Appendix 2  
3.2.4-3.2.5\_攻撃パスを通常通信ルートに限定して抽出

**概要：**

* 組織内システムの「通常時に確立されている通信ルート」を基準として、その正規通信経路上の資産間のみから成立する攻撃パスを自動抽出する手法の実装例。
* ExcelやCSV形式の資産－通信経路データを解析し、MITRE ATT&CK Cloud Matrixの各攻撃テクニックを「実際に現場でありうる（＝閉じられた通信網で成立する）最小限の経路」に絞って抽出。
* IPA方式で推奨される「攻撃コストが低い正規通信ルート上のみを優先」する方針を実現し、想定外の通信・横展開リスクの現実的な可視化に寄与。
* 研究や実務においてシナリオ分析や攻撃グラフ抽出の自動化の基盤となり、特にクラウド・ゼロトラスト環境のリスクモデル運用に直結する処理例を示す。
* 各種ファイル形式の前処理・パラメータ指定も柔軟で、実際の運用現場データで容易に検証・応用可能。

Google Colab用Pythonコード（正規通信路限定攻撃パス抽出・Excel/CSVファイル入力自動化）

# 必要ライブラリのインストール（初回のみ）

!pip install -q pandas openpyxl

# 手動アップロード（stepwise\_techniques\_assets.xlsx）

from google.colab import files

print("『stepwise\_techniques\_assets.xlsx』をアップロードしてください")

uploaded = files.upload()

import pandas as pd

# ステップ別データの読み込み

stepwise\_file = list(uploaded.keys())[0]

xls = pd.ExcelFile(stepwise\_file)

# シートごとにTechnique\_ID × Asset\_Nameのペアを抽出

step\_data = {}

for sheet in xls.sheet\_names:

df = xls.parse(sheet)

df = df.iloc[:, :2] # 最初の2列のみ使う

df.columns = ["Technique\_ID", "Asset\_Name"]

step\_data[sheet] = df

# 手動アップロード（valid\_communication\_routes.csv）

print(" 次に『valid\_communication\_routes.csv』をアップロードしてください")

uploaded = files.upload()

routes\_file = list(uploaded.keys())[0]

routes\_df = pd.read\_csv(routes\_file)

# 全組合せ（25,088通り）を生成

import itertools

step1 = step\_data["Step1"]

step2 = step\_data["Step2"]

step3 = step\_data["Step3"]

step4 = step\_data["Step4"]

combinations = list(itertools.product(

step1.itertuples(index=False),

step2.itertuples(index=False),

step3.itertuples(index=False),

step4.itertuples(index=False)

))

all\_paths = []

for a, b, c, d in combinations:

all\_paths.append({

"Step1\_TechniqueID": a.Technique\_ID,

"Step1\_Asset": a.Asset\_Name,

"Step2\_TechniqueID": b.Technique\_ID,

"Step2\_Asset": b.Asset\_Name,

"Step3\_TechniqueID": c.Technique\_ID,

"Step3\_Asset": c.Asset\_Name,

"Step4\_TechniqueID": d.Technique\_ID,

"Step4\_Asset": d.Asset\_Name,

})

all\_paths\_df = pd.DataFrame(all\_paths)

all\_paths\_df.to\_csv("all\_attack\_paths\_25088.csv", index=False, encoding="utf-8-sig")

# 通信ルートに合致するもののみ抽出

valid\_set = set(

tuple(x) for x in routes\_df[[

"Step1\_Asset", "Step2\_Asset", "Step3\_Asset", "Step4\_Asset"

]].values

)

filtered\_df = all\_paths\_df[

all\_paths\_df[["Step1\_Asset", "Step2\_Asset", "Step3\_Asset", "Step4\_Asset"]]

.apply(tuple, axis=1)

.isin(valid\_set)

]

filtered\_df.to\_csv("filtered\_attack\_paths\_256.csv", index=False, encoding="utf-8-sig")

# 生成ファイルをダウンロード

from google.colab import files

print("📥 生成されたファイルをダウンロードします")

files.download("all\_attack\_paths\_25088.csv")

files.download("filtered\_attack\_paths\_256.csv")