black hat ASIA 2025
APRIL 3-4, 2025
BRIEFINGS

The Problems of Embedded Python in Excel

(How to excel in pwning pandas)

Who are we?

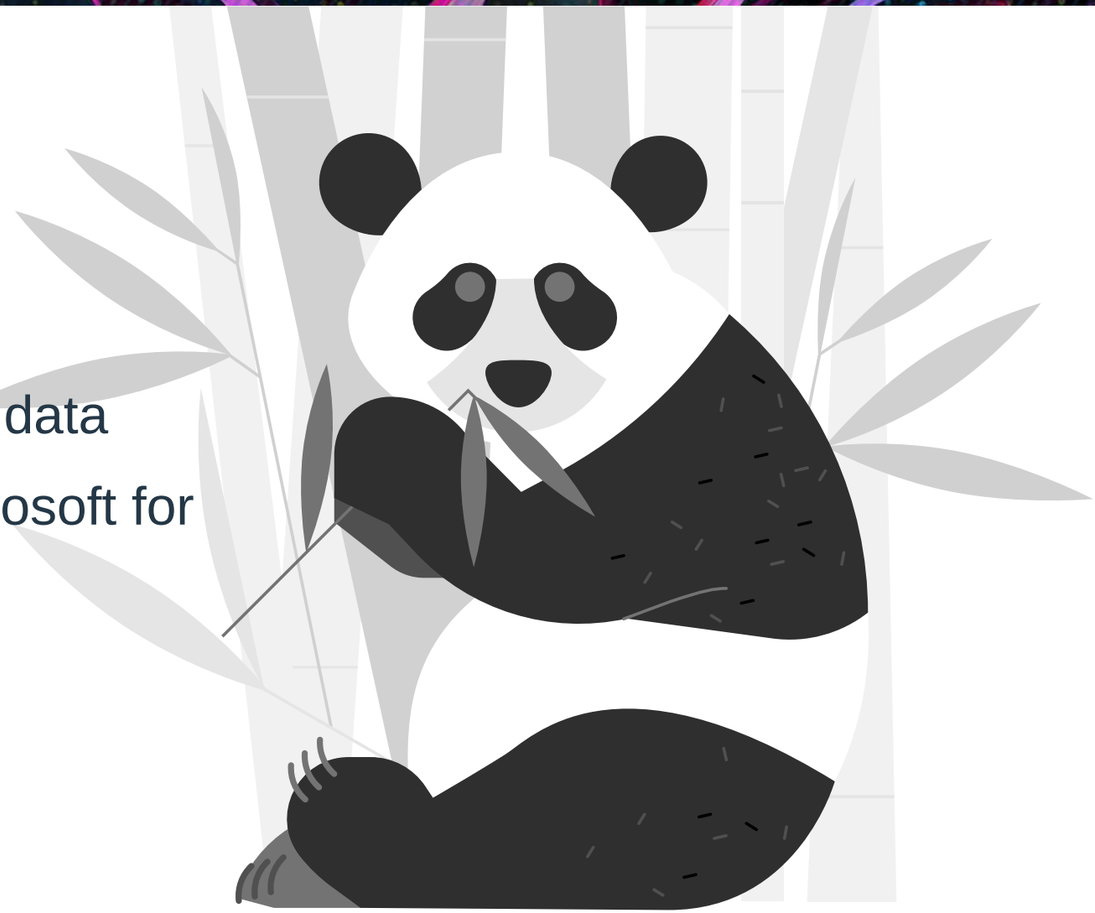
Shalom Carmel

CIO @ GlobalDots.com

Ofir Carmel

CS Student

TL;DR



The facts

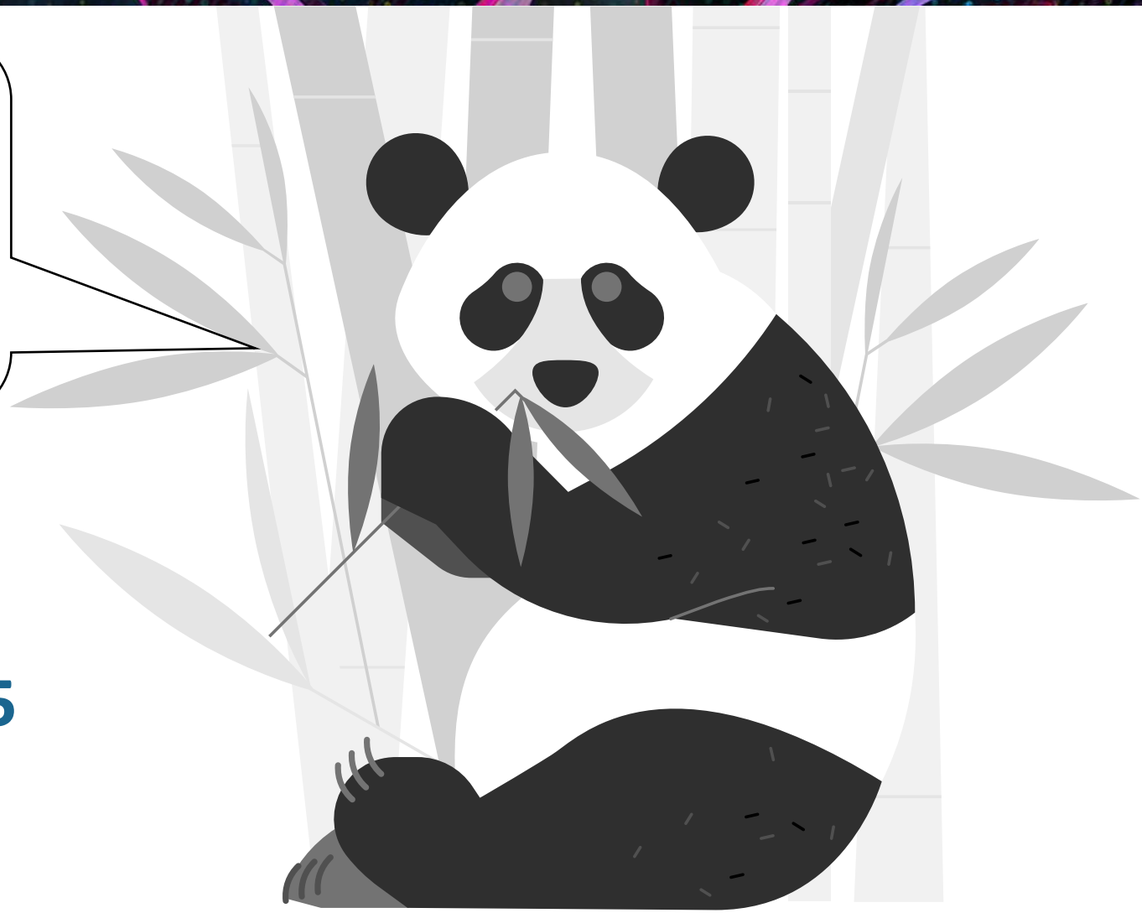
- Excel 365 runs python which refers to spreadsheet data
- The python code and the data are sent over to Microsoft for processing

I found out that

- We can play with and modify the PY cloud runtime environment
- PY cloud runtime environments seem to be shared between sessions[1]

[1] *See March 27, 2025 update*

Work In Progress / Proof Of Concept



https://github.com/shalomc/bhasia2025

Thank you, GlobalDots HR

HR made me take PTO near the end of the year.

My wife told me to fix the shed.

I pleaded for a couple of hours to look at some new productivity stuff.

This is the result, some days and nights later

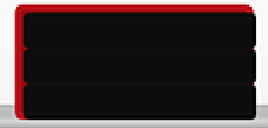
Timeline

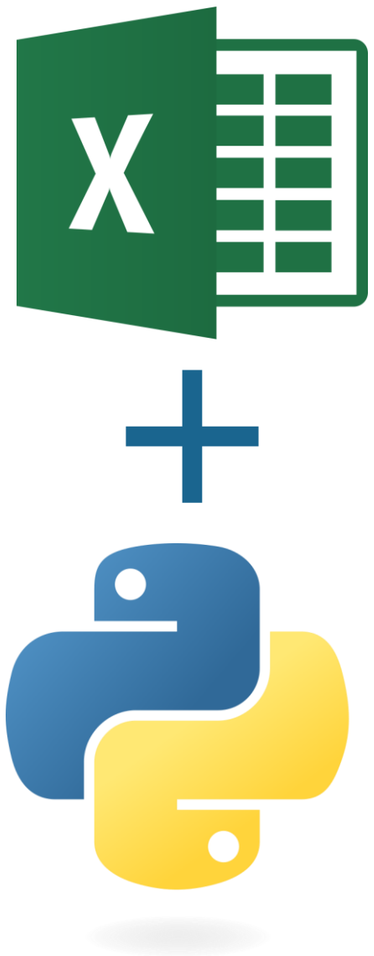
- Introduction
- The PY environment
- Execution of remote server code
- Shared user sessions (??)
- Uploading of custom binary files
- Live demo
- Summary
- Q&A

Introduction

- Excel is used by millions worldwide, and Excel 365 supports python
- The Python environment is geared at heavy data processing business and academia users
- It is expected to be a secure and private environment
- Prevalence of VB macros & automation in the finance sector





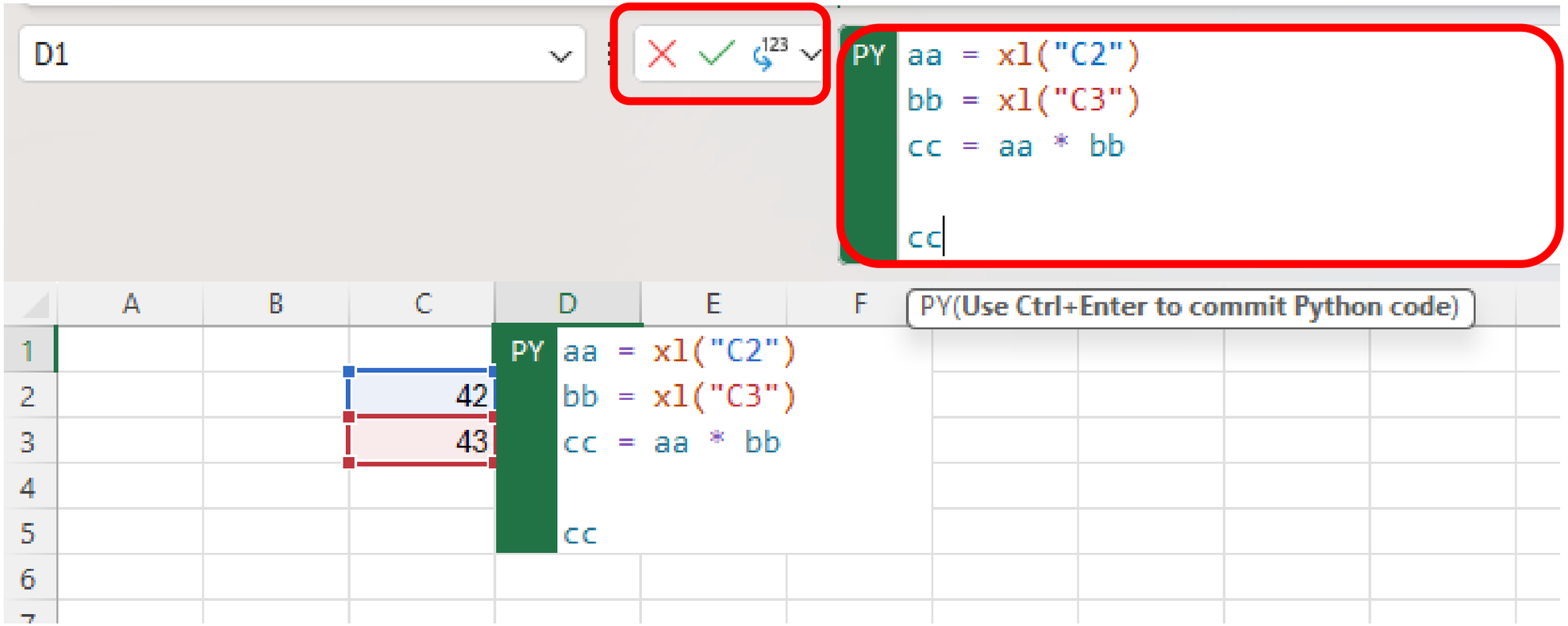


Excel-specific embedded Python extensions

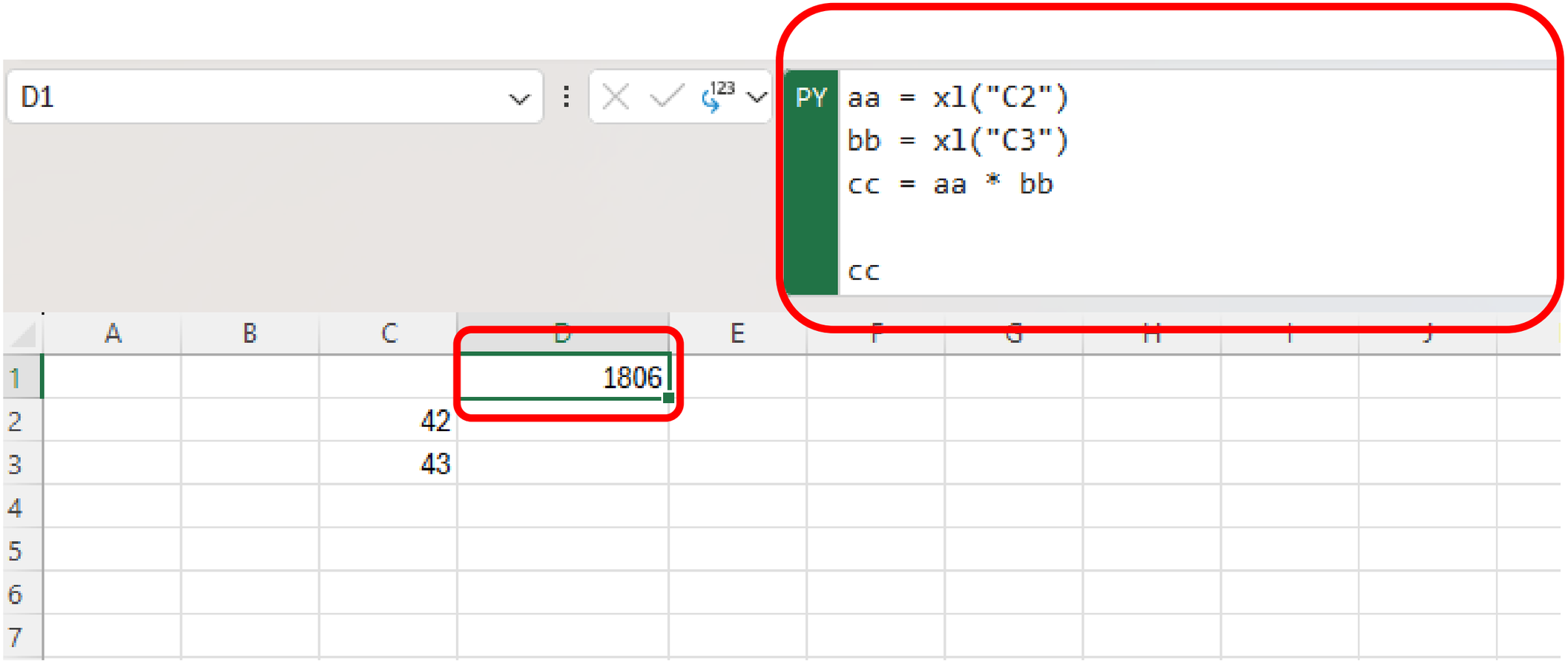
- =PY() Excel formula
- The xl() python function refers to Excel cells and ranges
- CTRL+Enter sends the code to execution
- Input options: cells, ranges, named ranges, tables
- Output options: pandas dataframes, lists, discrete values, python objects

How it works?

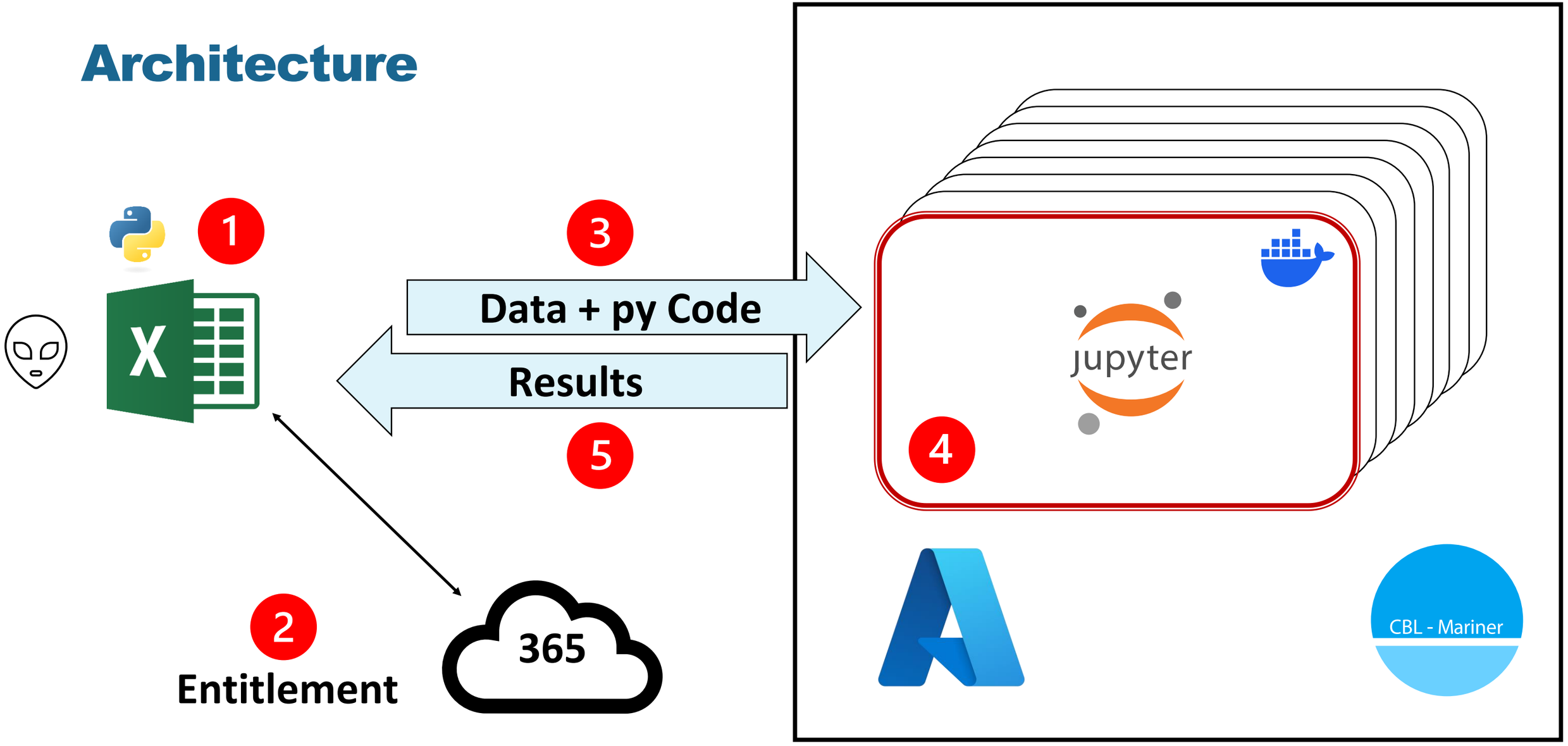
https://support.microsoft.com/en-us/office/get-started-with-python-in-excel-a33fbcbe-065b-41d3-82cf-23d05397f53d



How it works?



Architecture



Python in Excel Security

https://support.microsoft.com/en-gb/office/data-security-and-python-in-excel-33cc88a4-4a87-485e-9ff9-f35958278327

* The container has Python and a curated set of secured libraries provided by Anaconda.
* The Python code doesn't have access to your computer, devices, or account.
* The Python code doesn't have network access.
* The Python code doesn't have access to a user token.
* Python formulas have access to read cell values within the workbook, based on the cell reference, or values from external data sources, through the Power Query connection name.
* Python functions cannot return other object types like macros, VBA code, or other formulas.
* The Python code doesn’t have access to other properties in the workbook, such as formulas, charts, PivotTables, macros, or VBA code.
* The containers stay online as long as the workbook is open or until a timeout occurs.
* Data isn't persisted in the Microsoft Cloud.
* Python code runs within the compliance boundary of your organization.

Jupyter in Excel

- The spreadsheet becomes a Jupyter notebook
- Every Excel cell is a Jupyter cell. Evaluation order:
  - sheets → rows ↓ columns →
- Jupyter magic commands supported

Jupyter magic commands

Executed on the host (Azure container)

Command	Action
%system %sx !!	*Execute a shell command*
%env	*Returns a dictionary of environment variables*
%%bash	*Execute an entire bash script*
%lsmagic	*Lists available magic commands*

Interesting environment variables

```
list(os.environ.items())
```

```
sorted(os.environ.items(), key=lambda y: y[0].lower())
```

Variable	Value
OFFICEPY_OUTBOUND_BROKER_IP_ADDRESS	*10.0.1.250*
OFFICEPY_OUTBOUND_BROKER_PORT	*5050*
Fabric_NET-0-[Delegated]	*10.32.0.99*
Fabric_NodeIPOrFQDN	*10.92.0.12*
JUPYTER_TOKEN	*F57431********9cc9*
JPY_SESSION_NAME	*A5e*******38c6.py*

Python standard libraries

```
def list_installed_packages():
    try:
        import importlib.metadata
        installed_packages = importlib.metadata.distributions()
    except ImportError:
        installed_packages = []
    packages_list = [(package.metadata['Name'],
                      str(package.version),
                      str(package.metadata['Summary']),
                      str(package._path)
                      )
                      for package in installed_packages]
    packages_list_sorted = [(p00, p01, p02, p03 ) for p00, p01, p02, p03
                in sorted(packages_list, key=lambda y: y[0].lower())]
    return packages_list_sorted
list_installed_packages()
```

User writable file system

```
import os
def find_writable_directories(start_dir='/'):
    writable_dirs = []
    # Walk through the directory tree
    for root, dirs, files in os.walk(start_dir, onerror=lambda e: None):
        try:
            # Check if the directory is writable
            if os.access(root, os.W_OK):
                writable_dirs.append(root)
        except Exception as e:
            continue
    return writable_dirs

find_writable_directories()
```

User writable file system

Directory	Note
/app/officepy/*	
/app/officepy/lib/python3.12/*	*All of the python modules*
/app/officepy/share/jupyter/*	
/proc/[pid]/fd/	
/home/jovyan/.ipython/profile_default/startup/	*This is the IPython startup directory*
/home/jovyan/.local/share/jupyter/runtime/	*Jupyter connections and session files*

Some system commands

Command	Action
ping -c 3 4.4.4.4	*/bin/bash: line 1: ping: command not found*
dig www.blackhat.com	*/bin/bash: line 1: dig: command not found*
nmap localhost	*/bin/bash: line 1: nmap: command not found*
nc --v	*/bin/bash: line 1: nc: command not found*
which curl	*/bin/bash: line 1: which: command not found*
curl --help	*Usage: curl [options...] <url> -d, --data <data> HTTP POST data …..*
pip --version	*pip 24.0 from /app/officepy/lib/python3.12/site-packages/pip (python 3.12)*
rpm --version	*RPM version 4.18.0*

Install custom python module

```
%%bash


cd $HOME
mkdir fubar
cd fubar
mkdir excelpypwn
echo "fubar = 42" > excelpypwn/__init__.py
echo "from setuptools import setup, find_packages" > setup.py
echo "setup(" >> setup.py
echo "    name='excelpypwn'," >> setup.py
echo "    version='0.666.0'" >> setup.py
echo ")" >> setup.py
pip install .  > ../fubar.log
cd ..
```

Install custom python module

```
Processing /home/jovyan/fubar
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: excelpypwn
  Building wheel for excelpypwn (setup.py): started
  Building wheel for excelpypwn (setup.py): finished with status 'done'
  Created wheel for excelpypwn: filename=excelpypwn-0.666.0-py3-none-any.whl size=1168
sha256=2372634cb81cd91ce0ca431965ae4e43245049313f036aefd22528fc63077ad0
  Stored in directory: /tmp/pip-ephem-wheel-cache-
lnir11ew/wheels/d5/aa/24/40817f0f8c7b89493c6ccba31edcbc4c495be26e4987f7be54
Successfully built excelpypwn
Installing collected packages: excelpypwn
Successfully installed excelpypwn-0.666.0
```

Install custom python module

```
# verify installation of custom module
import excelpypwn
excelpypwn.fubar
```


Jupyter Sessions

March 27, 2025 Update

In a call with Microsoft security, we were informed that as of February 2025 there is a tightening of session management security, and there will be a session per open file instead of a session per user

According to Microsoft

- Containers are spun up pre-emptively to improve performance
- When there are not a lot of users, a container could be in the hotpool for a few hours.
- Before February, different documents from the same user could share the same container. If a user opened multiple workbook, or open/close a workbook multiple times, or clicked the Reset Runtime from Excel UI, there will be multiple Jupyter sessions created within a single container.
- After February, we changed to use one container per document per user. So different documents from the same users will not share the same container. There could still be multiple Jupyter sessions in a single container if a user open/close the same workbook multiple times, or clicked Reset Runtime from Excel UI.

How to find Jupyter sessions

```
%sx jupyter -paths

config:
    /home/jovyan/.jupyter
    /home/jovyan/.local/etc/jupyter
    /app/officepy/etc/jupyter
    /usr/local/etc/jupyter
    /etc/jupyter
data:
    /home/jovyan/.local/share/jupyter
    /app/officepy/share/jupyter
    /usr/local/share/jupyter
    /usr/share/jupyter
runtime:
    /home/jovyan/.local/share/jupyter/runtime
```

How to find Jupyter sessions

```
%sx ls -al $(jupyter --runtime-dir)


total 28
drwx-----T 2 jovyan users 4096 Mar 29 22:09 .
drwxr-xr-x 3 jovyan users 4096 Mar 29 19:44 ..
-rw-r--r-- 1 jovyan users  510 Mar 29 19:44 jpserver-71-open.html
-rw------- 1 jovyan users  293 Mar 29 19:44 jpserver-71.json
-rw------- 1 jovyan users   45 Mar 29 19:44 jupyter_cookie_secret
-rw------- 1 jovyan users  334 Mar 29 19:45 kernel-1bc54bd1-b96d-4b6a-b00f-f976054faa3f.json
-rw------- 1 jovyan users  334 Mar 29 22:09 kernel-81d23646-a712-4199-bf1f-14c8a95dd90e.json
```

How to find Jupyter sessions

```
%sx jupyter server list
Currently running servers:
http://SandboxHost-638789305605774754:8888/?token=a8c9a32b-fef8-4926-a45f-4889b9afd71a :: /mnt/file_upload

%sx cat /home/jovyan/.local/share/jupyter/runtime/jpserver*json | grep token
 "token": "a8c9a32b-fef8-4926-a45f-4889b9afd71a",

%env JUPYTER_TOKEN
a8c9a32b-fef8-4926-a45f-4889b9afd71a

os.environ["JUPYTER_TOKEN"]
a8c9a32b-fef8-4926-a45f-4889b9afd71a
```

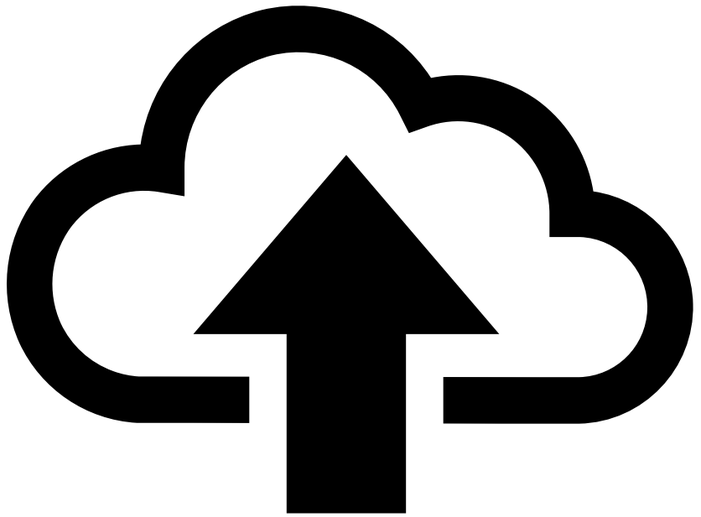
How to find Jupyter sessions

```
%sx curl -s http://${HOSTNAME}:8888/api/sessions?token=${token}
```


```
[
        {
                "id": "f3cddf3b-5cf3-45de-9145-2a7e54c5dde0",
                "path": "3872d4e5-564f-4c2b-aee9-a522473559de.py",
                "name": null,
                "type": "notebook",
                "kernel": {
                        "id": "859fdaa0-7703-4881-af62-18fe3130bdfc",
                        "name": "python3",
                        "last_activity": "2025-03-30T11:55:21.642558Z",
                        "execution_state": "busy",
                        "connections": 1
                },
                "notebook": {
                        "path": "3872d4e5-564f-4c2b-aee9-a522473559de.py",
                        "name": null
                }
        },
        {
                "id": "7192eeb7-c201-447b-9b4c-ff21e89325f2",
                "path": "f5ead0be-35b4-418a-9d25-eafc30b6a99f.py",
                "name": null,
                "type": "notebook",
                "kernel": {
                        "id": "d877ba13-961d-4df1-b109-0d87faf0d5a8",
                        "name": "python3",
                        "last_activity": "2025-03-30T11:38:40.905620Z",
                        "execution_state": "idle",
                        "connections": 1
                },
                "notebook": {
                        "path": "f5ead0be-35b4-418a-9d25-eafc30b6a99f.py",
                        "name": null
                }
        }
]
```


Upload binary files

- Unincluded python modules
- Overwrite anaconda curated modules with custom code
- Network analysis tools
- Container breakout attempts
- Useful Linux utilities



Upload binary files – practical steps

1. Create a base64 file from the binary
2. Don’t copy/paste
3. Using Excel PowerQuery, import the file to a named range
4. In your code, refer to **xl(“named_range”)**
5. In code, save the text to remote disk
6. Restore binary from base64 representation
7. Run rpm / tar / pip / chmod as necessary
8. Add the binaries to PATH, LD_LIBRARY_PATH, PYTHONPATH

Upload binary files tips

- Binaries from WSL work just fine
- Use the ldd command to find dependencies
  - Handy script on github
- Use VBA to automate process


Summary

Controlling python via registry

- Block python

```
reg add HKCU\software\policies\microsoft\office\16.0\excel\security /v
PythonFunctionWarnings /t REG_DWORD /d 2 /f
```

Value	Meaning
0	Allow all (default)
1	Warn
2	Block

Venues of research

- Jupyter is a code execution platform
  - https://www.youtube.com/watch?v=8vwo6gDANiY
- Is container breakout possible?
- Are sessions really limited?
- Exfiltration of data by custom pandas
- DOS by long-running containers

Attributions and thanks

- Jupyter logo By Cameron Oelsen - https://commons.wikimedia.org/w/index.php?curid=68763478

- www.globaldots.com

Black Hat Asia Sound Bytes

- Python in Excel is a remote code execution platform
- All new technology has bugs and security issues
- I am having fun, are you?


Q&A