

LEAKS

RKI

Der Download der neuen [ungeschwärzten RKI-Dateien](#) gestaltet sich bis heute schwierig. Diese Daten wurde später auch als [Torrent](#) bereitgestellt. Die Webseite <https://rki-transparenzbericht.de/> mit den Daten ist (aktuell) leider nicht erreichbar.

Dank an [Aya Velázquez](#), dem unbekanntem Whistleblower sowie allen weiteren Beteiligten für diesen Leak.

Der einfachste Weg zum Download dürfte aktuell mit einem Torrent Client/Programm, wie z.B. [Transmission](#) sein. Das Programm Transmission steht auch innerhalb der Suite [PortableApps](#) zur Verfügung.

Benötigt wird noch dieses [Torrent File](#), dann kann per Drag & Drop dieses Files in das Transmission Fenster der Download starten.

Expertenrat / Kanzleramt

468 Seiten Original Emailverkehr zwischen Kanzleramt und Expertenrat

https://x.com/aya_velazquez/status/1829782828634010066?t=ZANKWiA6Q3uqZSKjsXolRg&s=19

Israel Exposed Archives / Gaza

<https://watermeloncrimes.com/archives>

see X-Post for some notes: <https://x.com/xIsraelExposed/status/1988735326517444791>

Epstein Files (2026)

<https://www.justice.gov/epstein/doj-disclosures>

Die dort bereitgestellt Archive je Data Set sind leider wieder weg. Scheinbar werden einzelne Dokumente nachträglich noch depubliziert (unpublished)! Für 'Data Set 9' gab es das Archiv nie vollständig bzw. nur korrupt und somit unbrauchbar.

Auf Reddit gibt es Diskussion, wer/wie was bereits heruntergeladen und gesichert hat:

- [Reddit 1](#)
- [Reddit 2](#)

Magnet links und torrents finden sich hier für die schwierigeren Datensätze (data sets).

Zum Auskundschaften aller 'Pages' - mit bis zu 50 Dokument-Links je Page - konnte ich dieses Python-Skript auf [PasteBin-Link](#) ausmachen. Unter Linux - auch in einer *Windows Subsystem for Linux* (WSL) kriege ich es zumindest für die Ermittlung der Dokumente zum Laufen. Es nutzt Playwright und automatisiert den Seitenabruf mit einem Chromium-Browser.

Mit minimaler Anpassung der Pfade lief das - in mehreren Versuchen durch - und ermittelte mir 279'753 Einzeldokumente aus knapp 20'500 Pages. Ergebnis ist die JSON-Datei `dataset9_index.json`.

Für die Nutzung vorher

```
python3 -m pip install --upgrade pip
python3 -m pip install playwright
playwright install
```

Mit dem Download der Einzeldokumente der PDFs hat das Skript ein Problem: Es hängt je PDF Datei!

Mit der Ergänzung der Kommandozeilen-Auswertung (ganz unten) um:

```
elif cmd == "printurls":
    with open(INDEX_FILE, 'r') as f:
        all_files = json.load(f)
    for file_info in all_files:
        url = file_info['url']
        print(f"{url}")
```

und dem Kommandozeilenaufruf

```
./parse\ dataset9.py printurls >urls.txt
```

bekomme ich ein flache Liste der Dokument-URLs, die ich dann mit einem simplen Bash Skript `get_all_urls.sh` unter Nutzung von `wget` herunterlade ..

```
#!/bin/bash

T="$1"
if [ -z "$T" ]; then
    T="1"
fi
mkdir DataSet_9
cd DataSet_9
N=$T

# grep .m4a ../urls.txt | while read LINE; do
# grep .mp4 ../urls.txt | tail -n "+$T" | while read LINE; do
tail -n "+$T" ../urls.txt | while read LINE; do
    echo "$N : $LINE" |tee -a ../get_all.log
    wget --no-verbose -c --header='Cookie: justiceGovAgeVerified=true' "$LINE"
2>&1 |tee -a ../get_all.log
    N=$((N + 1))
```

done

Hinterher sollte ich die Fehler aus dem Log File `get_all.log` ermitteln können.

Für reine einfache Sichtung ist [Jmail](#) sehr zu empfehlen.

Weitere Archive/Links auch hier [DataHoarding Archives](#). Die letzten 2 verlinkten Seiten enthalten Magnet/Torrent Links: [Epstein Data Sets](#) und [Epstein Files](#) mit etwas mehr Erklärung.

From:

<https://codingspirit.de/dokuwiki/> - **coding spirit**

Permanent link:

<https://codingspirit.de/dokuwiki/doku.php?id=social:leaks&rev=1770247595>

Last update: **2026/02/05**

