### **Python Code 一覧**

#### **6/4用（ZIP展開＋70段スキャン雛形）**

# =====================================================================

# 1) ZIP展開＋70段スキャン＋TamperSuspect＋日付混在＋集計

# =====================================================================

import os, re, zipfile, json, hashlib

from pathlib import Path

import pandas as pd

from datetime import datetime

# 出力ディレクトリ

outdir = Path("/mnt/data/KABUKI\_INV\_2025-06-01\_outputs")

outdir.mkdir(exist\_ok=True)

def sha256\_file(path):

h = hashlib.sha256()

with open(path, "rb") as f:

for chunk in iter(lambda: f.read(8192), b""):

h.update(chunk)

return h.hexdigest()

def extract\_zip\_to\_dir(zip\_path, extract\_to):

with zipfile.ZipFile(zip\_path, "r") as z:

z.extractall

**2. Template-3（6/4事案、6段式出力）**

# 必要なライブラリを再インポートして再実行（state reset対応）

from pathlib import Path

import pandas as pd

from docx import Document

from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer

from reportlab.lib.styles import getSampleStyleSheet

import zipfile

# 出力ディレクトリ

out\_dir = Path("/mnt/data/TEMPLATE3\_2025-06-04\_6STEP")

out\_dir.mkdir(exist\_ok=True)

# 共通データ（テンプレ3エントリ）

data = {

"date\_utc7": "2025-06-04 22:19",

"device": "iPhone 11 Pro (iPhone12,3)",

"event\_type": "強制stackshot（bug\_type 288）",

"event\_detail": "RTBuddyService / AppleSPU 同時稼働、Unicode改ざん痕跡あり",

"log\_ref": "bug\_type\_288-2025\_0604\_221905.docx; Text-06-bug-type-288-2025-0604-221905.docx",

"ref\_diff": "EVENTS\_TR-2025-06-04\_SCAN70\_FULL.csv; TAMPER\_JP\_TR-2025-06-04\_SCAN70.csv; DATE\_MAP\_TR-2025-06-04\_SCAN70.csv",

"tamper\_suspect": "187件（Unicode「認証」「設定」「監視」）",

"mixed\_date\_hits": "7件",

"top\_keywords\_FULL": "RTCR=521, triald=417, JetsamEvent=392, 認証=187, Viettel=143",

"top\_keywords\_CLEAN": "triald=412, 認証=187, RTCR=301, JetsamEvent=289, OKX=102",

"impact": "端末がフリーズし強制stackshot発生。入力妨害、セッション中断、認証改ざん疑惑。",

"severity": "Critical (4)",

"confidence": "0.93",

"location": "ホーチミン市 自宅",

"net\_context": "SSID=VNPT-Home, MCC=452, MNC=04, RAT=LTE",

"ledger\_no": "6",

"custody\_capture": "sha256(bug\_type288原本)",

"custody\_analysis": "sha256(EVENTS\_FULL解析CSV)",

"notes": "主体性ZIP part1/2/3にて一括走査。Tamper・日付混在を検出済。",

"flame\_flag": "Apple (Yes) / VN-Telco (Yes)"

}

# 1. CSV①

csv1\_path = out\_dir / "Template3\_2025-06-04\_entry.csv"

pd.DataFrame([data]).to\_csv(csv1\_path, index=False)

# 2. TXT

txt\_path = out\_dir / "Template3\_2025-06-04\_entry.txt"

with open(txt\_path, "w", encoding="utf-8") as f:

for k, v in data.items():

f.write(f"{k}: {v}\n")

# 3. CSV②（影響マッピング用）

csv2\_path = out\_dir / "Template3\_2025-06-04\_impact.csv"

pd.DataFrame([{

"date": data["date\_utc7"],

"impact": data["impact"],

"severity": data["severity"],

"confidence": data["confidence"],

"location": data["location"],

"net\_context": data["net\_context"],

"flame\_flag": data["flame\_flag"]

}]).to\_csv(csv2\_path, index=False)

# 4. Document (docx)

docx\_path = out\_dir / "Template3\_2025-06-04\_entry.docx"

doc = Document()

doc.add\_heading("Template-3 被害記録エントリ (2025-06-04)", 0)

for k, v in data.items():

doc.add\_paragraph(f"{k}: {v}")

doc.save(docx\_path)

# 5. PDF

pdf\_path = out\_dir / "Template3\_2025-06-04\_entry.pdf"

styles = getSampleStyleSheet()

story = [Paragraph("Template-3 被害記録エントリ (2025-06-04)", styles['Heading1']), Spacer(1, 12)]

for k, v in data.items():

story.append(Paragraph(f"<b>{k}</b>: {v}", styles['Normal']))

story.append(Spacer(1, 6))

SimpleDocTemplate(str(pdf\_path)).build(story)

# 6. ZIP

zip\_path = "/mnt/data/TEMPLATE3\_2025-06-04\_6STEP\_FULL.zip"

with zipfile.ZipFile(zip\_path, 'w', zipfile.ZIP\_DEFLATED) as zf:

zf.write(csv1\_path, arcname=csv1\_path.name)

zf.write(txt\_path, arcname=txt\_path.name)

zf.write(csv2\_path, arcname=csv2\_path.name)

zf.write(docx\_path, arcname=docx\_path.name)

zf.write(pdf\_path, arcname=pdf\_path.name)

zip\_path

**3. Template-4（クローズ＋総括統合 v1.1）**

import pandas as pd

from pathlib import Path

from docx import Document

from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer

from reportlab.lib.styles import getSampleStyleSheet

import zipfile

# 出力ディレクトリ

out\_dir = Path("/mnt/data/TEMPLATE4\_2025-06-04")

out\_dir.mkdir(exist\_ok=True)

# Template-4 記載内容（dict化）

template4\_data = {

"Case-ID": "KABUKI-INV",

"Maintainer": "Tajima",

"Reviewer": "GPT-5",

"date": "2025-06-04",

"device": "iPhone 11 Pro",

"log\_count": "352 (主体性ZIP part1/2/3 含む)",

"phase": "S3",

"custody": "sha256(bug\_type288原本), sha256(EVENTS\_FULL解析CSV)",

"summary": "RTCR=521, triald=417, JetsamEvent=392, 認証=187, Viettel=143",

"impact": "端末フリーズ・stackshot強制・入力妨害・Tamper痕跡あり",

"severity": "Critical (4)",

"confidence": "0.93",

"location": "ホーチミン市 自宅",

"net\_context": "SSID=VNPT-Home, MCC=452, MNC=04, RAT=LTE"

}

# 1. CSV

csv\_path = out\_dir / "Template4\_2025-06-04.csv"

pd.DataFrame([template4\_data]).to\_csv(csv\_path, index=False)

# 2. TXT

txt\_path = out\_dir / "Template4\_2025-06-04.txt"

with open(txt\_path, "w", encoding="utf-8") as f:

for k, v in template4\_data.items():

f.write(f"{k}: {v}\n")

# 3. JSON

json\_path = out\_dir / "Template4\_2025-06-04.json"

pd.DataFrame([template4\_data]).to\_json(json\_path, orient="records", force\_ascii=False, indent=2)

# 4. DOCX

docx\_path = out\_dir / "Template4\_2025-06-04.docx"

doc = Document()

doc.add\_heading("Template-4 クローズ＋総括統合 (2025-06-04)", 0)

for k, v in template4\_data.items():

doc.add\_paragraph(f"{k}: {v}")

doc.save(docx\_path)

# 5. PDF

pdf\_path = out\_dir / "Template4\_2025-06-04.pdf"

styles = getSampleStyleSheet()

story = [Paragraph("Template-4 クローズ＋総括統合 (2025-06-04)", styles['Heading1']), Spacer(1, 12)]

for k, v in template4\_data.items():

story.append(Paragraph(f"<b>{k}</b>: {v}", styles['Normal']))

story.append(Spacer(1, 6))

SimpleDocTemplate(str(pdf\_path)).build(story)

# 6. ZIP

zip\_path = "/mnt/data/TEMPLATE4\_2025-06-04\_FULL.zip"

with zipfile.ZipFile(zip\_path, 'w', zipfile.ZIP\_DEFLATED) as zf:

zf.write(csv\_path, arcname=csv\_path.name)

zf.write(txt\_path, arcname=txt\_path.name)

zf.write(json\_path, arcname=json\_path.name)

zf.write(docx\_path, arcname=docx\_path.name)

zf.write(pdf\_path, arcname=pdf\_path.name)

zip\_path