

# Threat Intelligence with MSTICPy

---



**Ian Hellen**

Principal Software Development Engineer  
Microsoft Threat Intelligence Center (MSTIC)

@ianhellen [www.github.com/ianhelle](https://www.github.com/ianhelle)



# Overview



**MSTICPy introduction**

**Attacks on user accounts**

**Powershell scripting attacks**

**Adversary persistence**



# The Companion Notebook



## GitHub Repo and Notebook

<https://bit.ly/msticpy-btt>

<https://bit.ly/msticpy-btt-nb>



## MyBinder – run notebook in the cloud

<https://bit.ly/msticpy-btt-binder>



# MSTICPy Overview



**Python Cyber Security tools package – free, open source**



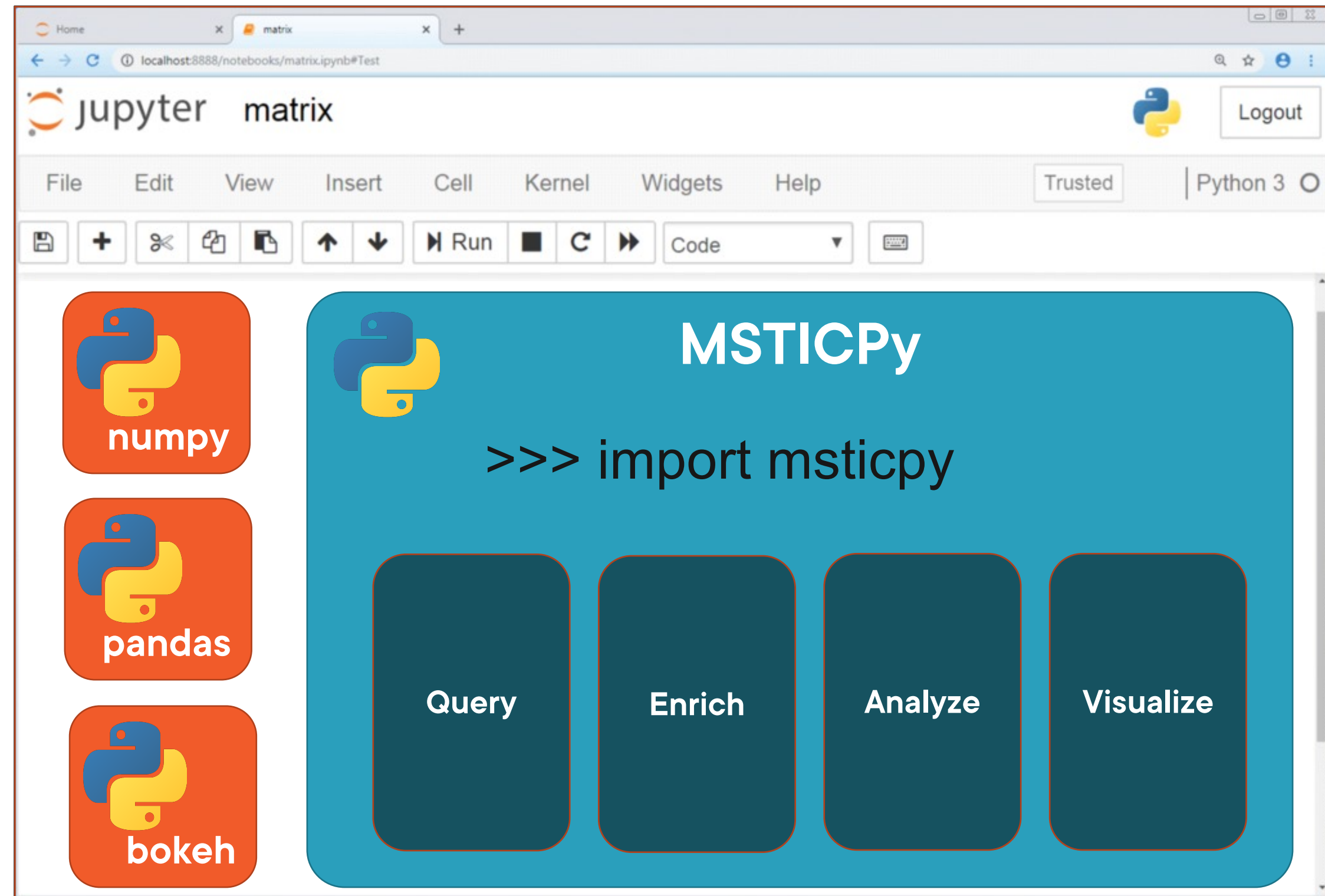
**Designed for Jupyter Notebooks but also usable in Python scripts and apps**



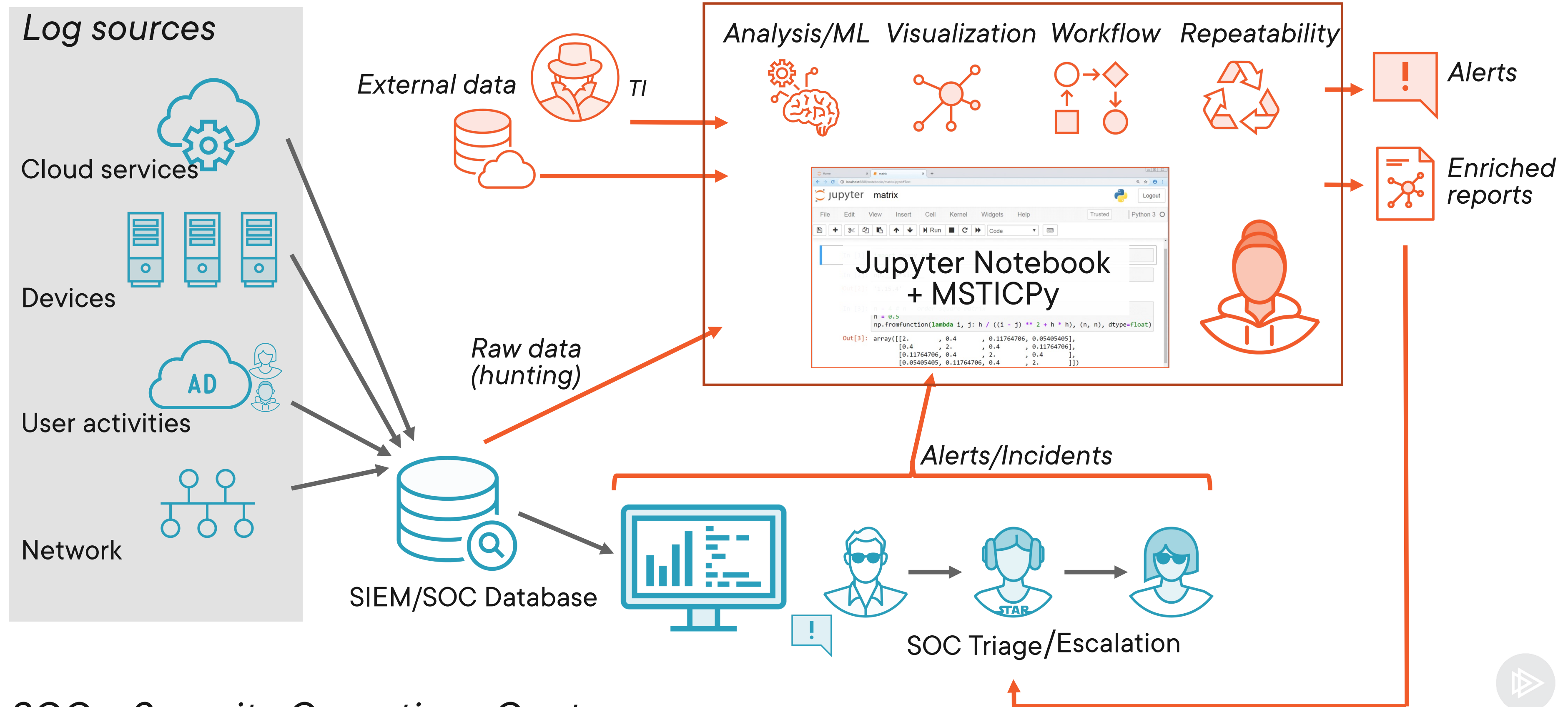
**An add-on for existing SIEM or log data source  
(*SIEM - Security Incident and Event Management*)**



Where does MSTICPy fit?



# MSTICPy and Notebooks in SOC Processes



# Prerequisites for This Course



**Jupyter notebooks environment with Python 3.8 or later.  
(Anaconda distribution is ideal)**



**Python coding (nothing complex) and Jupyter notebooks skills**



**MSTICPy installed `$ pip install msticpy` ↵  
and configured with the following:**



**API keys for one or more Threat Intelligence providers  
(VirusTotal, OTX, etc.)**



**API key for Maxmind GeoIPLite**



# Find out More



**Jupyter** - <https://jupyter.org>



**Anaconda** - <https://anaconda.com>



**Installing MSTICPy**  
[https://msticpy.readthedocs.io/getting\\_started/Installing](https://msticpy.readthedocs.io/getting_started/Installing)



**Configuring MSTICPy**  
[https://msticpy.readthedocs.io/getting\\_started/SettingsEditor](https://msticpy.readthedocs.io/getting_started/SettingsEditor)



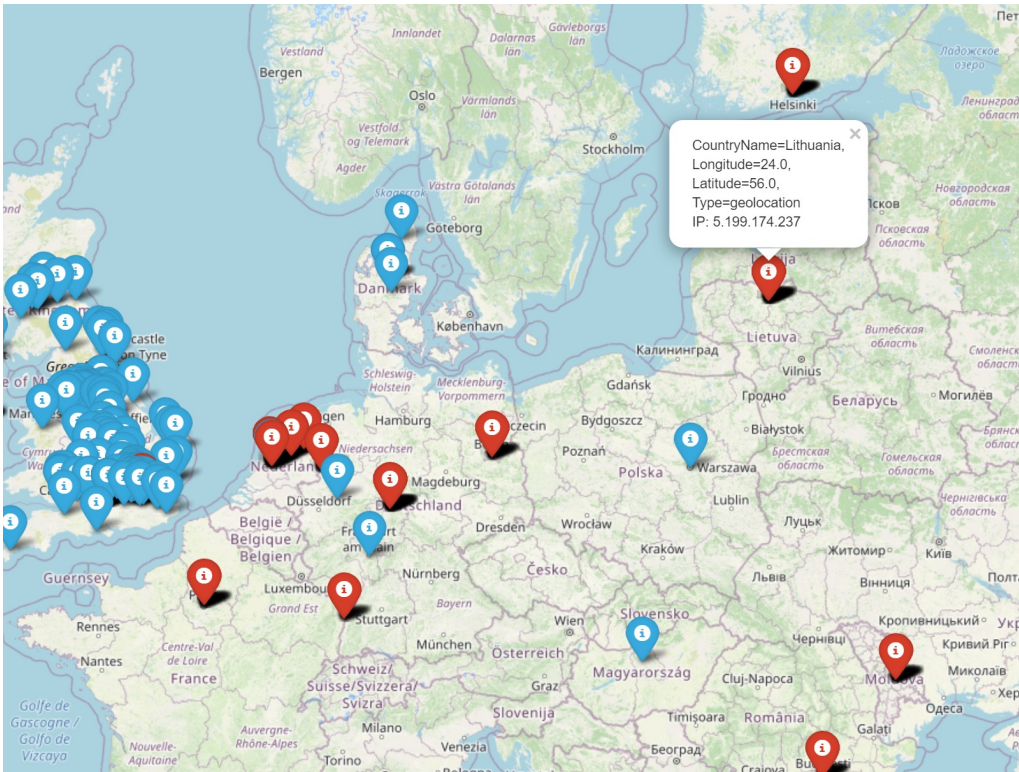


# Using MSTICPy to Understand Adversary Activity

---



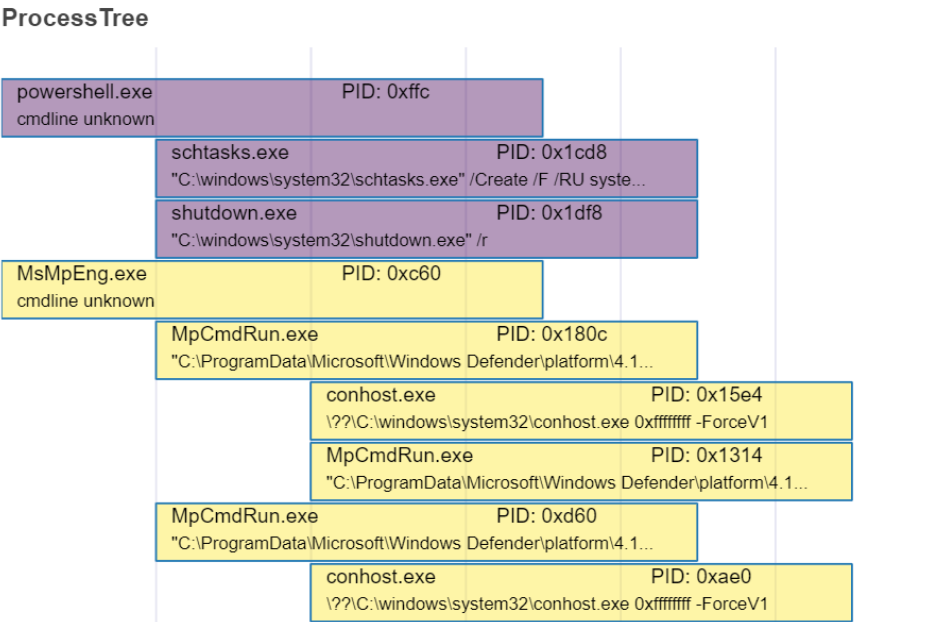
# MITRE ATT&CK – Techniques



**T1078.002 – Valid Accounts/Domain Accounts**

```
"$kernel32 = add-type -memberdefinition $methoddefinition -name 'kernel32' "
"-namespace 'win32' -passthru",
"$absd = 'amsis'+$canbuffer'",
"$handle = [win32.kernel32]::getmodulehandle('amsi.dll')",
'[IntPtr]$bufferaddress = [win32.kernel32]::getprocaddress($handle, $absd)',
'[uint32]$size = 0x5',
'[uint32]$protectflag = 0x40',
'[uint32]$oldprotectflag = 0',
'[win32.kernel32]::virtualprotect($bufferaddress, $size, $protectflag, '[ref]$oldprotectflag)',
'$buf = [byte[]]([uint32]0xb8,[uint32]0x57, [uint32]0x00, [uint32]0x07, '[uint32]0x80, [uint32]0xc3)',
'[system.runtime.interopservices.marshal]::copy($buf, 0, $bufferaddress, 6)',
```

**T1059.001 – Command and Scripting Interpreter: PowerShell**



**T1053.005 – Scheduled Task/Job: Scheduled Task**



# Demo



## Setup



# Account Attacks – Password Spray

---



# T1078.002 - Valid Accounts/ Domain Accounts

Automated attacks against existing accounts are common.

Password spray attacks can be launched with a variety of tools and techniques.

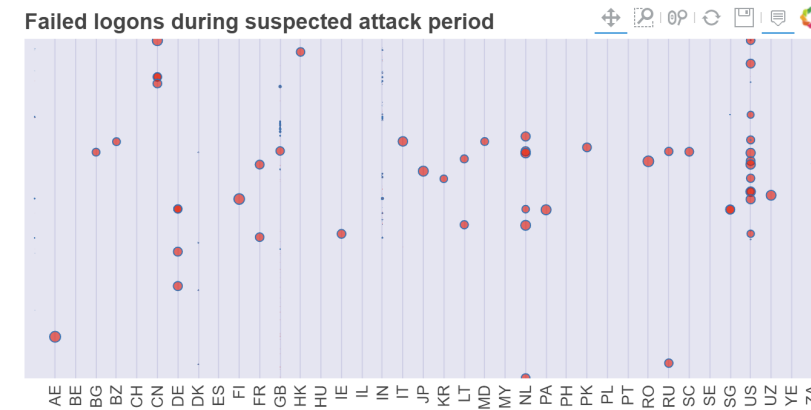
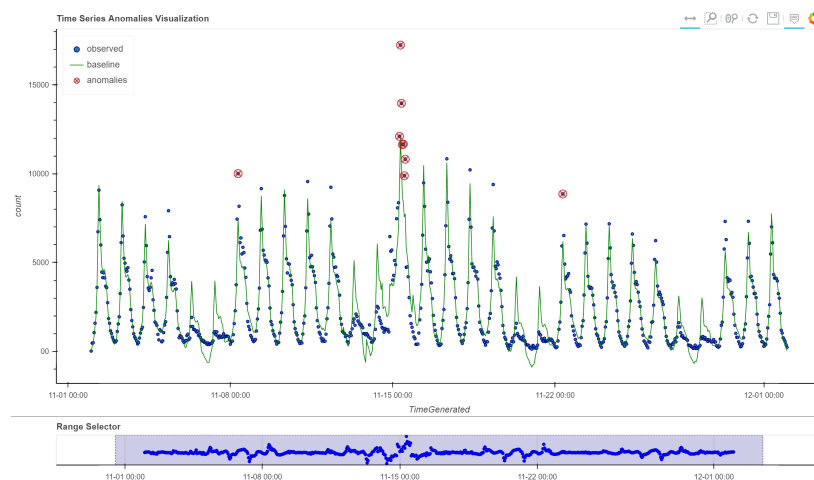
*“Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion.*

**Adversaries may compromise domain accounts**, some with a high level of privileges, **through various means such as** OS Credential Dumping or **password reuse**, allowing access to privileged resources of the domain.”

MITRE ATT&CK



# Overview



121.196.189.242	
Type: 'ipv4', Provider: OTX, severity: high	
Details	
OTX	
pulse_count	12
names	['Malware Command and Control IPs', 'C&C - IP', 'Stacey Grubb', 'Grubb', 'Stacey Grubb', 'Stacey Grubb', 'Stacey Grubb', 'Stacey Grubb', 'Stacey Grubb', 'malware and ip SG']

Detection –  
Time Series  
Analysis

Identification –  
Matrix plots

Isolation –  
determine  
characteristics  
of attackers

Assessment –  
What damage  
has been done



# Demo



## Investigating an Account Attack

- Detecting anomalous logons
- Identifying attacker logons
- Isolating malicious from benign events
- Damage assessment



# Next Steps



## Widen the Search

- Look for IP addresses in other logs
- Isolate more properties from attack events
- Search for those in successful events

## Investigate and Remediate: for example:

- Disable accounts
- Investigate user resources
- Isolate compromised machines
- ...



# Account Attacks - Summary

## Time series

Helps us identify anomalies in bulk activity

## Matrix plots

Help identify common characteristics of attack

## Mapping

Helps visualize geographic anomalies

## Threat Intelligence

Understand origin and techniques of attack

## Damage assessment

Where did the attackers breach?



# Scripted Attacks

---



# T1059.001 – Command and Scripting Interpreter: PowerShell

**Scripted attacks are the norm in both automated and manual intrusions.**

**Adversaries typically try to disguise their intent.**

*“Adversaries may abuse PowerShell commands and scripts for execution.*

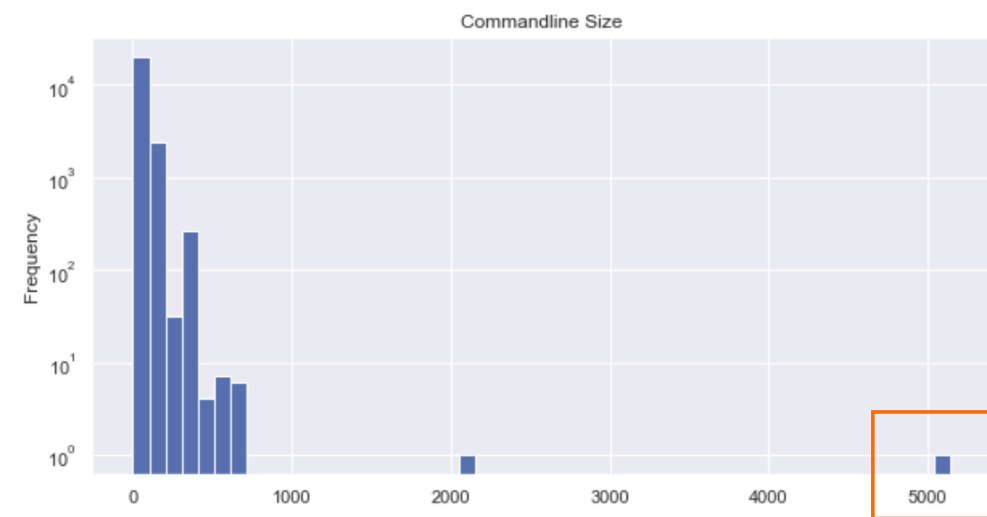
*Adversaries can use PowerShell to perform a number of actions, including **discovery of information** and **execution of code**.*

*PowerShell may also be used to download and run executables from the Internet, which **can be executed** from disk or **in memory without touching disk**.”*

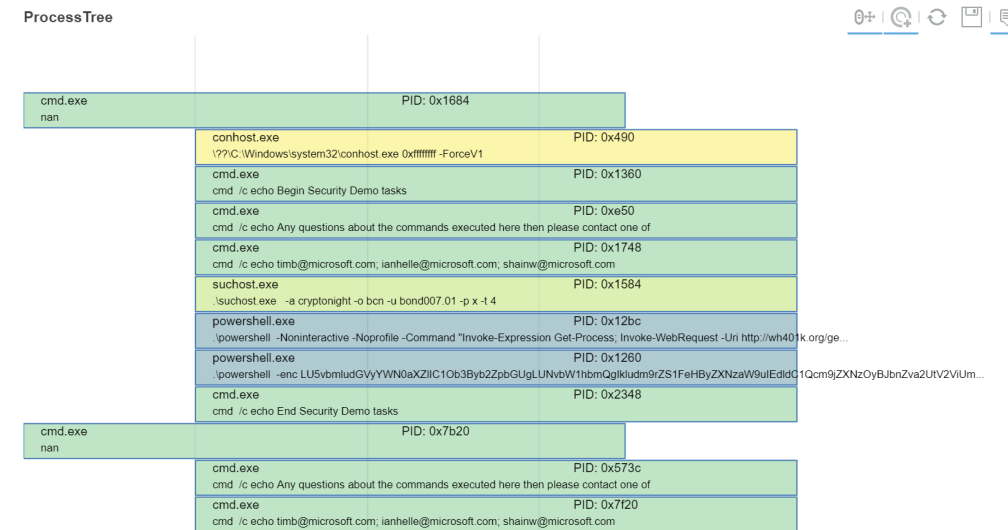
MITRE ATT&CK



# Overview



**Detection – Command  
line analysis**



**Identification –  
Process tree**

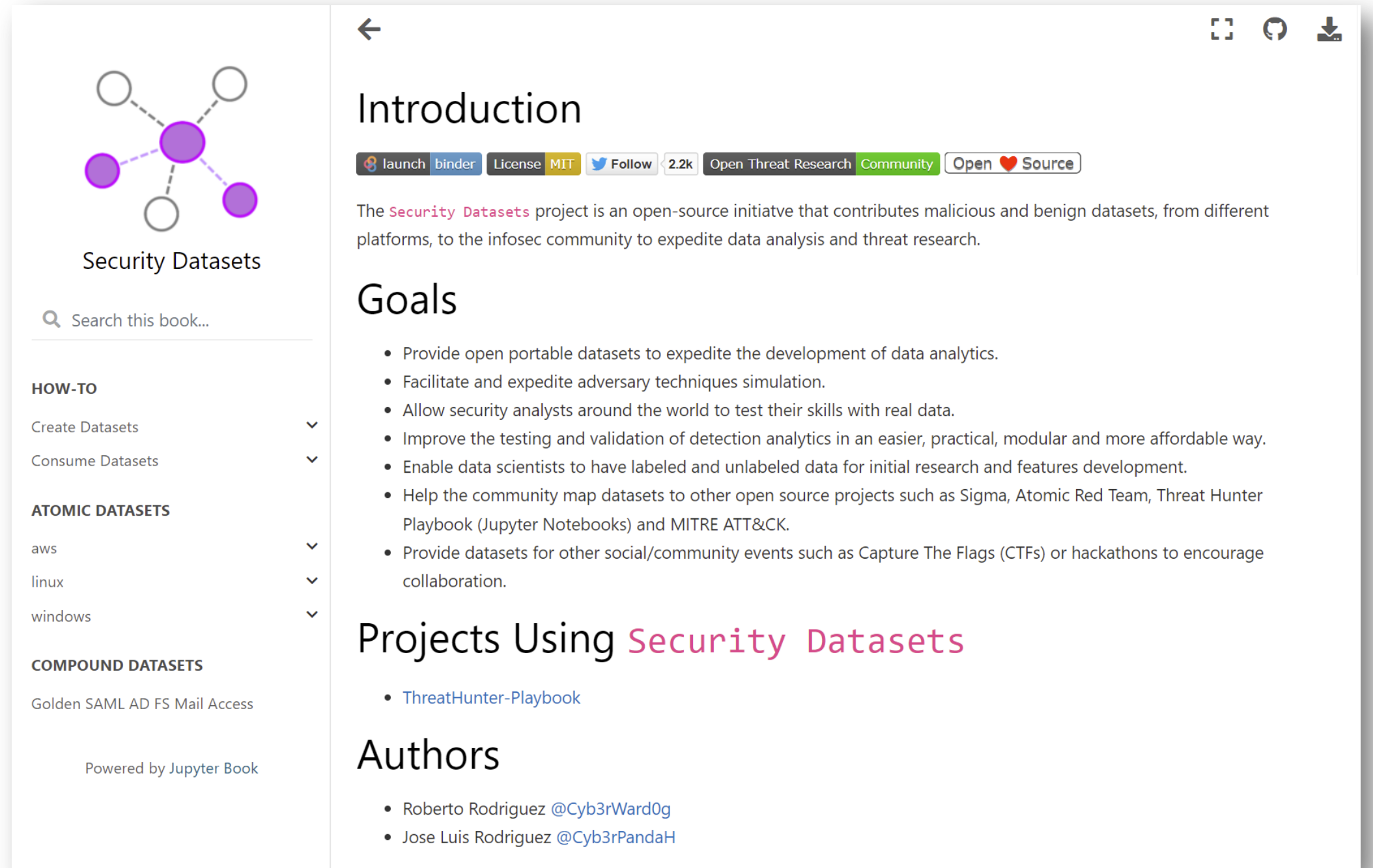
```
from msticpy.vis.code_view import display_html
display_html(format_powershell(decoded_script), language="powershell")
✓ 0.3s

if($psversiontable.psversion.major -ge 3)
{
    $6866=[ref].assembly.gettype('system.management.automation.utils')."getfield"('
    if($6866)
    {
        $1fe7=$6866.getvalue($null)
        if($1fe7['scriptblocklogging'])
        {
            $1fe7['scriptblocklogging']['enablescripblocklogging']=0
            $1fe7['scriptblocklogging']['enablescripblockinvocationlogging']=0
        }
        $val=[collections.generic.dictionary[string,system.object]]::new()
        $val.add('enablescripblocklogging',0)
        $val.add('enablescripblockinvocationlogging',0)
        $1fe7['hkey_local_machine\software\policies\microsoft\windows\powershell\sc
    }
}
```

**Assessment – What  
damage has been  
done?**



# OTRF Security Datasets



The screenshot displays the GitHub repository page for 'Security Datasets'. The left sidebar contains a navigation menu with sections: 'HOW-TO' (Create Datasets, Consume Datasets), 'ATOMIC DATASETS' (aws, linux, windows), and 'COMPOUND DATASETS' (Golden SAML AD FS Mail Access). The main content area features an 'Introduction' section with a description of the project as an open-source initiative for malicious and benign datasets. Below this is a 'Goals' section with a bulleted list of objectives, followed by a 'Projects Using Security Datasets' section listing 'ThreatHunter-Playbook'. The 'Authors' section at the bottom lists Roberto Rodriguez (@Cyb3rWard0g) and Jose Luis Rodriguez (@Cyb3rPandaH). The page includes various GitHub badges for launch, binder, license (MIT), follow (2.2k), and open source.

**Security Datasets**

Search this book...

**HOW-TO**

- Create Datasets
- Consume Datasets

**ATOMIC DATASETS**

- aws
- linux
- windows

**COMPOUND DATASETS**

- Golden SAML AD FS Mail Access

Powered by Jupyter Book

## Introduction

launch binder License MIT Follow 2.2k Open Threat Research Community Open Source

The **Security Datasets** project is an open-source initiative that contributes malicious and benign datasets, from different platforms, to the infosec community to expedite data analysis and threat research.

## Goals

- Provide open portable datasets to expedite the development of data analytics.
- Facilitate and expedite adversary techniques simulation.
- Allow security analysts around the world to test their skills with real data.
- Improve the testing and validation of detection analytics in an easier, practical, modular and more affordable way.
- Enable data scientists to have labeled and unlabeled data for initial research and features development.
- Help the community map datasets to other open source projects such as Sigma, Atomic Red Team, Threat Hunter Playbook (Jupyter Notebooks) and MITRE ATT&CK.
- Provide datasets for other social/community events such as Capture The Flags (CTFs) or hackathons to encourage collaboration.

## Projects Using Security Datasets

- [ThreatHunter-Playbook](#)

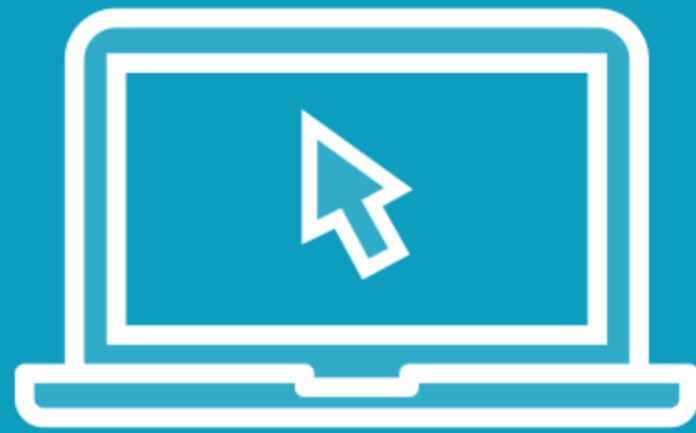
## Authors

- Roberto Rodriguez [@Cyb3rWard0g](#)
- Jose Luis Rodriguez [@Cyb3rPandaH](#)

**Open Threat Research Forge**  
Security Datasets - <https://securitydatasets.com>



# Demo



## **Investigating Endpoint Activity**

- Encoded PowerShell scripts
- Decoding and interpreting



# PowerShell Script Attacks - Summary

## Decoding

Help us identify obfuscated commands

## Process Tree

View command in context

## View Script

Understand technique of attack

## Identify the code

Find attack code sample



# Identifying Persistence

---





# T1053.005 - Scheduled Task/Job: Scheduled Task

Installing scheduled tasks (Windows) or cron jobs (Linux) is one of the most common ways of creating a persistent presence on a compromised host.

*“Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code.*

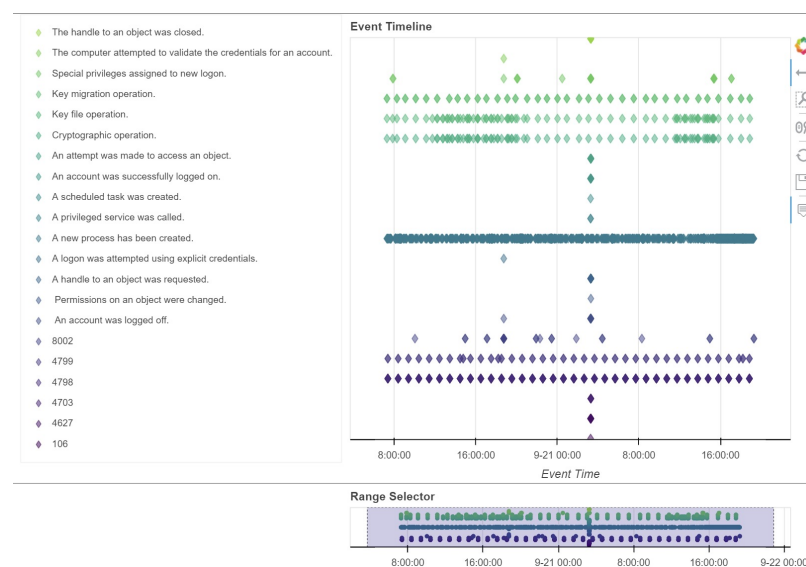
*In some cases, adversaries have used a **.NET wrapper for the Windows Task Scheduler**, and alternatively, adversaries have used the Windows netapi32 library to create a scheduled task.*

*An adversary may use Windows Task Scheduler to **execute programs at system startup or on a scheduled basis for persistence.**”*

MITRE ATT&CK



# Overview



**Detection – Event analysis**

```
print(
    sch_tasks_procs[sch_tasks_procs.CommandLine.str.contains("schtasks", case=False)]
    .iloc[0]
    .CommandLine
    .replace("/", "\n /").replace("-", "\n -")
)
✓ 0.1s

"C:\windows\system32\schtasks.exe"
/Create
/F
/SC DAILY
/ST 09:00
/TN MordorSchtask
/TR "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
-NonI
-W hidden
-c "IEX ([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String((gp HKCU:\Software\Microsoft
debug).debug)))\""
```

**Identification – Task parameters**

```
from msticpy.vis.code_view import display_html
display_html(format_powershell(decoded_script), language="powershell")
✓ 0.3s

if($psversiontable.psversion.major -ge 3)
{
    $6866=[ref].assembly.gettype('system.management.automation.utils')."getfield"('
    if($6866)
    {
        $1fe7=$6866.getvalue($null)
        if($1fe7['scriptblocklogging'])
        {
            $1fe7['scriptblocklogging']['enablescriptblocklogging']=0
            $1fe7['scriptblocklogging']['enablescriptblockinvocationlogging']=0
        }
        $val=[collections.generic.dictionary[string,system.object]]::new()
        $val.add('enablescriptblocklogging',0)
        $val.add('enablescriptblockinvocationlogging',0)
        $1fe7['hkey_local_machine\software\policies\microsoft\windows\powershell\sc
    }
}
```

**Assessment – What damage has been done?**



# Demo



## **Schedule tasks and attacker persistence**

- Finding scheduled task creation
- What is the task doing?



# Scheduled Tasks and Persistence - Summary

## Event analysis

Finding schedule task  
creation events

## Extracting parameters

Identifying what the  
task is doing

## Decoding

Piecing together the  
attack elements



# Conclusion

---



# Conclusion



**Introduced MSTICPy in Jupyter notebooks for InfoSec**



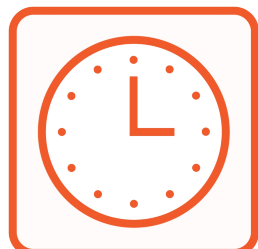
**Looked at three MITRE ATT&CK Techniques:**



**1. Attacks on Valid Accounts**



**2. Scripted attacks using PowerShell**



**3. Persistence with Windows Scheduled Tasks**



# Conclusion



**Introduced MSTICPy in Jupyter notebooks for InfoSec**



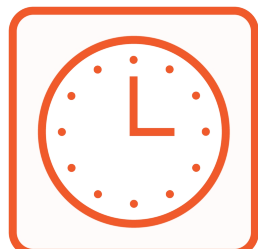
**Looked at three MITRE ATT&CK Techniques:**



**1. Attacks on Valid Accounts**



**2. Scripted attacks using PowerShell**



**3. Persistence with Windows Scheduled Tasks**



# Threat Intelligence with MSTICPy

---

## Resources



**Ian Hellen**

Principal Software Development Engineer  
Microsoft Threat Intelligence Center (MSTIC)

@ianhellen [www.github.com/ianhelle](https://www.github.com/ianhelle)





# MSTICPy Resources

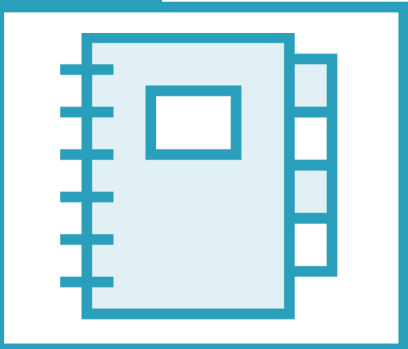


**MSTICPy Documentation, <https://msticpy.readthedocs.io>**



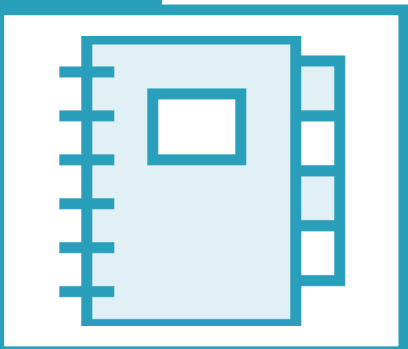
**Configuring MSTICPy**

**[https://msticpy.readthedocs.io/getting\\_started/SettingsEditor](https://msticpy.readthedocs.io/getting_started/SettingsEditor)**



**Example Microsoft Sentinel Notebooks using MSTICPy**

**<https://github.com/azure/azure-sentinel-notebooks>**



**MSTICPy Feature notebooks**

**<https://github.com/microsoft/msticpy/tree/main/docs/notebooks>**



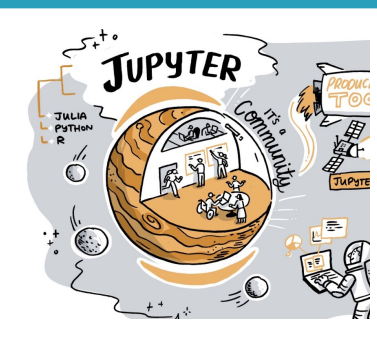
# Jupyter and Python Resources



**Jupyter** – <https://jupyter.org>



**Anaconda** – <https://anaconda.com>



**InfoSec Jupyterthon – Using notebooks in InfoSec Workshops**  
<https://infosecjupyterthon.com/>



**Pandas User Guide**  
[https://pandas.pydata.org/docs/user\\_guide/index.html](https://pandas.pydata.org/docs/user_guide/index.html)



# The Companion Notebook



## GitHub Repo and Notebook

<https://bit.ly/msticpy-btt-repo>

<https://bit.ly/msticpy-btt-notebook>



## MyBinder – run notebook in the cloud

<https://bit.ly/msticpy-btt-binder>



# Other Resources



**MITRE ATT&CK**

<https://attack.mitre.org/>



**Open Threat Research Forge Security Datasets**

<https://securitydatasets.com>



**UltimateWindowsSecurity.com**

<https://www.ultimatewindowssecurity.com/securitylog/>

