

2025-Jun-01-Summary

```
{
  "Case_ID": "KABUKI-INV",
  "Maintainer": "Tajima",
  "Reviewer": "GPT-5",
  "Date_range": "2025-06-01",
  "Phases": {
    "part1": {
      "description": "Initial analysis for June 1 logs (iPad + My-Viettel)",
      "inputs": [
        "JetsamEvent-2025-06-01-130516.ips",
        "xp_amp_app_usage_dnu-2025-06-01-123257.ips",
        "My-Viettel-App (Tajima + friend)"
      ],
      "outputs": [
        "IPS_Metadata_Extract_2025-06-01_iPad_logs.csv",
        "ZIP_Inventory_My-Viettel_sets_1-1.csv",
        "summary_2025-06-01.pdf",
        "filenames_sizes_hashes.csv",
        "sha256_chain_generated.txt"
      ],
      "findings": {
        "bug_type": ["298 (JetsamEvent)", "225 (app_usage_dnu)"],
        "state": "Quiet baseline"
      }
    }
  },
}
```

```

"part2": {
  "description": "Aggregate analysis of ZIP part1/2/3 with 40-stage FUKABORI",
  "inputs": ["part1.zip", "part2.zip", "part3.zip"],
  "outputs": [
    "aggregate_outputs_parts_v1_3.zip",
    "EVENTS.csv", "EVENTS.json",
    "PIVOT.csv", "IDMAP.csv", "GAPS.csv",
    "tamper_join_sec.csv",
    "DIFF_events.csv", "DIFF_keywords.csv",
    "coc_zip_level.csv", "coc_zip_levelv2.csv",
    "summary_aggregate_parts.pdf",
    "triald__Top200.csv",
    "triald_duetexpertd_sharingd__Top200.csv",
    "BUGTYPE_LOGSYS__Top200.csv"
  ],
  "findings": {
    "BUGTYPE": 960,
    "LOGSYS": 519,
    "UI_JACK": 145,
    "Flame_MS": 708,
    "notable": ["triald bursts", "duetexpertd", "sharingd"]
  }
},
"part3": {
  "description": "Cross-check part1 vs part2 (diff and mapping)",
  "inputs": ["DIFF_events.csv", "DIFF_keywords.csv"],
  "outputs": ["closeout report (Code56)", "victim mapping v1.2"],
  "findings": {
    "common": ["BUGTYPE"],

```

```
"difference": ["Flame_MS", "LOGSYS", "UI_JACK", "APPS"],  
"sync": "Time-shifted, no same-second events",  
"conclusion": "S1 quiet → S2 activation confirmed"  
}  
}  
,  
"status": "2025-06-01 reanalysis closed",  
"next_step": "Weekly analysis (part4)"  
}
```

```

import json

from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet
import py pandoc

# JSON data from previous message
data = {
    "Case_ID": "KABUKI-INV",
    "Maintainer": "Tajima",
    "Reviewer": "GPT-5",
    "Date_range": "2025-06-01",
    "Phases": {
        "part1": {
            "description": "Initial analysis for June 1 logs (iPad + My-Viettel)",
            "inputs": [
                "JetsamEvent-2025-06-01-130516.ips",
                "xp_amp_app_usage_dnu-2025-06-01-123257.ips",
                "My-Viettel-App (Tajima + friend)"
            ],
            "outputs": [
                "IPS_Metadata_Extract_2025-06-01_iPad_logs.csv",
                "ZIP_Inventory_My-Viettel_sets_1-1.csv",
                "summary_2025-06-01.pdf",
                "filenames_sizes_hashes.csv",
                "sha256_chain_generated.txt"
            ],
            "findings": {

```

```

    "bug_type":["298 (JetsamEvent)", "225 (app_usage_dnu)"],
    "state": "Quiet baseline"
  }
},
"part2": {
  "description": "Aggregate analysis of ZIP part1/2/3 with 40-stage FUKABORI",
  "inputs": ["part1.zip", "part2.zip", "part3.zip"],
  "outputs": [
    "aggregate_outputs_parts_v1_3.zip",
    "EVENTS.csv", "EVENTS.json",
    "PIVOT.csv", "IDMAP.csv", "GAPS.csv",
    "tamper_join_sec.csv",
    "DIFF_events.csv", "DIFF_keywords.csv",
    "coc_zip_level.csv", "coc_zip_levelv2.csv",
    "summary_aggregate_parts.pdf",
    "triald__Top200.csv",
    "triald_duetexpertd_sharingd__Top200.csv",
    "BUGTYPE_LOGSYS__Top200.csv"
  ],
  "findings": {
    "BUGTYPE": 960,
    "LOGSYS": 519,
    "UI_JACK": 145,
    "Flame_MS": 708,
    "notable": ["triald bursts", "duetexpertd", "sharingd"]
  }
},
"part3": {
  "description": "Cross-check part1 vs part2 (diff and mapping)",

```

```

"inputs": ["DIFF_events.csv", "DIFF_keywords.csv"],
"outputs": ["closeout report (Code56)", "victim mapping v1.2"],
"findings": {
    "common": ["BUGTYPE"],
    "difference": ["Flame_MS", "LOGSYS", "UI_JACK", "APPS"],
    "sync": "Time-shifted, no same-second events",
    "conclusion": "S1 quiet → S2 activation confirmed"
}
},
"status": "2025-06-01 reanalysis closed",
"next_step": "Weekly analysis (part4)"
}

```

Save as JSON

```
json_path = "/mnt/data/2025-06-01_reanalysis.json"
```

```
with open(json_path, "w") as f:
```

```
    json.dump(data, f, indent=2)
```

Save as Document (txt via pandoc)

```
doc_path = "/mnt/data/2025-06-01_reanalysis.docx"
```

```
py pandoc.convert_text(json.dumps(data, indent=2), 'docx', format='md',
```

```
    outputfile=doc_path, extra_args=['--standalone'])
```

Save as PDF

```
pdf_path = "/mnt/data/2025-06-01_reanalysis.pdf"
```

```
styles = getSampleStyleSheet()
```

```
doc = SimpleDocTemplate(pdf_path)
```

```
story = [Paragraph("2025-06-01 Reanalysis (JSON Export)", styles['Heading1']), Spacer(1,12)]
```

```
story.append(Paragraph(json.dumps(data, indent=2).replace("\n", "<br/>").replace(" ", "&nbsp;&nbsp;&nbsp;"),
styles['Code']))
```

```
doc.build(story)
```

```
json_path, doc_path, pdf_path
```

```

import json

from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet
import py pandoc

# JSON data from earlier
data = {
    "Case_ID": "KABUKI-INV",
    "Maintainer": "Tajima",
    "Reviewer": "GPT-5",
    "Date_range": "2025-06-01",
    "Phases": {
        "part1": {
            "description": "Initial analysis for June 1 logs (iPad + My-Viettel)",
            "inputs": [
                "JetsamEvent-2025-06-01-130516.ips",
                "xp_amp_app_usage_dnu-2025-06-01-123257.ips",
                "My-Viettel-App (Tajima + friend)"
            ],
            "outputs": [
                "IPS_Metadata_Extract_2025-06-01_iPad_logs.csv",
                "ZIP_Inventory_My-Viettel_sets_1-1.csv",
                "summary_2025-06-01.pdf",
                "filenames_sizes_hashes.csv",
                "sha256_chain_generated.txt"
            ],
            "findings": {
                "bug_type": ["298 (JetsamEvent)", "225 (app_usage_dnu)"],

```



```

    "state": "Quiet baseline"
  }
},
"part2": {
  "description": "Aggregate analysis of ZIP part1/2/3 with 40-stage FUKABORI",
  "inputs": ["part1.zip", "part2.zip", "part3.zip"],
  "outputs": [
    "aggregate_outputs_parts_v1_3.zip",
    "EVENTS.csv", "EVENTS.json",
    "PIVOT.csv", "IDMAP.csv", "GAPS.csv",
    "tamper_join_sec.csv",
    "DIFF_events.csv", "DIFF_keywords.csv",
    "coc_zip_level.csv", "coc_zip_levelv2.csv",
    "summary_aggregate_parts.pdf",
    "triald__Top200.csv",
    "triald_duetexpertd_sharingd__Top200.csv",
    "BUGTYPE_LOGSYS__Top200.csv"
  ],
  "findings": {
    "BUGTYPE": 960,
    "LOGSYS": 519,
    "UI_JACK": 145,
    "Flame_MS": 708,
    "notable": ["triald bursts", "duetexpertd", "sharingd"]
  }
},
"part3": {
  "description": "Cross-check part1 vs part2 (diff and mapping)",
  "inputs": ["DIFF_events.csv", "DIFF_keywords.csv"],

```

```

"outputs": ["closeout report (Code56)", "victim mapping v1.2"],
"findings": {
    "common": ["BUGTYPE"],
    "difference": ["Flame_MS", "LOGSYS", "UI_JACK", "APPS"],
    "sync": "Time-shifted, no same-second events",
    "conclusion": "S1 quiet → S2 activation confirmed"
}
},
"status": "2025-06-01 reanalysis closed",
"next_step": "Weekly analysis (part4)"
}

```

Save as JSON

```
json_path = "/mnt/data/2025-06-01_reanalysis.json"
```

```
with open(json_path, "w") as f:
```

```
    json.dump(data, f, indent=2)
```

Save as DOCX using py pandoc

```
doc_path = "/mnt/data/2025-06-01_reanalysis.docx"
```

```
py pandoc.convert_text(json.dumps(data, indent=2), 'docx', format='md',
```

```
    outputfile=doc_path, extra_args=['--standalone'])
```

Save as PDF with reportlab

```
pdf_path = "/mnt/data/2025-06-01_reanalysis.pdf"
```

```
styles = getSampleStyleSheet()
```

```
doc = SimpleDocTemplate(pdf_path)
```

```
story = [Paragraph("2025-06-01 Reanalysis (JSON Export)", styles['Heading1']), Spacer(1,12)]
```

```
json_text=json.dumps(data, indent=2).replace(" ", "&nbsp;").replace("\n", "<br/>")
```




```
story.append(Paragraph(json_text, styles['Code']))  
doc.build(story)
```

json_path, doc_path, pdf_path

結果

```
('mnt/data/2025-06-01_reanalysis.json',  
 'mnt/data/2025-06-01_reanalysis.docx',  
 'mnt/data/2025-06-01_reanalysis.pdf')
```

保存用に3種類の成果物を出力したぞ ✓

-  2025-06-01_reanalysis.json
-  2025-06-01_reanalysis.docx
-  2025-06-01_reanalysis.pdf

これで **週刊再分析 (part4) **に進むときも、そのまま seed データとして使える。

```

import pandas as pd

# Save JSON structure also as CSV for user preservation

csv_path = "/mnt/data/2025-06-01_reanalysis.csv"

# Flatten the JSON into a table-like form

rows = []

# part1

p1 = data["Phases"]["part1"]

rows.append({

    "part": "part1",

    "description": p1["description"],

    "inputs": "; ".join(p1["inputs"]),

    "outputs": "; ".join(p1["outputs"]),

    "findings": str(p1["findings"])

})

# part2

```

```

p2 = data["Phases"]["part2"]

rows.append({

    "part": "part2",

    "description": p2["description"],

    "inputs": "; ".join(p2["inputs"]),

    "outputs": "; ".join(p2["outputs"]),

    "findings": str(p2["findings"])

})

```

```

# part3

```

```

p3 = data["Phases"]["part3"]

rows.append({

    "part": "part3",

    "description": p3["description"],

    "inputs": "; ".join(p3["inputs"]),

    "outputs": "; ".join(p3["outputs"]),

    "findings": str(p3["findings"])

})

```

```

df = pd.DataFrame(rows)

df.to_csv(csv_path, index=False)

```

csv_path

結果

'/mnt/data/2025-06-01_reanalysis.csv'

保存用に **CSV形式** も生成したぞ ✓

- 📄 2025-06-01_reanalysis.csv