Que es una **Dataframe**?

Un dataframe es una estructura de datos con dos dimensiones que se encuentra categorizada por columnas que podrían incluir diferentes tipos de datos.



Que podemos hacer con **Pandas**?

- Gestionar data faltante
- Agregar data (Group By)
- Segmentar, indexar y categorizar data
- Representar data como dataframe otras estructuras de datos
- Correlacionar data (Join, Merge)
- Análisis de series de tiempo
- Crear visualizaciones
- Y mucho más...

https://pandas.pydata.org/docs/getting_started/overview.html

Cómo instalar **Pandas**?

- A través de Anaconda
- A través de Miniconda
  - conda install pandas
  - conda install pandas=0.20.3
- Instalando desde PyPi (Python Package Index)
  - pip install pandas

https://pandas.pydata.org/docs/getting_started/install.html

El Proceso de Análisis de Datos

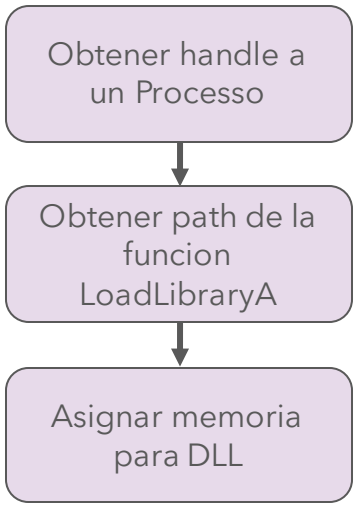
El proceso de Análisis de Datos

El proceso de Análisis de Datos

Definir un Objetivo → Identificar Data Relevante → Simular al Adversario → Analizar Datos → Interpretar Resultados

MITRE | ATT&CK

Strategia del Adversario



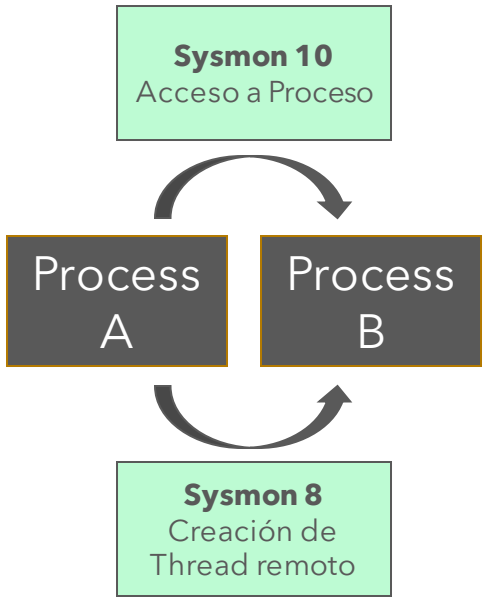
El proceso de Análisis de Datos

Definir un Objetivo → Identificar Data Relevante → Simular al Adversario → Analizar Datos → Interpretar Resultados

MITRE | ATT&CK®

Strategia del Adversario

- Obtener handle a un Processo
- Obtener path de la funcion LoadLibraryA
- Asignar memoria para DLL



El proceso de Análisis de Datos

Definir un Objetivo → Identificar Data Relevante → Simular al Adversario → Analizar Datos → Interpretar Resultados

MITRE | ATT&CK

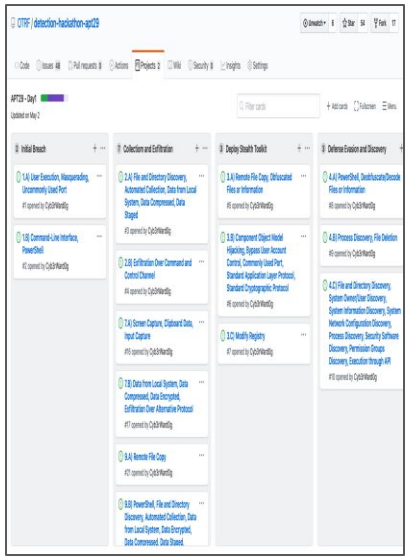
Strategia del Adversario

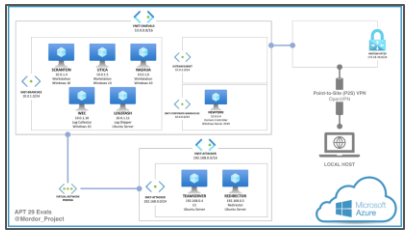
- Obtener handle a un Processo
- Obtener path de la funcion LoadLibraryA
- Asignar memoria para DLL

Sysmon 10
Acceso a Proceso

Process A → Process B

Sysmon 8
Creación de Thread remoto





El proceso de Análisis de Datos

Definir un Objetivo → Identificar Data Relevante → Simular al Adversario → Analizar Datos → Interpretar Resultados

MITRE | ATT&CK

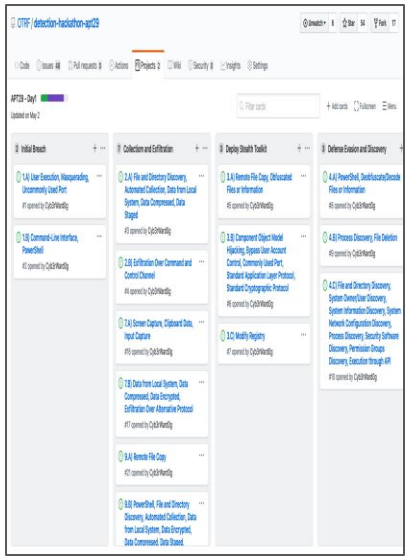
Strategia del Adversario

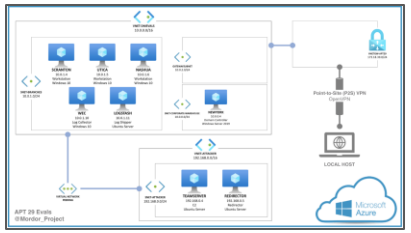
- Obtener handle a un Processo
- Obtener path de la funcion LoadLibraryA
- Asignar memoria para DLL

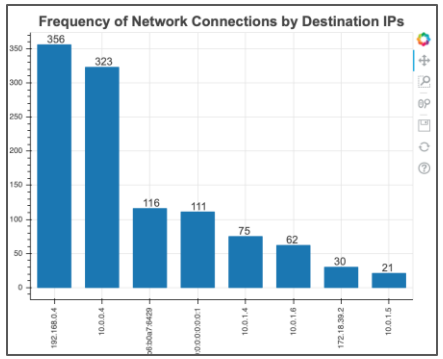
Sysmon 10
Acceso a Proceso

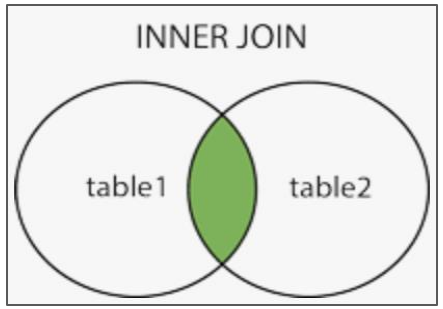
Process A → Process B

Sysmon 8
Creación de Thread remoto









El proceso de Análisis de Datos

Definir un Objetivo → Identificar Data Relevante → Simular al Adversario → Analizar Datos → Interpretar Resultados

MITRE | ATT&CK

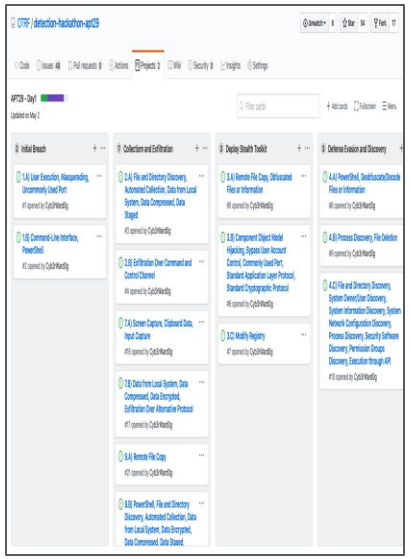
Strategia del Adversario

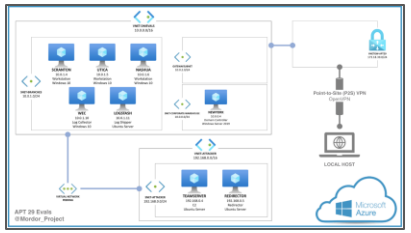
- Obtener handle a un Processo
- Obtener path de la funcion LoadLibraryA
- Asignar memoria para DLL

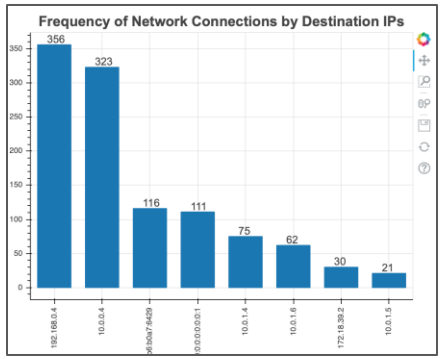
Sysmon 10
Acceso a Proceso

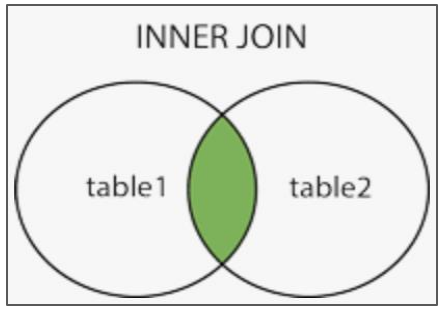
Process A → Process B

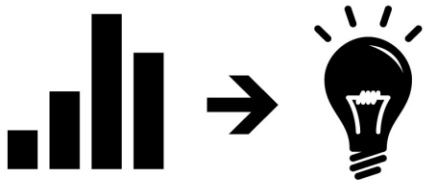
Sysmon 8
Creación de Thread remoto











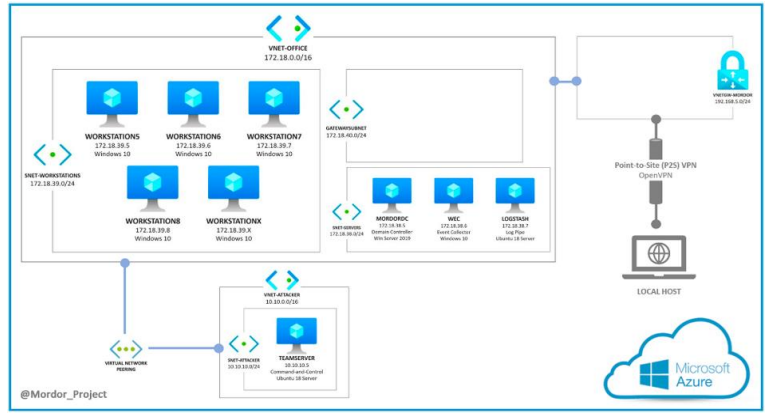
Ideas para comenzar a definir estrategias de detección

Necesitamos Data?...

MORDOR 😈

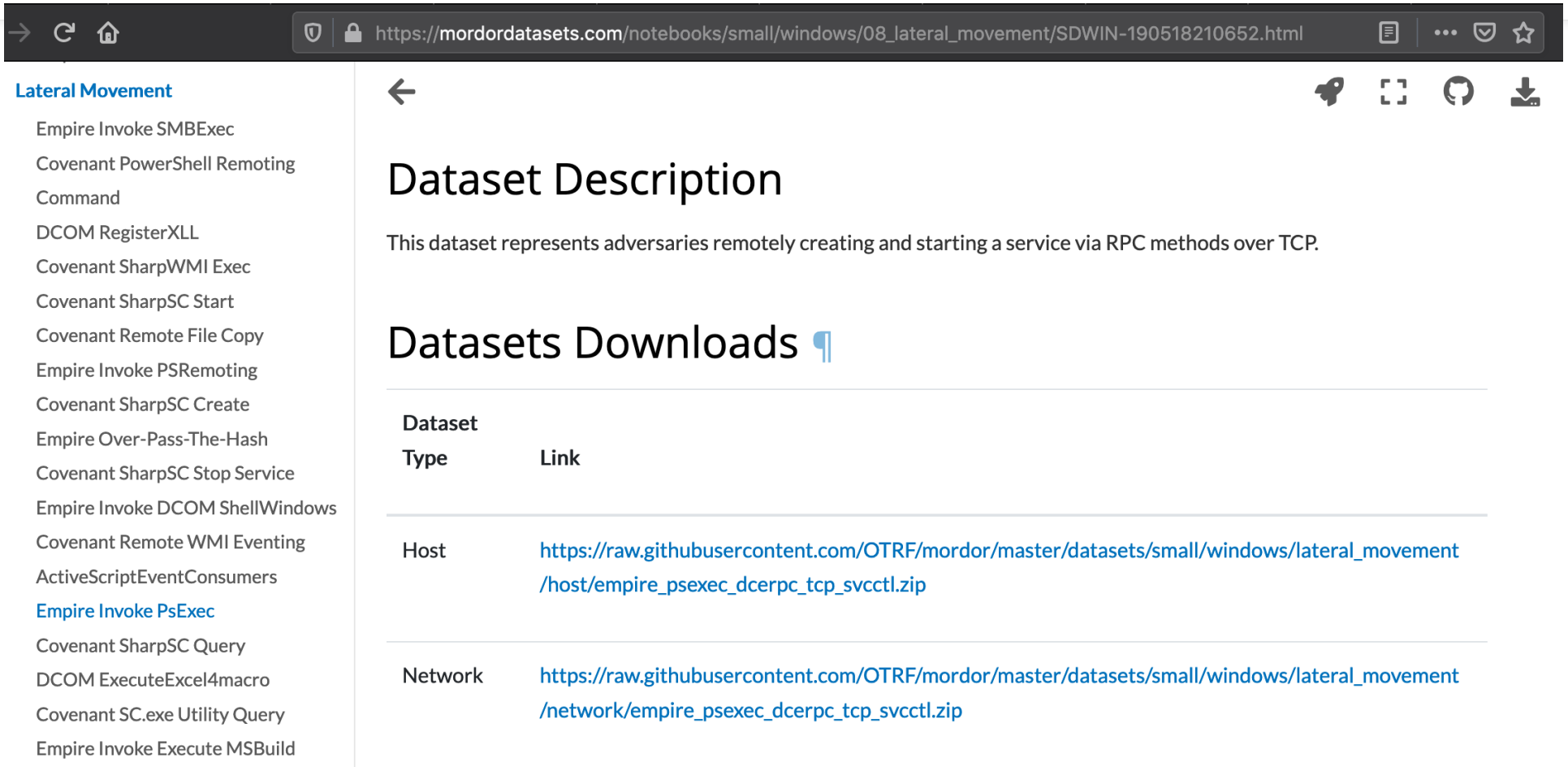
@Mordor_Project

- Eventos de seguridad pre-grabados, generados a través de la simulación de técnicas usadas por adversarios en
- Formato JavaScript Object Notation (**JSON**)
- Sets de datos categorizados por plataformas, grupos de adversarios, tácticas y técnicas definidas por MITRE - ATT&CK
- Datasets pequeñas y grandes



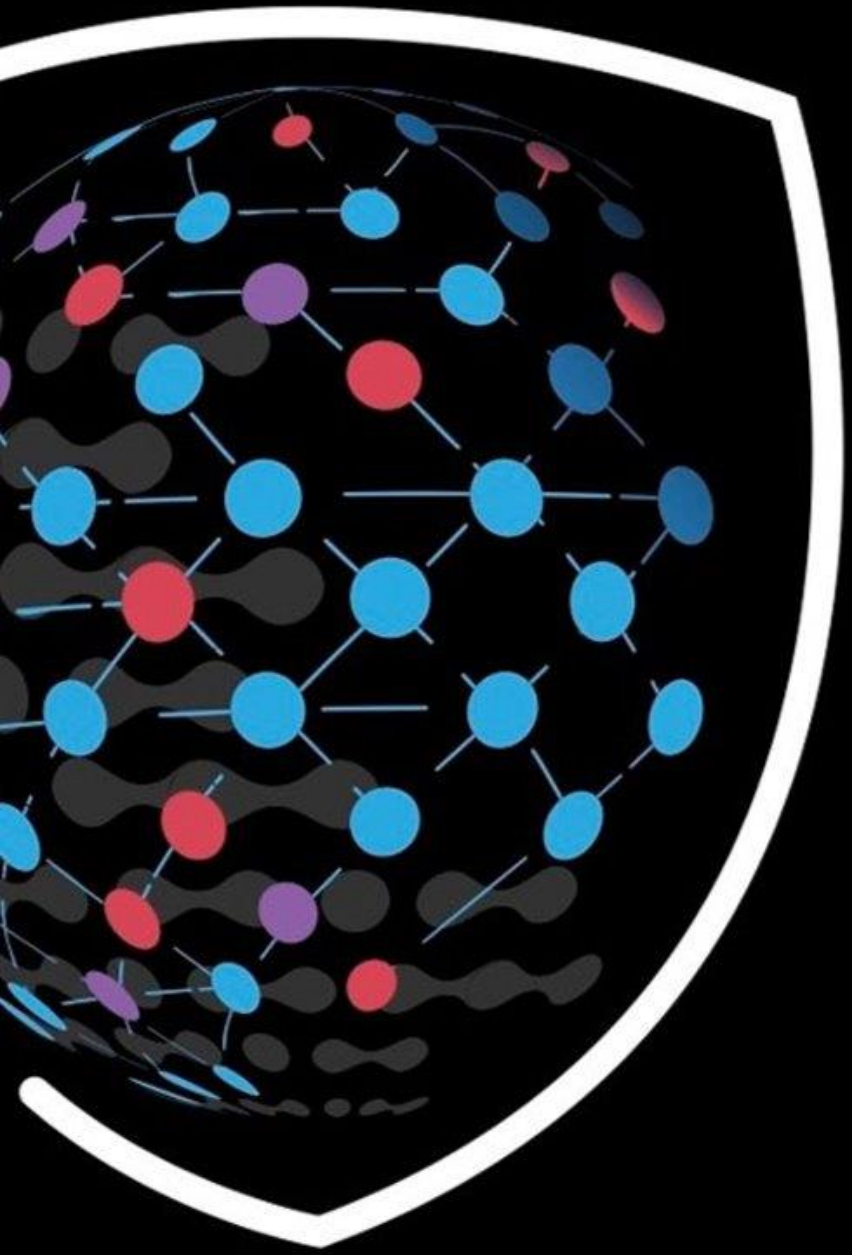
https://mordordatasets.com

@Mordor_Project



https://mordordatasets.com/notebooks/small/windows/08_lateral_movement/SDWIN-190518210652.html

Ejemplo 1: Importando sets de datos



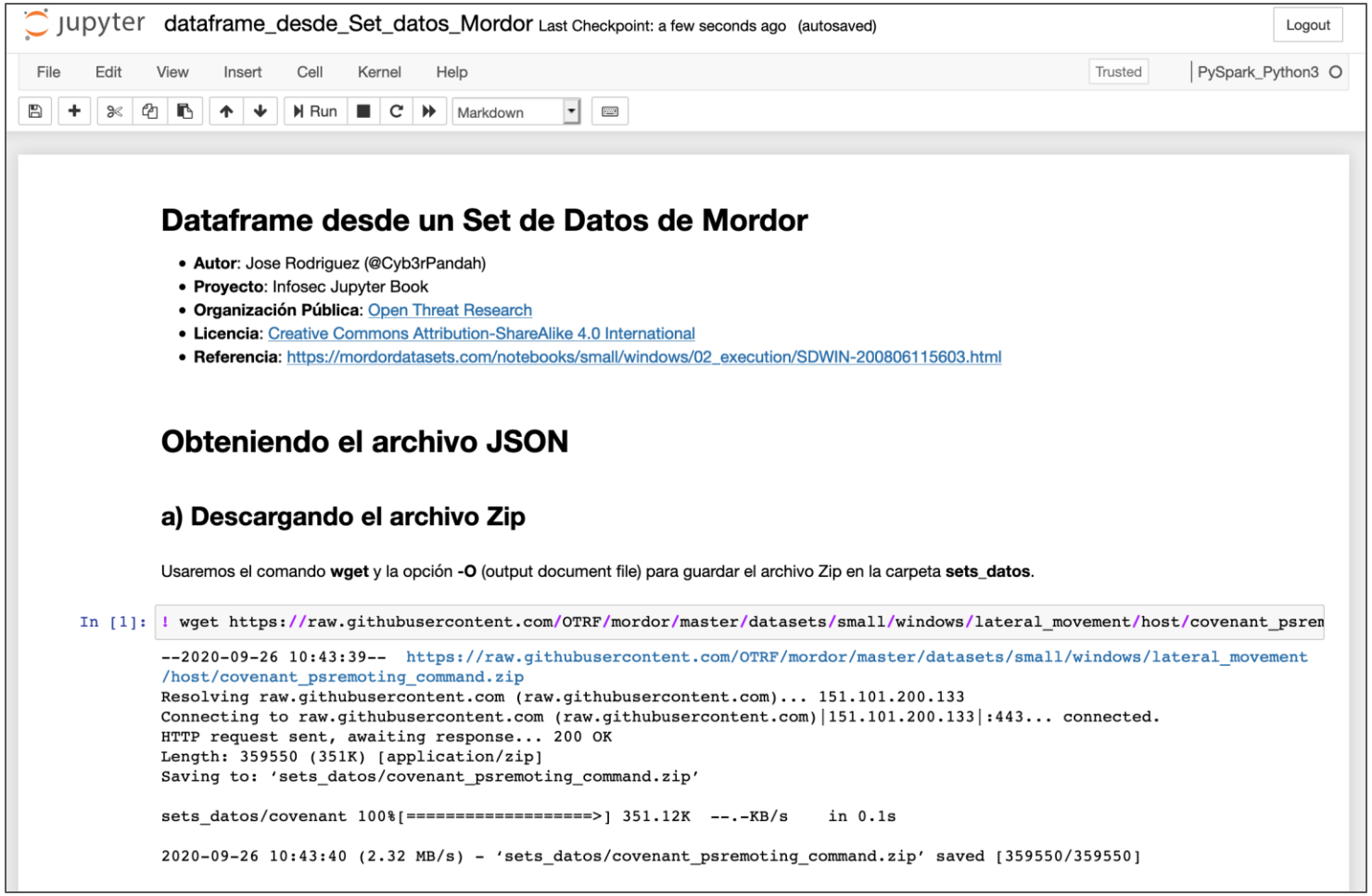
OPEN THREAT RESEARCH

EMPOWERING THE INFOSEC COMMUNITY

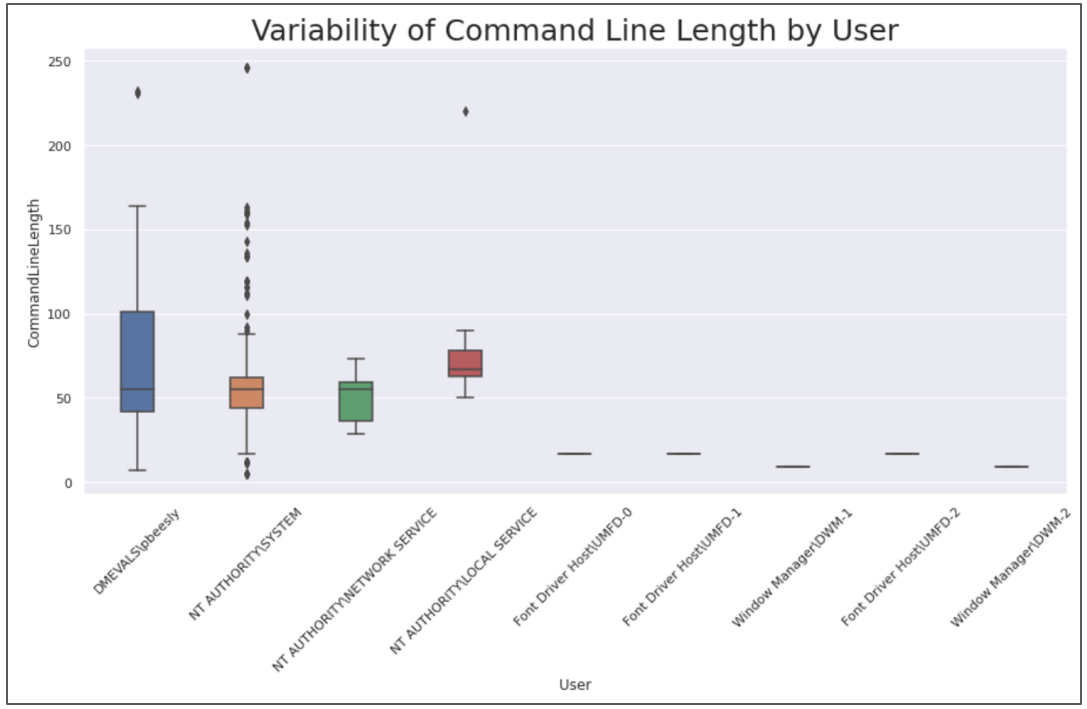
Quieres Seguir Los Demos? (Notebook #2)

https://otrf.github.io/workshop-ekoparty-bluespace-2020/conceptos-basicos/2_dataframe_desde_Set_datos_Mordor.html

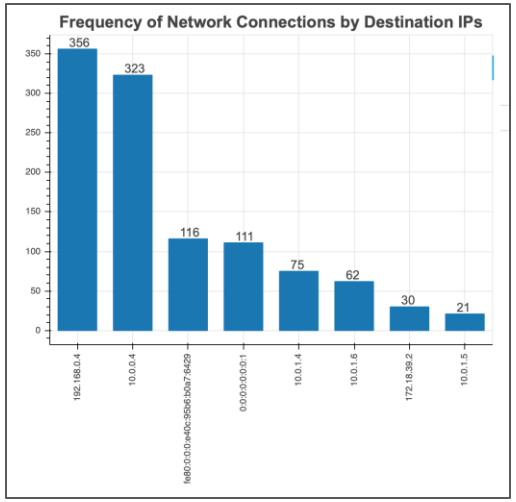
Importando sets de datos desde Mordor



Técnicas para Análisis de Datos

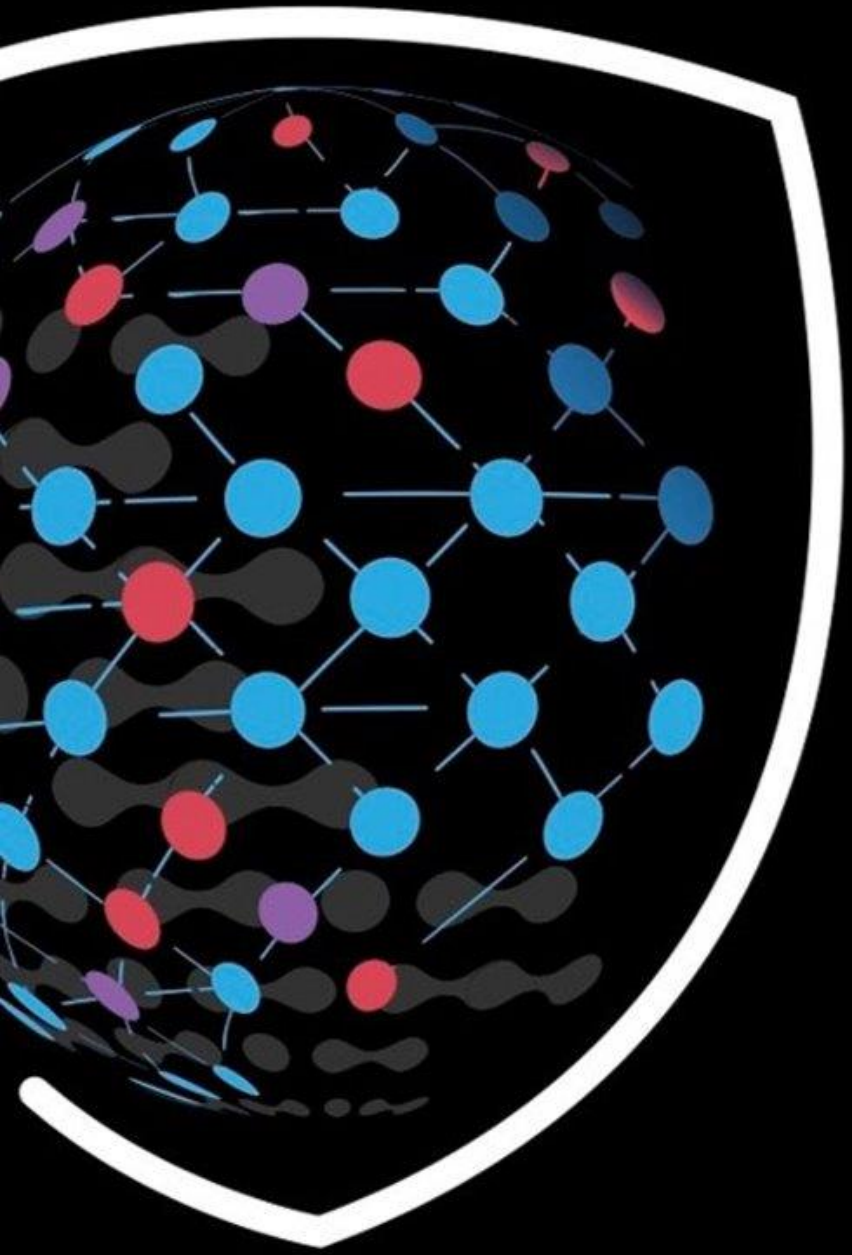


GrantedAccess	Count
0x1000	463
0x3000	83
0x40	4
0x1fffff	2
0x1400	2
0x1410	2
0x1478	2
0x1f3fff	1
0x100000	1
0x101541	1



Algunas técnicas basicas son:

- Resumir data
- Filtrar columnas y/o filas
- Transformar data
- Correlacionar data
- Visualizar data



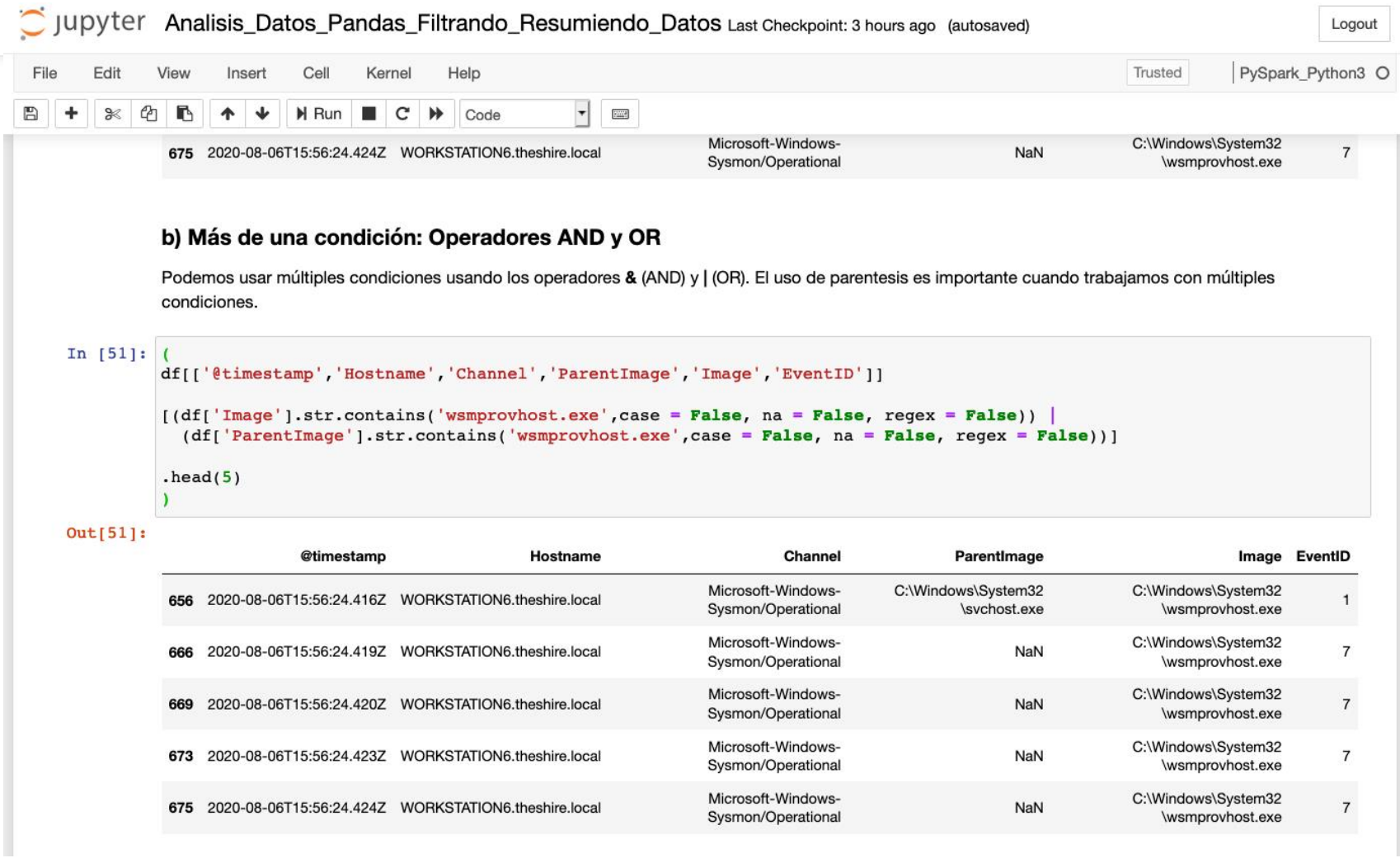
OPEN THREAT RESEARCH

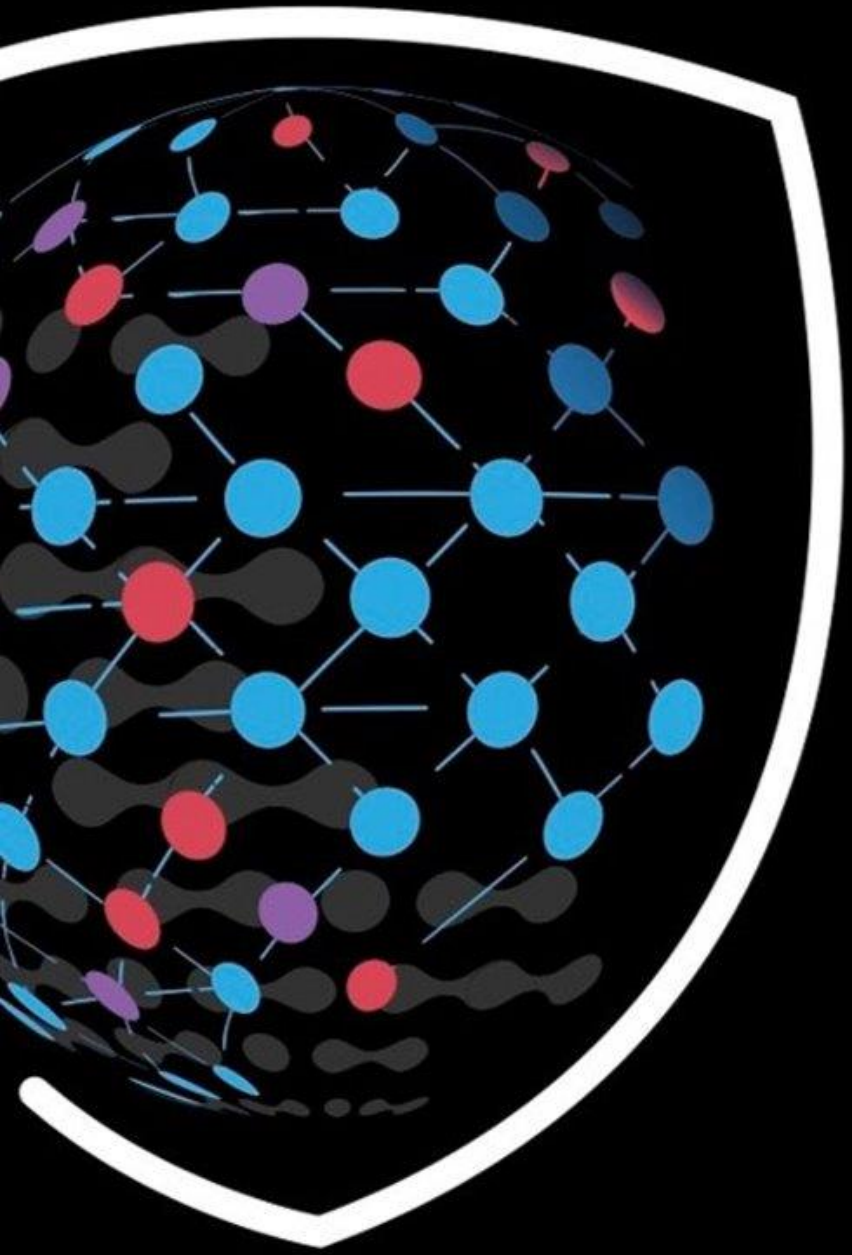
EMPOWERING THE INFOSEC COMMUNITY

Quieres Seguir Los Demos? (Notebook #3)

https://otrf.github.io/workshop-ekoparty-bluespace-2020/conceptos-basicos/3_Analisis_Datos_Pandas_Filtrando_Resumiendo_Datos.html

Resumiendo data con Pandas





OPEN THREAT RESEARCH

EMPOWERING THE INFOSEC COMMUNITY

Quieres Seguir Los Demos? (Notebook #4)

https://otrf.github.io/workshop-ekoparty-bluespace-2020/conceptos-basicos/4_Analisis_Datos_Pandas_Transformando_Datos.html

Transformando Datos

Calculando la Longitud del CommandLine

Usaremos el método **assign** para agregar una columna nueva a nuestro dataframe. Esta nueva columna mostrará el calculo de la longitud del command line que el processo utilizó.

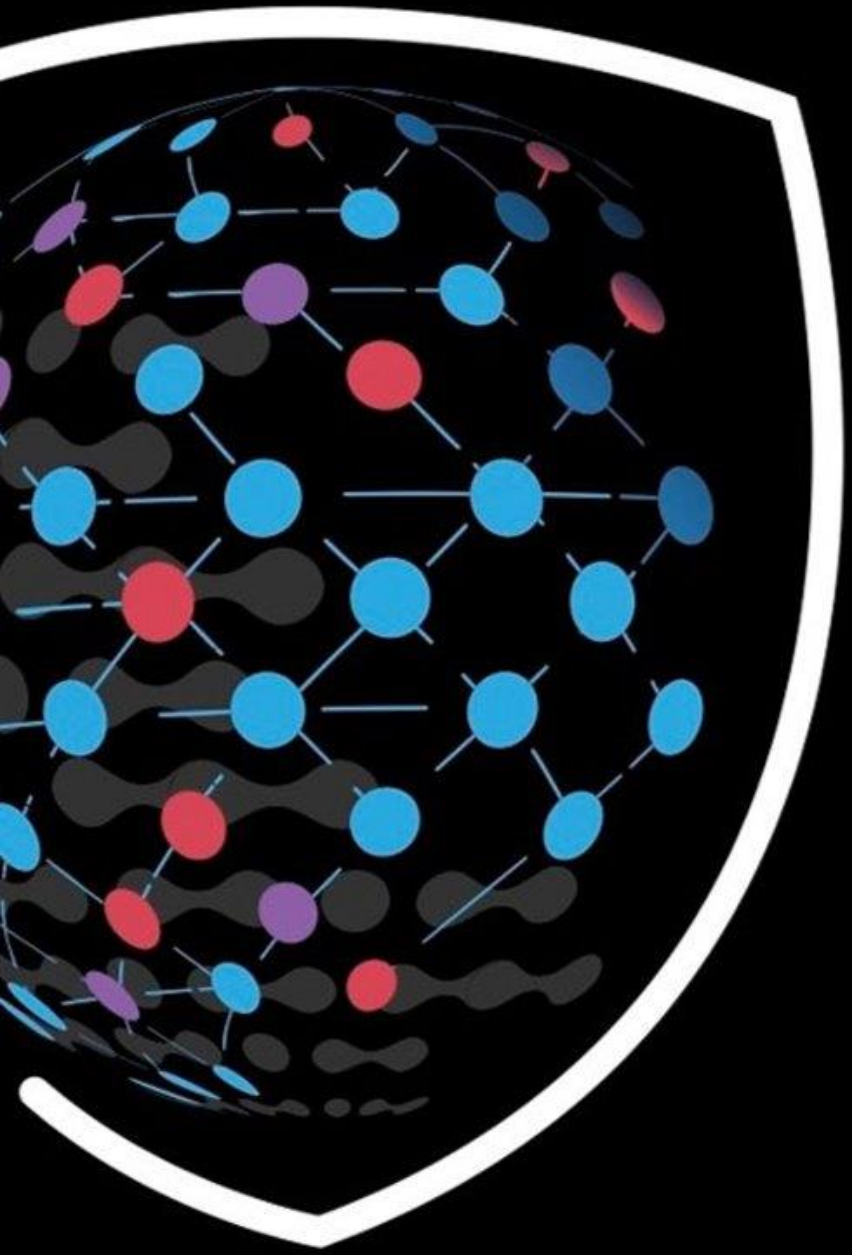
Referencia: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.assign.html

In [6]:
```
(
df[['@timestamp','Image','CommandLine']]

[(df['EventID'] == 1) & (df['Channel'].str.contains('sysmon',case = False, na = False, regex = False))]

.assign(Command_Length = df['CommandLine'].str.len())
)
```

Out[6]:

	@timestamp	Image	CommandLine	Command_Length
372	2020-05-02T02:55:57.730Z	C:\ProgramData\victim\â€®cod.3aka3.scr	"C:\ProgramData\victim\â€®cod.3aka3.scr" /S	43.0
621	2020-05-02T02:56:05.822Z	C:\Windows\System32\conhost.exe	\\?\C:\windows\system32\conhost.exe --headless...	99.0
649	2020-05-02T02:56:05.830Z	C:\Windows\System32\cmd.exe	"C:\windows\system32\cmd.exe"	29.0
827	2020-05-02T02:56:15.884Z	C:\Windows\System32\WindowsPowerShell\v1.0\pow...	powershell	10.0
3620	2020-05-02T02:57:02.831Z	C:\Windows\System32\SearchProtocolHost.exe	"C:\windows\system32\SearchProtocolHost.exe" G...	308.0
...	...	...	...	...
193780	2020-05-02T03:25:33.847Z	C:\Windows\System32\UsoClient.exe	C:\windows\system32\usoclient.exe StartScan	43.0
193812	2020-05-02T03:25:33.858Z	C:\Windows\System32\usocoreworker.exe	C:\Windows\System32\usocoreworker.exe -Embedding	48.0
194036	2020-05-02T03:25:50.013Z	C:\Windows\System32\taskhostw.exe	taskhostw.exe Logon	19.0
194568	2020-05-02T03:26:12.287Z	C:\Windows\System32\UsoClient.exe	C:\windows\system32\usoclient.exe StartScan	43.0
194596	2020-05-02T03:26:12.298Z	C:\Windows\System32\usocoreworker.exe	C:\Windows\System32\usocoreworker.exe -Embedding	48.0

447 rows × 4 columns



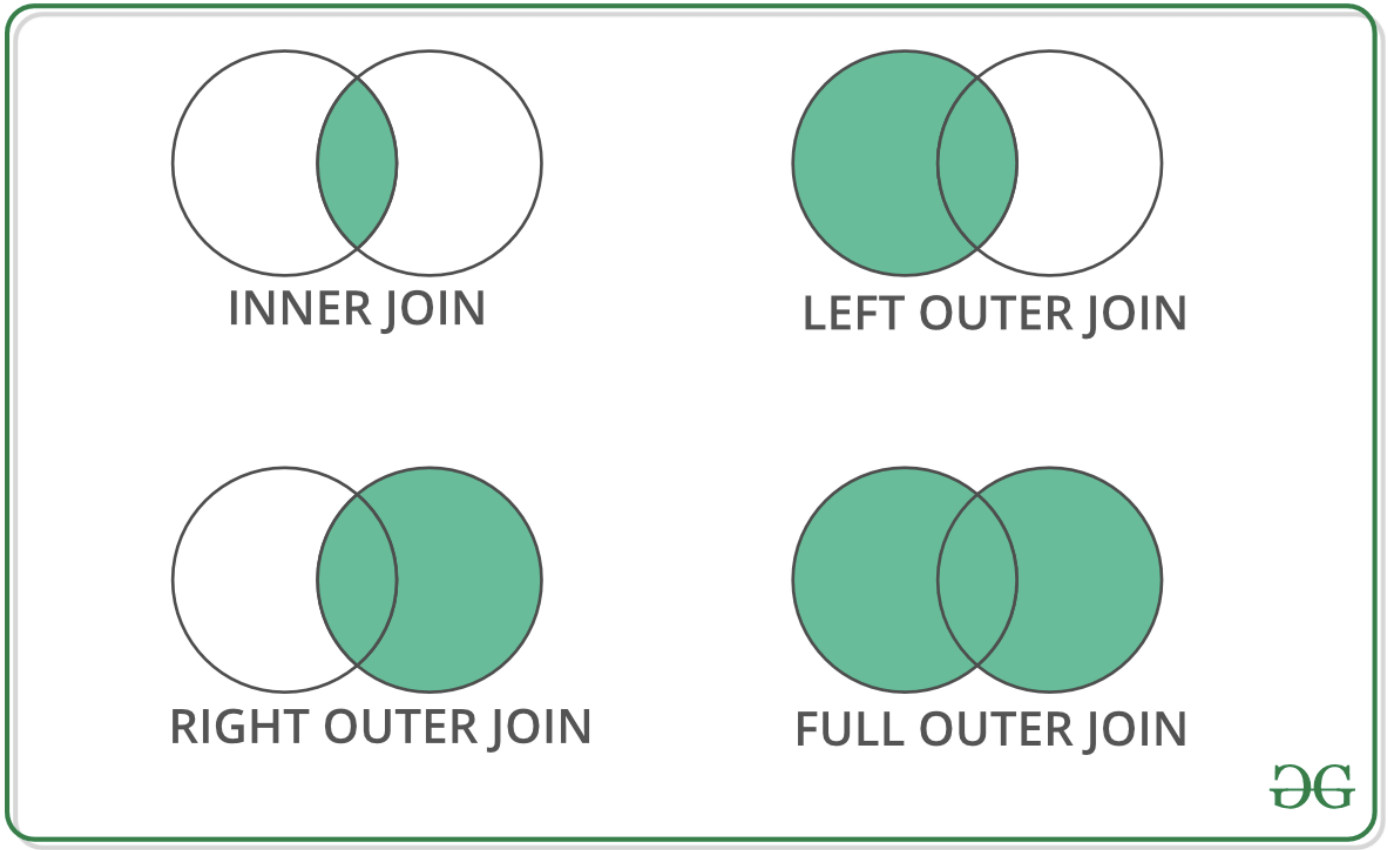
OPEN THREAT RESEARCH

EMPOWERING THE INFOSEC COMMUNITY

Quieres Seguir Los Demos? (Notebook #5)

https://otrf.github.io/workshop-ekoparty-bluespace-2020/conceptos-basicos/5_Analisis_Datos_Pandas_Correlacionando_Datos.html

Correlacionando Data con Pandas (JOINs)



JOIN

- Allows you to combine rows from the same or different tables
- JOINs can be performed with **join()** or **merge()** with the following options **(LEFT, RIGHT, INNER, FULL)** and the columns to join on **(column names or indices)**.

https://www.geeksforgeeks.org/python-pandas-merging-joining-and-concatenating/

Preparando dataframes para JOIN

```
# Creating a dataframe with information of Security event 4624: An account was successfully logged on
Security4624 = (
    apt29[(apt29['Channel'].str.lower() == 'security') & (apt29['EventID'] == 4624)].dropna(axis = 1, how = 'all')
)
Security4624.shape
```

```
(297, 55)
```

```
# Creating a dataframe with information of Security event 4688: A new process has been created
Security4688 = (
    apt29[(apt29['Channel'].str.lower() == 'security') & (apt29['EventID'] == 4688)].dropna(axis = 1, how = 'all')
)
Security4688.shape
```

```
(460, 42)
```

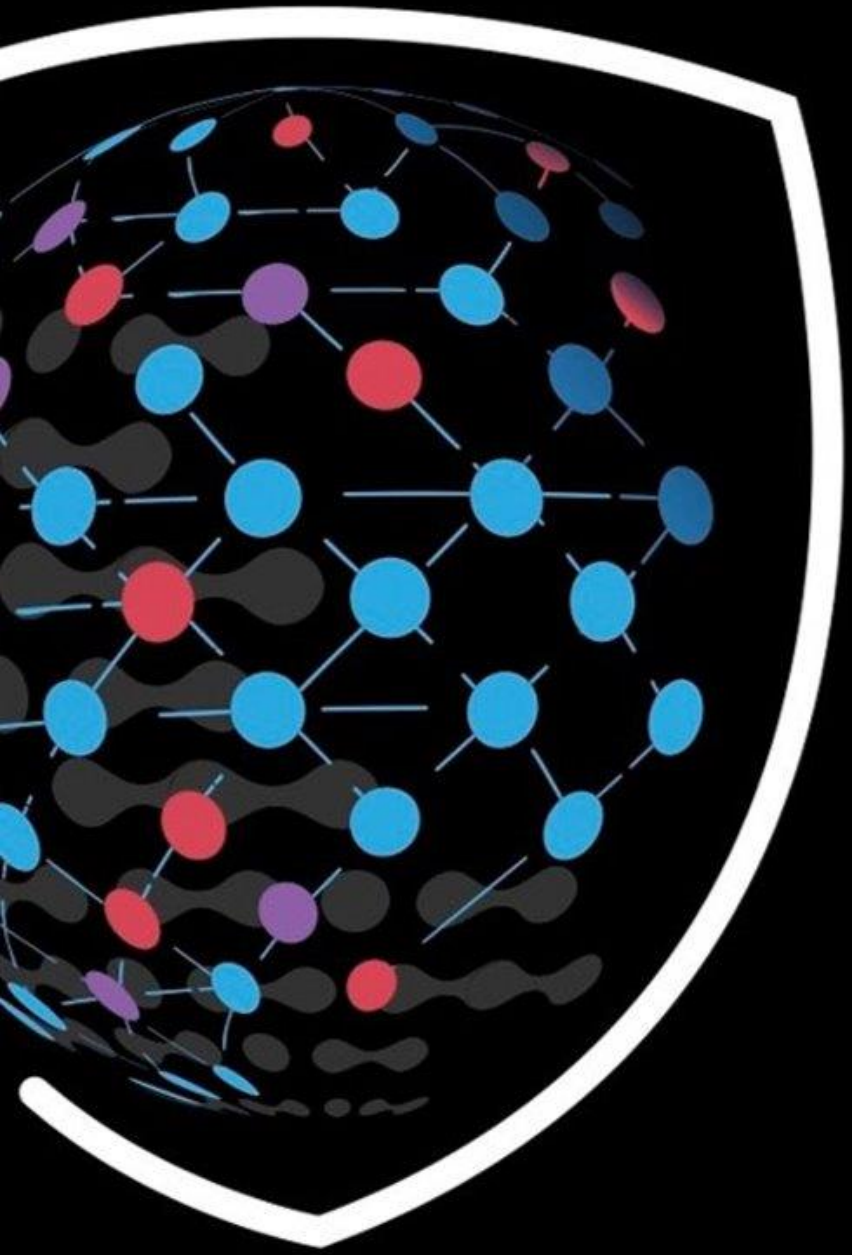
```
# Creating a dataframe with information of Security event 4697: A service was installed in the system
Security4697 = apt29[
    (apt29['Channel'].str.lower() == 'security') & (apt29['EventID'] == 4697)
].dropna(axis = 1, how = 'all')
Security4697.shape
```

```
(23, 37)
```

Procesos Siendo Ejecutados en el contexto de un logon de network (Tipo 3)

```
# merge performs an INNER JOIN by default
(
pd.merge(Security4688, Security4624[Security4624['LogonType'] == 3],
         on = 'TargetLogonId', how = 'inner')
[['NewProcessId','NewProcessName','ProcessId_x','ParentProcessName','TargetUserName_y','IpAddress']]
)
```

	NewProcessId	NewProcessName	ProcessId_x	ParentProcessName	TargetUserName_y	IpAddress
0	0x1e28	C:\Windows\System32\wsmprovhost.exe	0x374	C:\Windows\System32\svchost.exe	pbeesly	-



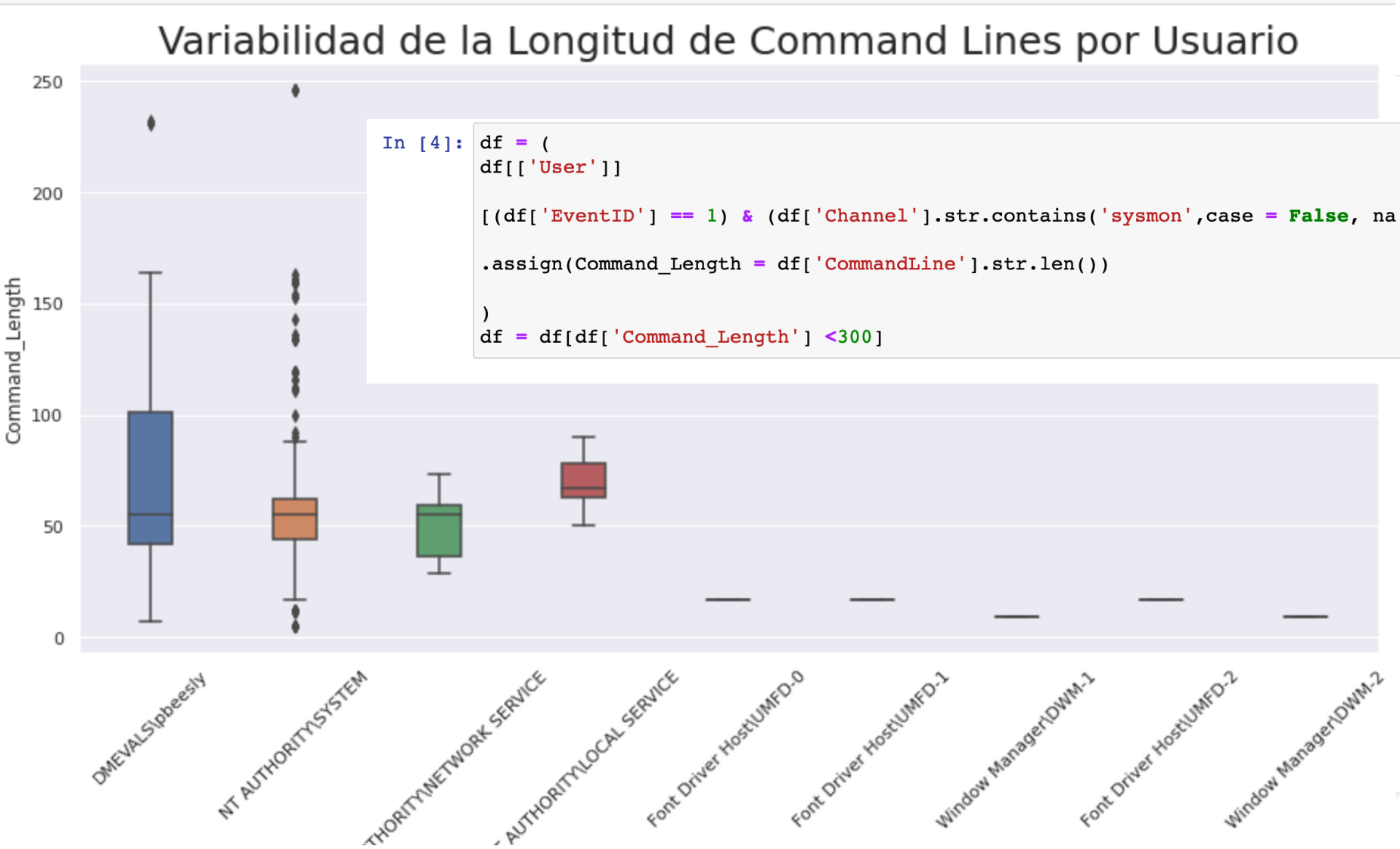
OPEN THREAT RESEARCH

EMPOWERING THE INFOSEC COMMUNITY

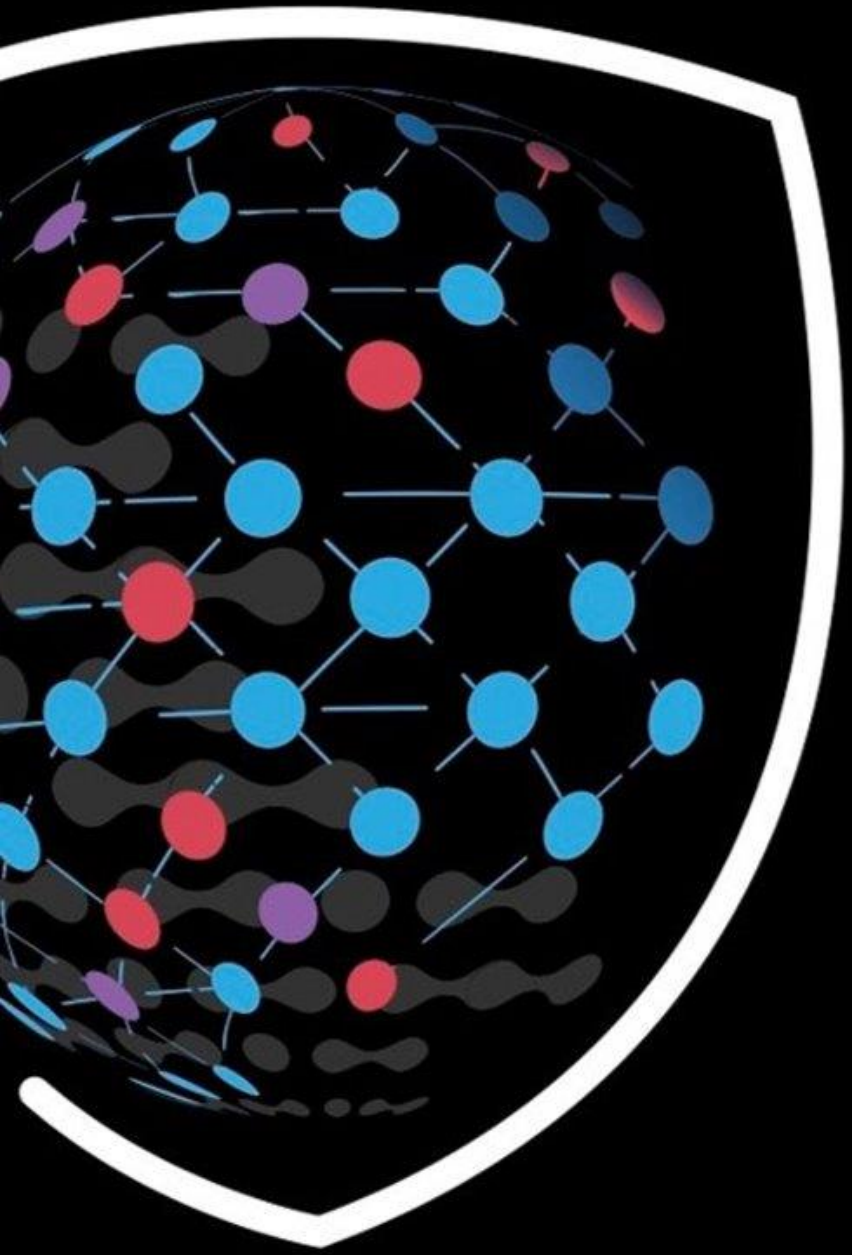
Quieres Seguir Los Demos? (Notebook #6)

https://otrf.github.io/workshop-ekoparty-bluespace-2020/conceptos-basicos/6_Analisis_Datos_Pandas_Visualizando_Datos.html

Visualizando Datos




```
In [4]: df = (
        df[['User']]

        [(df['EventID'] == 1) & (df['Channel'].str.contains('sysmon',case = False, na = False, regex

        .assign(Command_Length = df['CommandLine'].str.len())

        )
        df = df[df['Command_Length'] <300]
```



OPEN THREAT RESEARCH

EMPOWERING THE INFOSEC COMMUNITY