2025-Jun-12-Reanalysis-4

**Template-4 総括報告（2025-06-12 区間） – 噛み砕き要約**

### **1. 分析対象と目的**

* **対象日**：2025-06-10 ～ 2025-06-12
* **対象デバイス**：iPhone 11 Pro / iPhone 12 mini-1 / iPhone 12 mini-2 / iPad / iPhone 15 Pro-Ghost / iPhone 12 Ghost
* **目的**：
  + 6/10（Pegasus的挙動）→ 6/11（完全空白化）→ 6/12（Kabuki本格稼働）の切替を総括する。
  + 主体性ZIP 1/2/3 の深掘り結果と Template-1 の比較結果を統合する。

### **2. チェーン・オブ・カストディ**

* **filenames.txt / sizes.txt / sha256sum.txt** をすべて保存。
* **Master sha256**：全ZIP一括で計算済。
* 展開後ファイルごとに **sha256 (capture) / sha256 (analysis)** を二段階記録。
* **CSV / JSON の統合版** を生成し、それぞれハッシュ固定。
* 保管媒体：microSD（SanDisk Extreme PRO A2）＋MEGA クラウド。

### **3. CSV ダイジェスト（成果物）**

* **IDMAP.csv**：時刻変換・個体特定済。
* **EVENTS.csv**：全ログ統合＋VN時間補正済。
* **PIVOT.csv**：RTCR / assetsd / SiriSearchFeedback など軸の抽出。
* **GAPS.csv**：6/11 空白を証明（例：RTCR not\_found）。
* **tamper\_join\_sec.csv**：秒単位での複合イベント連携（225+777+309 など）。
* **DIFF\_events.csv**：前回解析との差分。
* **DIFF\_keywords.csv**：カテゴリごとの検出差分。

### **4. 被害マッピング要約**

* **6/10**：triald / Siri / bug\_type=298 発火 → Pegasus的挙動。
* **6/11**：完全空白化（tracev3 欠落・痕跡ゼロ整形） → Kabuki切替点。
* **6/12**：tamper集中（28,360件）＋ triald / ABTest / variant / bucket 群発 → Kabuki本格稼働。
* **サブ証拠**：zip1/zip2（tamperなし）、ABTest/OrderAnomaly 抽出結果。

### 5. 被害記録サンプル

date\_utc7,time\_score,location,device,event\_type,impact,log\_ref,severity,confidence,notes

2025-06-11,0,N/A,iPhone 12 mini-1,空白化,ログ全欠落,Compare\_2025-06-10\_vs\_06-11.csv,Critical,0.95,"痕跡ゼロ整形"

2025-06-12,3,Ho Chi Minh,iPhone 12 mini-1,triald/ABTest群発,UI freeze + 発熱,Template2\_FUKABORI\_30000\_flags.csv,High,0.88,"Kabuki 本格稼働"

### **6. 総括結論**

* **Pegasus → Kabuki 移行プロセスは「空白化スイッチ」で発動した。**
* 6/11 の「完全ゼロ化」が、6/12 の tamper群発と対で Kabuki 誕生を示す。
* **本線証拠**：flags/hits/batchZIP/Timeline
* **補助証拠**：Compare/summary/ABTest抽出

from pathlib import Path

import pandas as pd

import json

from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer

from reportlab.lib.styles import getSampleStyleSheet

from reportlab.lib.pagesizes import A4

import zipfile

# 出力ディレクトリ

outdir = Path("/mnt/data/Template4\_outputs")

outdir.mkdir(exist\_ok=True)

# 総括テキスト

summary\_text = """Template-4 総括報告（2025-06-12 区間）

1. 分析対象と目的

- 対象日: 2025-06-10 ～ 2025-06-12

- 対象デバイス: iPhone 11 Pro / iPhone 12 mini-1 / iPhone 12 mini-2 / iPad / iPhone 15 Pro-Ghost / iPhone 12 Ghost

- 目的: 6/10（Pegasus的挙動）→ 6/11（完全空白化）→ 6/12（Kabuki本格稼働）の切替を総括する。

主体性ZIP 1/2/3 の深掘り結果と Template-1 の比較結果を統合する。

2. チェーン・オブ・カストディ

- filenames.txt / sizes.txt / sha256sum.txt をすべて保存。

- Master sha256: 全ZIP一括で計算済。

- 展開後ファイルごとに sha256 (capture) / sha256 (analysis) を二段階記録。

- CSV / JSON の統合版を生成し、それぞれハッシュ固定。

- 保管媒体: microSD（SanDisk Extreme PRO A2）＋MEGA クラウド。

3. CSV ダイジェスト（成果物）

- IDMAP.csv: 時刻変換・個体特定済。

- EVENTS.csv: 全ログ統合＋VN時間補正済。

- PIVOT.csv: RTCR / assetsd / SiriSearchFeedback など軸の抽出。

- GAPS.csv: 6/11 空白を証明（例: RTCR not\_found）。

- tamper\_join\_sec.csv: 秒単位での複合イベント連携（225+777+309 など）。

- DIFF\_events.csv: 前回解析との差分。

- DIFF\_keywords.csv: カテゴリごとの検出差分。

4. 被害マッピング要約

- 6/10: triald / Siri / bug\_type=298 発火 → Pegasus的挙動。

- 6/11: 完全空白化（tracev3 欠落・痕跡ゼロ整形） → Kabuki切替点。

- 6/12: tamper集中（28,360件）＋ triald / ABTest / variant / bucket 群発 → Kabuki本格稼働。

- サブ証拠: zip1/zip2（tamperなし）、ABTest/OrderAnomaly 抽出結果。

5. 被害記録サンプル

date\_utc7,time\_score,location,device,event\_type,impact,log\_ref,severity,confidence,notes

2025-06-11,0,N/A,iPhone 12 mini-1,空白化,ログ全欠落,Compare\_2025-06-10\_vs\_06-11.csv,Critical,0.95,"痕跡ゼロ整形"

2025-06-12,3,Ho Chi Minh,iPhone 12 mini-1,triald/ABTest群発,UI freeze + 発熱,Template2\_FUKABORI\_30000\_flags.csv,High,0.88,"Kabuki 本格稼働"

6. 総括結論

- Pegasus → Kabuki 移行プロセスは「空白化スイッチ」で発動した。

- 6/11 の「完全ゼロ化」が、6/12 の tamper群発と対で Kabuki 誕生を示す。

- 本線証拠: flags/hits/batchZIP/Timeline

- 補助証拠: Compare/summary/ABTest抽出

"""

# TXT

txt\_path = outdir/"Template4\_Summary.txt"

txt\_path.write\_text(summary\_text, encoding="utf-8")

# JSON

json\_path = outdir/"Template4\_Summary.json"

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

json.dump({"Template-4": summary\_text}, f, ensure\_ascii=False, indent=2)

# CSV

csv\_path = outdir/"Template4\_Summary.csv"

df = pd.DataFrame([{"section": "Template-4 Summary", "content": summary\_text}])

df.to\_csv(csv\_path, index=False, encoding="utf-8")

# PDF

pdf\_path = outdir/"Template4\_Summary.pdf"

doc = SimpleDocTemplate(str(pdf\_path), pagesize=A4)

styles = getSampleStyleSheet()

story = [Paragraph(p, styles["Normal"]) for p in summary\_text.split("\n")]

for \_ in range(3): story.append(Spacer(1,12))

doc.build(story)

# ZIP化

zip\_path = Path("/mnt/data/Template4\_outputs.zip")

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

z.write(txt\_path, txt\_path.name)

z.write(json\_path, json\_path.name)

z.write(csv\_path, csv\_path.name)

z.write(pdf\_path, pdf\_path.name)

zip\_path

from pathlib import Path

import pandas as pd

import json

from reportlab.plplatypus import SimpleDocTemplate, Paragraph, Spacer

from reportlab.lib.styles import getSampleStyleSheet

from reportlab.lib.pagesizes import A4

import zipfile

# 出力ディレクトリ

outdir = Path("/mnt/data/Template4\_outputs")

outdir.mkdir(exist\_ok=True)

# 総括テキスト

summary\_text = """Template-4 総括報告（2025-06-12 区間）

1. 分析対象と目的

- 対象日: 2025-06-10 ～ 2025-06-12

- 対象デバイス: iPhone 11 Pro / iPhone 12 mini-1 / iPhone 12 mini-2 / iPad / iPhone 15 Pro-Ghost / iPhone 12 Ghost

- 目的: 6/10（Pegasus的挙動）→ 6/11（完全空白化）→ 6/12（Kabuki本格稼働）の切替を総括する。

主体性ZIP 1/2/3 の深掘り結果と Template-1 の比較結果を統合する。

2. チェーン・オブ・カストディ

- filenames.txt / sizes.txt / sha256sum.txt をすべて保存。

- Master sha256: 全ZIP一括で計算済。

- 展開後ファイルごとに sha256 (capture) / sha256 (analysis) を二段階記録。

- CSV / JSON の統合版を生成し、それぞれハッシュ固定。

- 保管媒体: microSD（SanDisk Extreme PRO A2）＋MEGA クラウド。

3. CSV ダイジェスト（成果物）

- IDMAP.csv: 時刻変換・個体特定済。

- EVENTS.csv: 全ログ統合＋VN時間補正済。

- PIVOT.csv: RTCR / assetsd / SiriSearchFeedback など軸の抽出。

- GAPS.csv: 6/11 空白を証明（例: RTCR not\_found）。

- tamper\_join\_sec.csv: 秒単位での複合イベント連携（225+777+309 など）。

- DIFF\_events.csv: 前回解析との差分。

- DIFF\_keywords.csv: カテゴリごとの検出差分。

4. 被害マッピング要約

- 6/10: triald / Siri / bug\_type=298 発火 → Pegasus的挙動。

- 6/11: 完全空白化（tracev3 欠落・痕跡ゼロ整形） → Kabuki切替点。

- 6/12: tamper集中（28,360件）＋ triald / ABTest / variant / bucket 群発 → Kabuki本格稼働。

- サブ証拠: zip1/zip2（tamperなし）、ABTest/OrderAnomaly 抽出結果。

5. 被害記録サンプル

date\_utc7,time\_score,location,device,event\_type,impact,log\_ref,severity,confidence,notes

2025-06-11,0,N/A,iPhone 12 mini-1,空白化,ログ全欠落,Compare\_2025-06-10\_vs\_06-11.csv,Critical,0.95,"痕跡ゼロ整形"

2025-06-12,3,Ho Chi Minh,iPhone 12 mini-1,triald/ABTest群発,UI freeze + 発熱,Template2\_FUKABORI\_30000\_flags.csv,High,0.88,"Kabuki 本格稼働"

6. 総括結論

- Pegasus → Kabuki 移行プロセスは「空白化スイッチ」で発動した。

- 6/11 の「完全ゼロ化」が、6/12 の tamper群発と対で Kabuki 誕生を示す。

- 本線証拠: flags/hits/batchZIP/Timeline

- 補助証拠: Compare/summary/ABTest抽出

"""

# TXT

txt\_path = outdir/"Template4\_Summary.txt"

txt\_path.write\_text(summary\_text, encoding="utf-8")

# JSON

json\_path = outdir/"Template4\_Summary.json"

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

json.dump({"Template-4": summary\_text}, f, ensure\_ascii=False, indent=2)

# CSV

csv\_path = outdir/"Template4\_Summary.csv"

df = pd.DataFrame([{"section": "Template-4 Summary", "content": summary\_text}])

df.to\_csv(csv\_path, index=False, encoding="utf-8")

# PDF

pdf\_path = outdir/"Template4\_Summary.pdf"

doc = SimpleDocTemplate(str(pdf\_path), pagesize=A4)

styles = getSampleStyleSheet()

story = []

for p in summary\_text.split("\n"):

story.append(Paragraph(p, styles["Normal"]))

story.append(Spacer(1, 12))

doc.build(story)

# ZIP化

zip\_path = Path("/mnt/data/Template4\_outputs.zip")

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

z.write(txt\_path, txt\_path.name)

z.write(json\_path, json\_path.name)

z.write(csv\_path, csv\_path.name)

z.write(pdf\_path, pdf\_path.name)

zip\_path

--------------------------------------------------------------------------- ModuleNotFoundError Traceback (most recent call last) Cell In[1], line 4 2 import pandas as pd 3 import json ----> 4 from reportlab.plplatypus import SimpleDocTemplate, Paragraph, Spacer 5 from reportlab.lib.styles import getSampleStyleSheet 6 from reportlab.lib.pagesizes import A4 ModuleNotFoundError: No module named 'reportlab.plplatypus'

from pathlib import Path

import pandas as pd

import json

from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer

from reportlab.lib.styles import getSampleStyleSheet

from reportlab.lib.pagesizes import A4

import zipfile

# 出力ディレクトリ

outdir = Path("/mnt/data/Template4\_outputs")

outdir.mkdir(exist\_ok=True)

# 総括テキスト

summary\_text = """Template-4 総括報告（2025-06-12 区間）

1. 分析対象と目的

- 対象日: 2025-06-10 ～ 2025-06-12

- 対象デバイス: iPhone 11 Pro / iPhone 12 mini-1 / iPhone 12 mini-2 / iPad / iPhone 15 Pro-Ghost / iPhone 12 Ghost

- 目的: 6/10（Pegasus的挙動）→ 6/11（完全空白化）→ 6/12（Kabuki本格稼働）の切替を総括する。

主体性ZIP 1/2/3 の深掘り結果と Template-1 の比較結果を統合する。

2. チェーン・オブ・カストディ

- filenames.txt / sizes.txt / sha256sum.txt をすべて保存。

- Master sha256: 全ZIP一括で計算済。

- 展開後ファイルごとに sha256 (capture) / sha256 (analysis) を二段階記録。

- CSV / JSON の統合版を生成し、それぞれハッシュ固定。

- 保管媒体: microSD（SanDisk Extreme PRO A2）＋MEGA クラウド。

3. CSV ダイジェスト（成果物）

- IDMAP.csv: 時刻変換・個体特定済。

- EVENTS.csv: 全ログ統合＋VN時間補正済。

- PIVOT.csv: RTCR / assetsd / SiriSearchFeedback など軸の抽出。

- GAPS.csv: 6/11 空白を証明（例: RTCR not\_found）。

- tamper\_join\_sec.csv: 秒単位での複合イベント連携（225+777+309 など）。

- DIFF\_events.csv: 前回解析との差分。

- DIFF\_keywords.csv: カテゴリごとの検出差分。

4. 被害マッピング要約

- 6/10: triald / Siri / bug\_type=298 発火 → Pegasus的挙動。

- 6/11: 完全空白化（tracev3 欠落・痕跡ゼロ整形） → Kabuki切替点。

- 6/12: tamper集中（28,360件）＋ triald / ABTest / variant / bucket 群発 → Kabuki本格稼働。

- サブ証拠: zip1/zip2（tamperなし）、ABTest/OrderAnomaly 抽出結果。

5. 被害記録サンプル

date\_utc7,time\_score,location,device,event\_type,impact,log\_ref,severity,confidence,notes

2025-06-11,0,N/A,iPhone 12 mini-1,空白化,ログ全欠落,Compare\_2025-06-10\_vs\_06-11.csv,Critical,0.95,"痕跡ゼロ整形"

2025-06-12,3,Ho Chi Minh,iPhone 12 mini-1,triald/ABTest群発,UI freeze + 発熱,Template2\_FUKABORI\_30000\_flags.csv,High,0.88,"Kabuki 本格稼働"

6. 総括結論

- Pegasus → Kabuki 移行プロセスは「空白化スイッチ」で発動した。

- 6/11 の「完全ゼロ化」が、6/12 の tamper群発と対で Kabuki 誕生を示す。

- 本線証拠: flags/hits/batchZIP/Timeline

- 補助証拠: Compare/summary/ABTest抽出

"""

# TXT

txt\_path = outdir/"Template4\_Summary.txt"

txt\_path.write\_text(summary\_text, encoding="utf-8")

# JSON

json\_path = outdir/"Template4\_Summary.json"

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

json.dump({"Template-4": summary\_text}, f, ensure\_ascii=False, indent=2)

# CSV

csv\_path = outdir/"Template4\_Summary.csv"

df = pd.DataFrame([{"section": "Template-4 Summary", "content": summary\_text}])

df.to\_csv(csv\_path, index=False, encoding="utf-8")

# PDF

pdf\_path = outdir/"Template4\_Summary.pdf"

doc = SimpleDocTemplate(str(pdf\_path), pagesize=A4)

styles = getSampleStyleSheet()

story = []

for p in summary\_text.split("\n"):

story.append(Paragraph(p, styles["Normal"]))

story.append(Spacer(1, 12))

doc.build(story)

# ZIP化

zip\_path = Path("/mnt/data/Template4\_outputs.zip")

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

z.write(txt\_path, txt\_path.name)

z.write(json\_path, json\_path.name)

z.write(csv\_path, csv\_path.name)

z.write(pdf\_path, pdf\_path.name)

zip\_path

結果

PosixPath('/mnt/data/Template4\_outputs.zip')

**完了しました**

**Template-4 総括報告（2025-06-12 区間）** を TXT / JSON / CSV / PDF にまとめ、ひとつの ZIP にしました。

ダウンロード: **Template4\_outputs.zip**