Leveraging User Activity Logs for Analysis & Predictions with Python

RAZI RAIS

Senior Technical Program Manager
Identity Engineering Microsoft

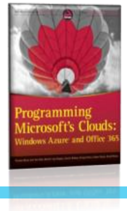
JUNE 2021 | #identiverse

identiverse®

Razi Rais

SCAN ME

- ✓ 16+ Years | Engineering | Architecture | Training | Authoring
- ✓ Software Engineer → Architect → Technical Program Manager
- ✓ Areas of Interest → Identity, Blockchain, Privacy (Homomorphic Encryption, etc.).
- ✓ Living in New York City but worked in Asia, Middle East & Europe
- ✓ I like stand-up comedy, writing, cycling, and yoga.



Single tool for the following tasks on logs ?

Gathering

Analyzing

Visualization

Prediction

Cross Platform

Agenda

- Analysis, Visualization, and Predictions Using Python
- Demos

Prerequisites

- Familiarity with Python OR any modern programming language

Resources:

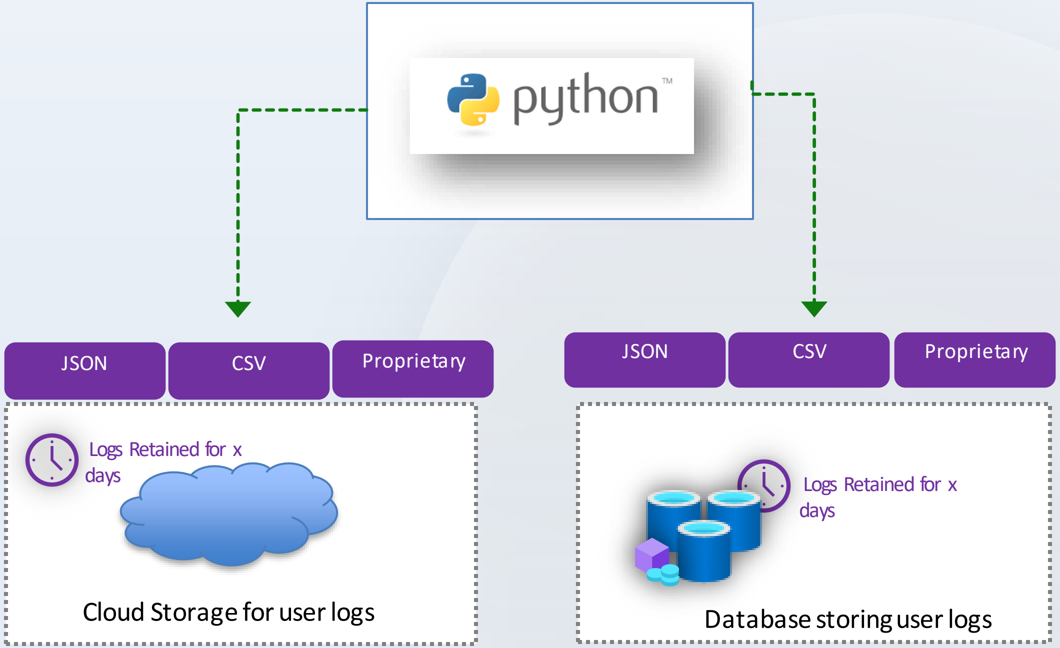
- Download Demos : https://aka.ms/sessiondemos

Gathering & Parsing User Logs

Problem:

Logs are scattered across various sources and in different formats.

Identity & Access Management solutions/ services rely on JSON, CSV, and other propriety formats to persist user logs

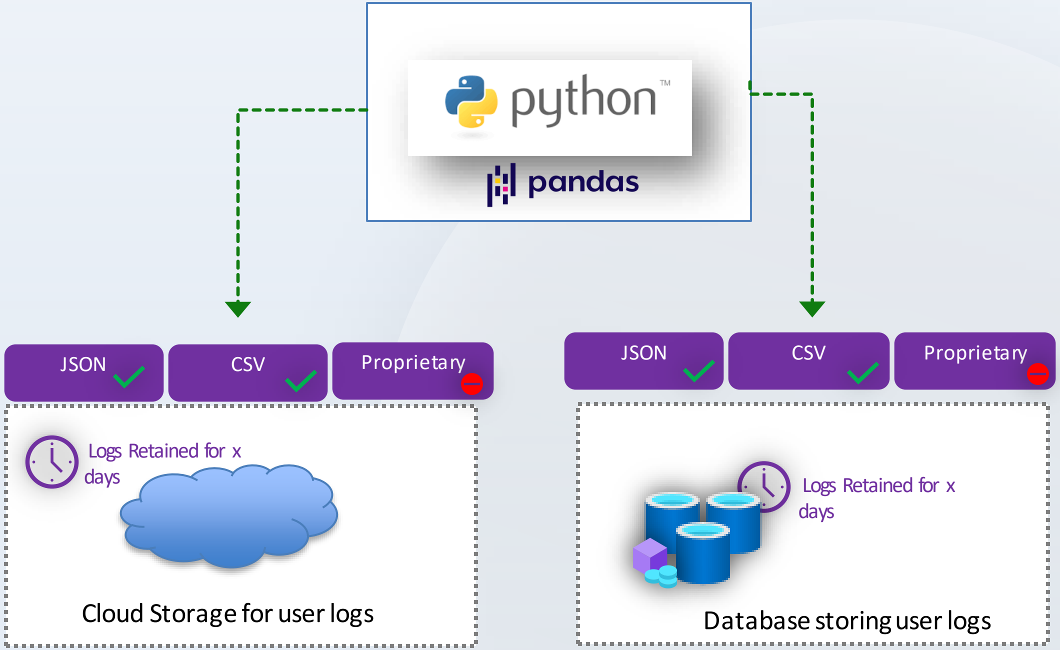


Gathering & Parsing User Logs

Solution:

✓ Use native Python libraries to load the data easily for csv, json and other common data formats.

✓ Avoid proprietary data formats especially if they are binary.



python™

pandas

JSON ✓ | CSV ✓ | Proprietary ⛔

Logs Retained for x days

Cloud Storage for user logs

JSON ✓ | CSV ✓ | Proprietary ⛔

Logs Retained for x days

Database storing user logs

Demo:

Gathering User Logs using Python

Analyzing User Logs

Problem :

User attributes are scattered.

For example, *id* and ***name*** are in one log file and **geo-location** data is in a different file.

Sign-In activity

ID	Name	Activity
123	Adam	Pwd Reset
456	Eve	Profile Edit

+

Sign-In Geolocation

ID	Longitude	Latitude
123	40.7831	73.9712
456	32.2946	64.7859

Sign-In activity with geolocation!

ID	Name	Activity	Longitude	Latitude
123	Adam	Pwd Reset	40.7831	73.9712
456	Eve	Profile Edit	32.2946	64.7859

Analyzing User Logs

Solution

✓ **Python's Pandas library** help you perform merge across the data based on the desired criteria.

Sign-In activity

ID	Name	Activity
123	Adam	Pwd Reset
456	Eve	Profile Edit

+

Sign-In Geolocation

ID	Longitude	Latitude
123	40.7831	73.9712
456	32.2946	64.7859

Sign-In activity with geolocation!

ID	Name	Activity	Longitude	Latitude
123	Adam	Pwd Reset	40.7831	73.9712
456	Eve	Profile Edit	32.2946	64.7859

✓

pandas

Demo:
Analyzing Logs Using Pandas

Analysis via Visualization

When dealing with large dataset, using *visualization* over *tabular* format radically improves analysis

- Users Sign-In activity on world map
- Top “n” User Agent, Devices, Sign-In activity load per month etc.
- Multivariable analysis: Top “n” User Agent + IP Addresses

A picture is worth a thousand words

Common Python Visualization Libraries

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Things to consider

- Highly customizable and foundation for other libraries like Seaborn and Ploty
- Can be too low level
- Too many parameters to customize

Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing statistical graphics.

Things to consider

- Wrapper around Metaplotlib
- Specializes in statistics visualization
- Default themes
- Less parameters compare to matplotlib

Plotly

Plotly is a Python graphing library to make interactive, publication-quality graphs.

Things to consider

- Wrapper around Metaplotlib
- Interactive clickable charts & graphs

ggplot2

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics by mapping variables to aesthetics.

Things to consider

- Wrapper around Metaplotlib
- Port of ggplot2 for R
- Use ”Grammar of Graphics” approach.
- Less mature than seaborn and Plotly.

Demo:
Log Analysis via Visualization

Predictions using Python ML Libraries

Can you predict following based on user logs?

- Predicting whether user activity is suspicious/anomalous
- Predicting whether attack like password spray/stuffing is taking place
- Predict if sign-ups are legitimate users or from a bot

Log Analysis

- User Agent
- Geo Location ^{PII}
- Phone Numbers ^{PII}
- IP Address ^{PII}
- Timestamp
- Display Names
- Email ^{PII}
- User ID
- MFA Details (SMS/Phone etc.)
- Social IdP Identifiers
- HTTP Codes
- Activity

Feature Extraction

The main purpose of this step is to extract valuable features from log events that could be fed into Machine Learning models.

Example:

- Suspicious Activity: User ID + Timestamp + Geo Location
- Bot Attack: Timestamp + IP + Geolocation
- Password Spray: HTTP Error Code + User ID + Timestamp
- Credential Stuffing: HTTP Error Code + User ID + Timestamp

Predictions (ML)

- Anomaly Detection
- Password Attacks

Machine Learning Techniques

- Supervised
- Unsupervised
- Reinforcement Learning

Python Libraries (non-exhaustive)

- scikit-learn
- Keras
- PyTorch
- PyCaret
- Tensorflow
- MSTIC
- Scipy
- Ledwig (AuthML)

Identity Developer Cheat Sheet

Ask yourself:

i. What is the objective of this analysis?

ii. Do I need to gather more data?

iii. Should I discard data not needed?

Do more of these in iterations:

- Prioritize visualization over tabular format
- Use Python notebooks for quick analysis
- Keep data sample size small but iterate over variety of data.
- Update your assumptions based on outcome of analysis (*not the other way around*)

Pay special attention to these:

- PII
- Outliers
- Causality vs Correlation
- Precision vs Recall

Cutting Edge: Data Analysis & Prediction

- Federated Learning
- Privacy Preserving Techniques

Look for these in the logs:

- User Agent
- Geo Location ^{PII}
- Phone Numbers ^{PII}
- IP Address ^{PII}
- Timestamp
- Display Names
- Email ^{PII}
- User ID
- MFA Details (SMS/Phone etc.)
- Social IdP Identifiers
- HTTP Codes
- Activity

+ Extend with server/service logs

- Token Success/Failures
- Protocol Specific Codes

+ Extend with Application logs

- Application Failures
- MFA Timestamp
- Biometrics

Cybersecurity Attacks Analysis

- Unfamiliar sign-ins : Geolocation + Timestamp
- IRSF attack: Phone Number
- Bot Attack: Timestamp + IP + Geolocation + HTTP Status
- Password Spray: HTTP Status+ User ID + Timestamp
- Credential Stuffing: HTTP Status + User ID + Timestamp
- Leaked Credentials: Email/User ID + Timestamp + HTTP Status

User Insights Analysis

- Most active user/dormant users: User ID + Timestamp
- User activity (sign-in/password reset, etc.): User ID + Activity + Timestamp
- Convergence rate (sign-up vs sign): User ID + Activity+ Timestamp
- Success/Failure Rate: User ID + HTTP codes
- Device Usage: User ID + User Agent
- Top User Locations: Geo Location

Python Libraries (non-exhaustive)

- Analysis & Visualization
  - Pandas
  - NumPy
  - MSTIC
  - Theano
  - Matplotlib
  - Seaborn
  - Ploty
  - Plotline
  - Vega
  - Vispy
  - Bokeh
  - Orange
- Prediction (Machine Learning)
  - scikit-learn
  - Keras
  - PyTorch
  - Tensorflow
  - Scipy
  - Ludwig (AutoML)

Thank You!

Please take our quick survey for a chance to win a pair of Microsoft Surface Earbuds

To access the survey, follow this link:

aka.ms/Identiverse21survey

-or-

Scan the QR code below:

Q&A