RANDORISEC

CoRIIN 2024
Apple Sysdiagnose For iOS Forensics

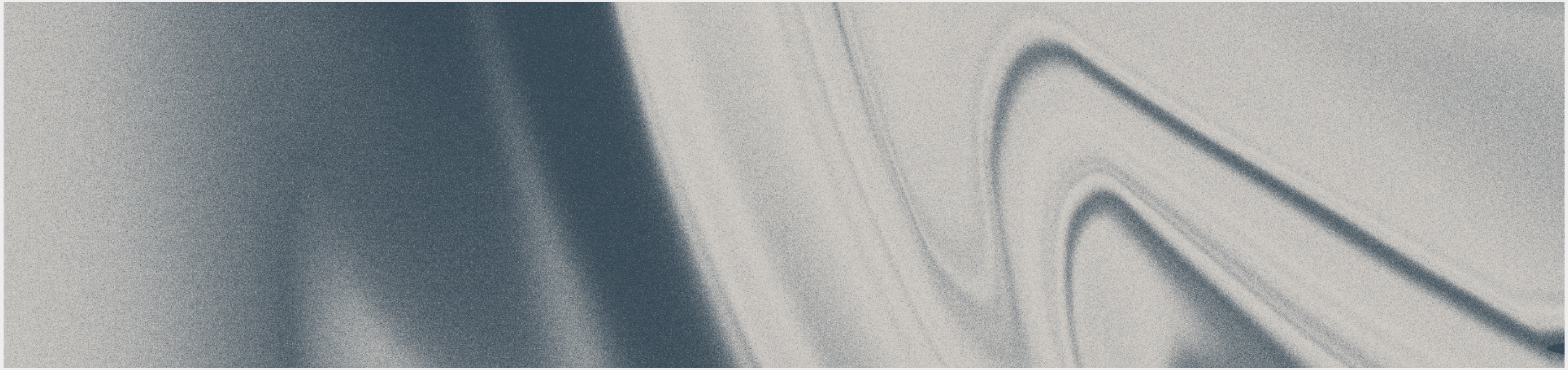
CECyF

Davy Douhine
Founder & Security Consultant

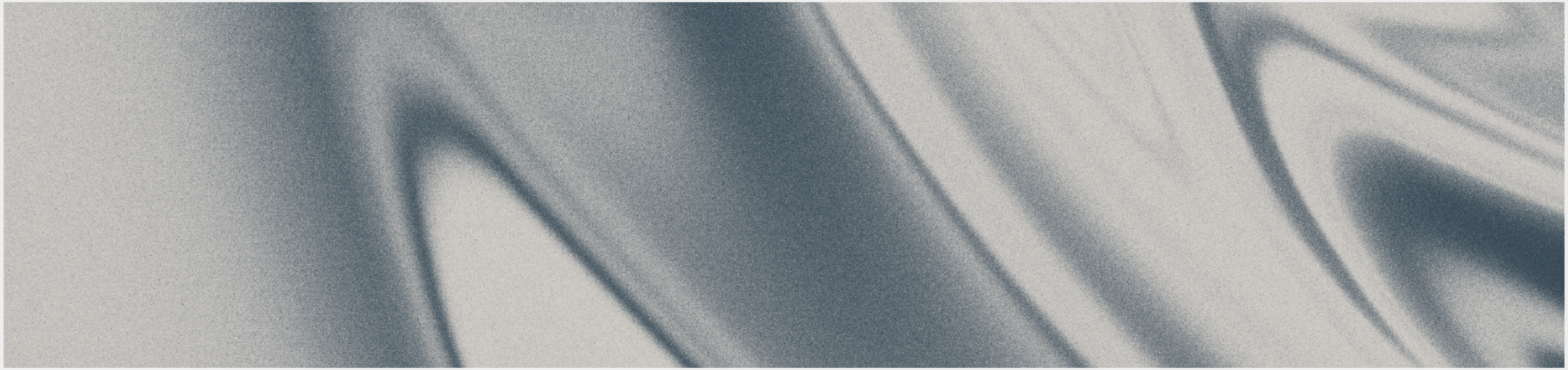
Amel Khamoum
Security Researcher Apprentice

Jérôme Rouaix
Solution Architect

Presentation Plan



I. Problem Statement



The Need for Device Analysis

Our smartphones contain a lot of sensitive data

- Emails and conversations
- Photos and videos

And they have many sensors

- Camera
- Microphone
- GPS

Access to this data and sensors is a serious concern regarding the security and privacy of individuals

Sophisticated Cyber Threats

Sophisticated cyber threats have emerged targeting iOS devices

- Zero-click exploits
- CVEs, kernel exploits, ...

More sensitive roles are being attacked

- Politicians
- Journalists
- Activists
- ...

Necessity for iOS forensics to safeguard the privacy and the security

PREDATOR FILES

NSO GROUP

PEGASUS

Bring Your Own Device

B.Y.O.D

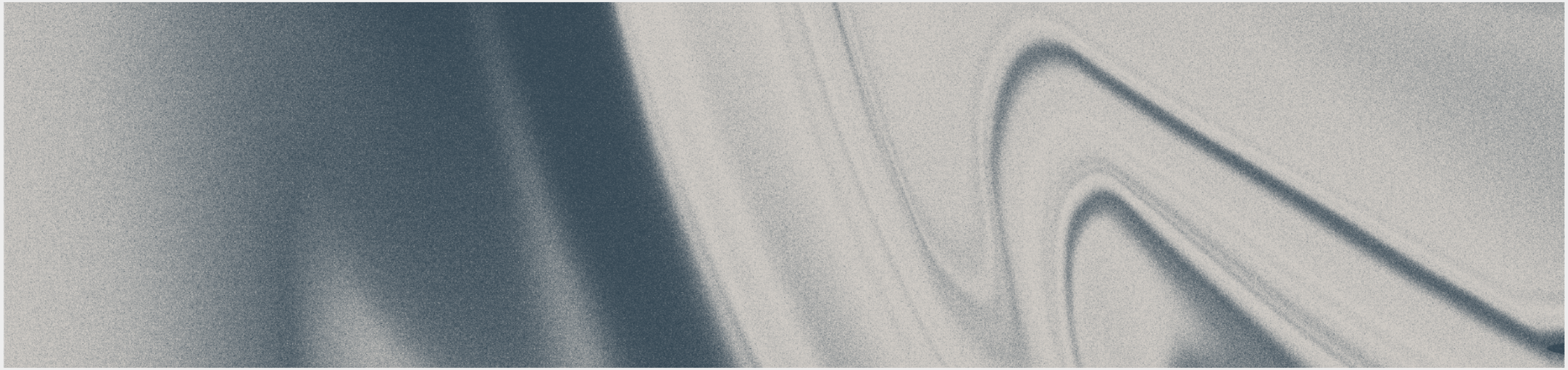
66%

Mobiles belong to employees

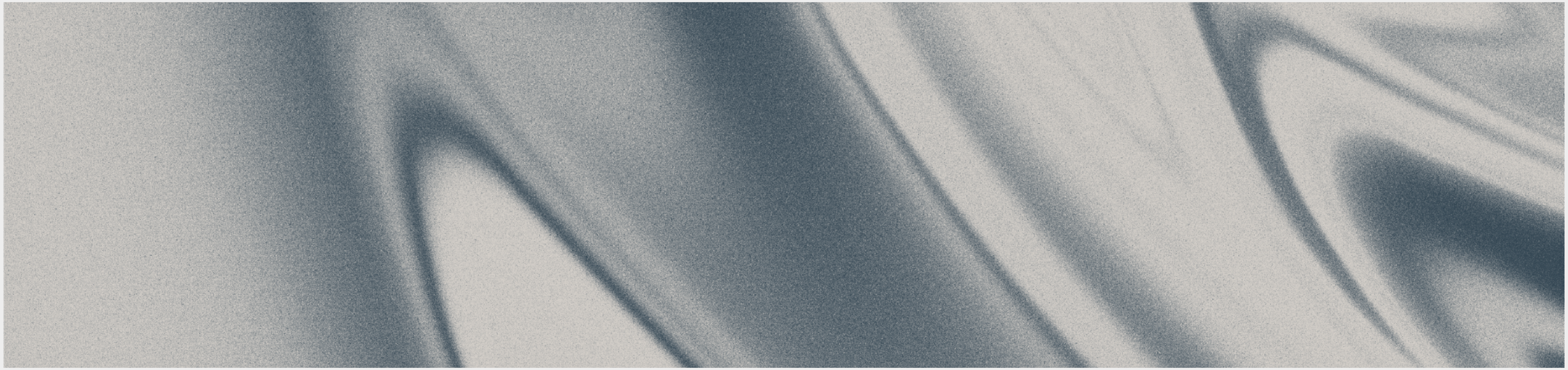
25%

European smartphone user
has already had at least 1 malware

Source: study carried out by Zimperium in 2022 https://www.zimperium.com/global-mobile-threat-report/



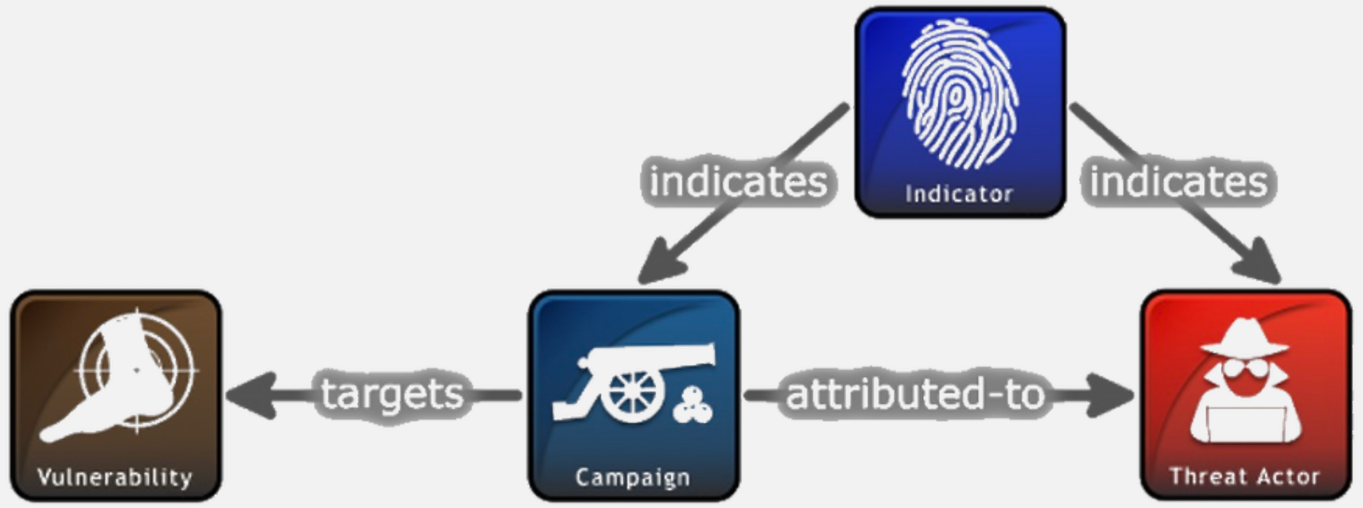
II. iOS Forensics



Indicator of Compromise

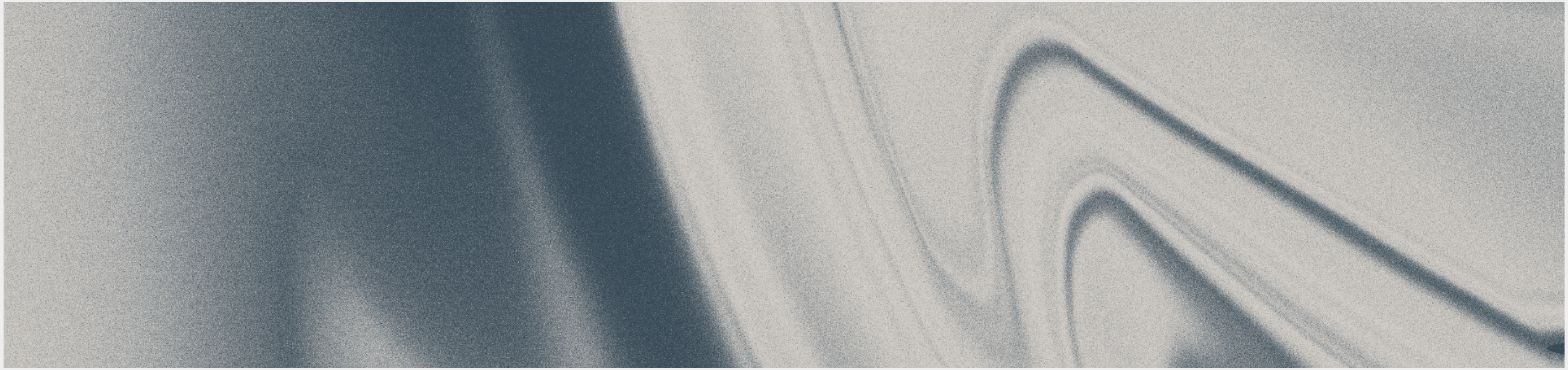
An IOC refers to any piece of information that can be used to detect malicious activity or a security breach

- File traces
- Suspicious processes and URLs
- Binary Hashes
- Network Traffic
- Provisioning profiles
- Trusted certifications

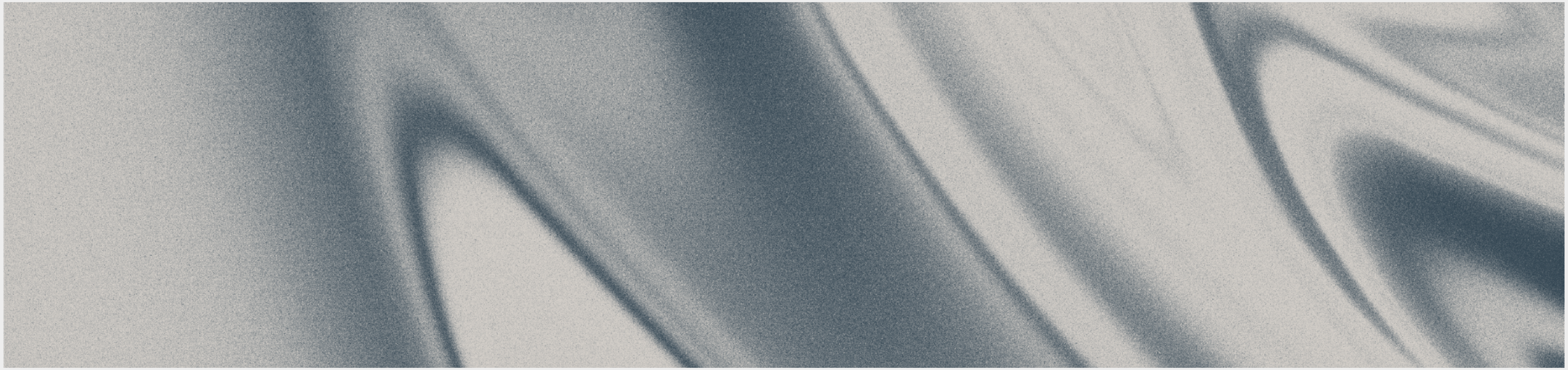


STIX 2 Relationship Example

STIX is a way to describe IoCs and to set them into relation

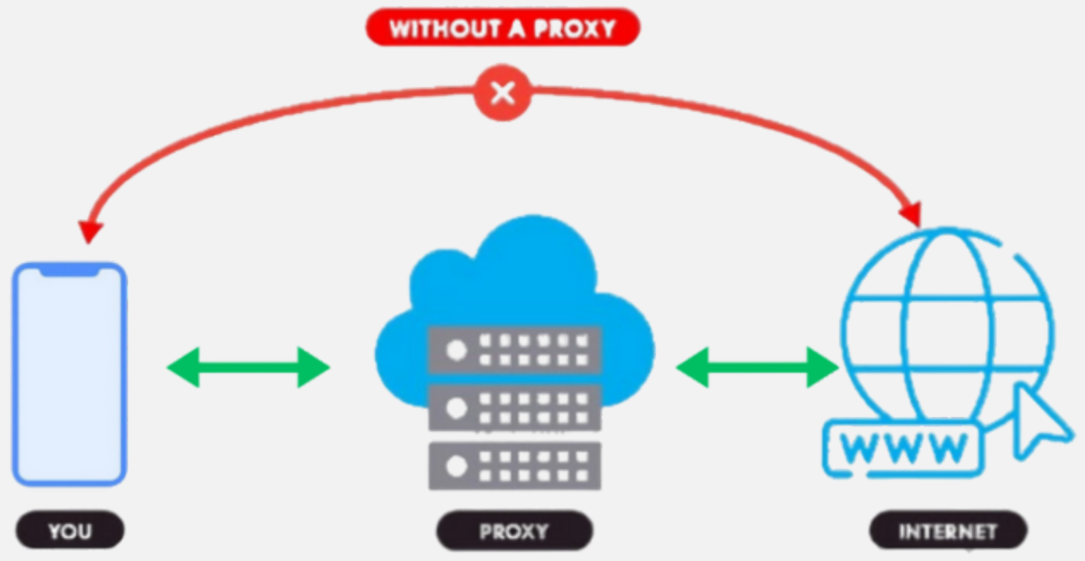


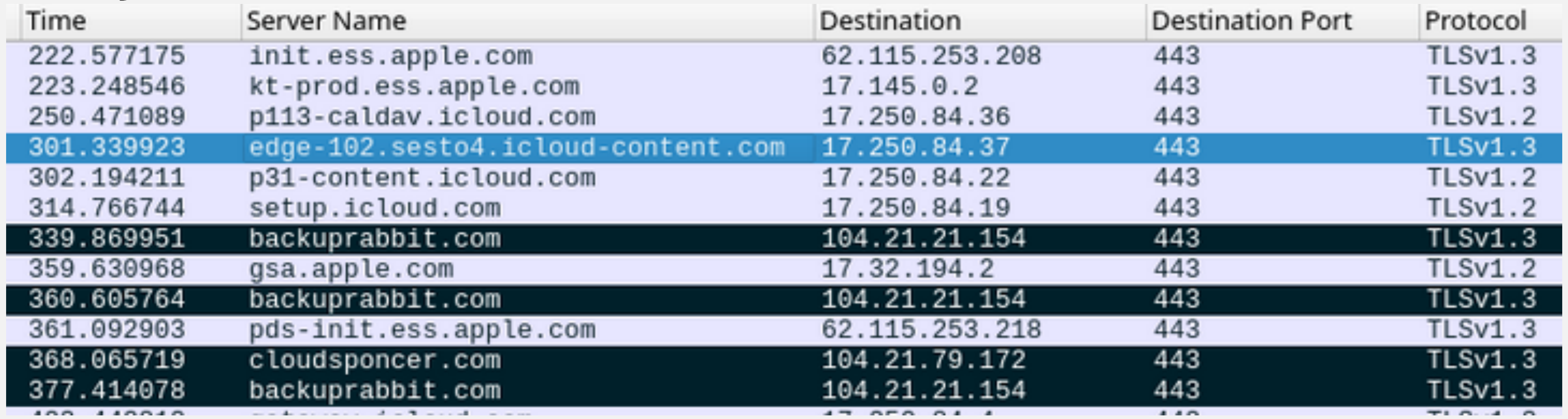
1. Network Traffic Analysis



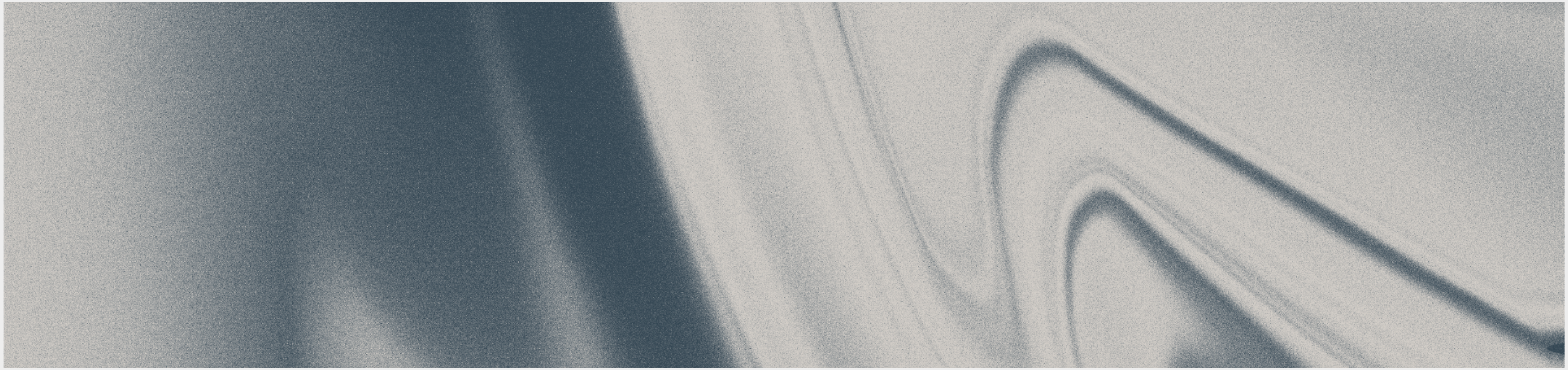
Network Traffic Analysis

- **Analyse connections between iOS devices and external servers**
- **Detection of potentially malicious activities in real time**
    - Malicious domain names
    - Data uploads to C&C servers
- **Used by kaspersky to detect Operation Triangulation**
    - Multiple connections to C&C domains
    - Malicious iMessage attachment

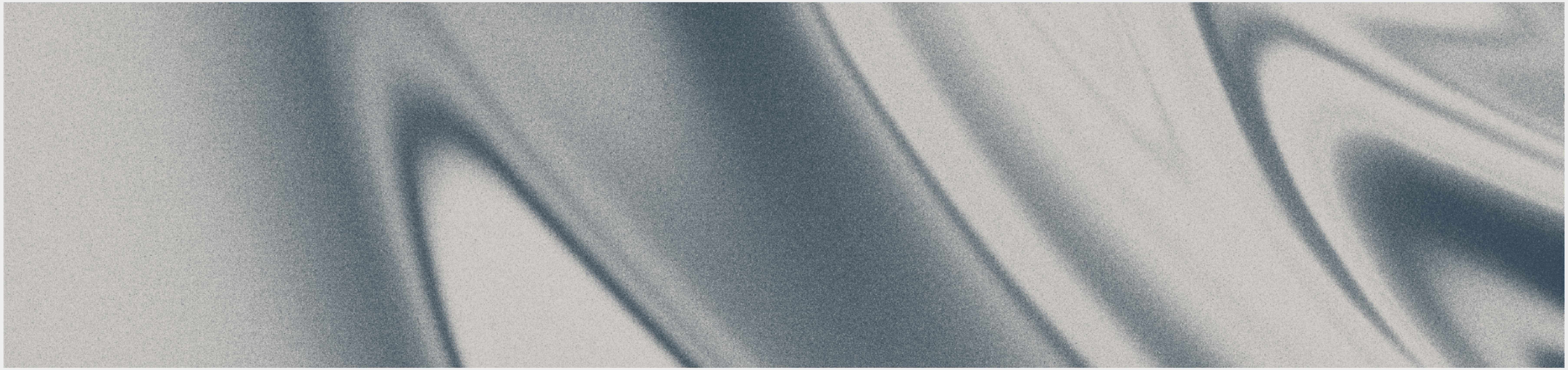




Time	Server Name	Destination	Destination Port	Protocol
222.577175	init.ess.apple.com	62.115.253.208	443	TLSv1.3
223.248546	kt-prod.ess.apple.com	17.145.0.2	443	TLSv1.3
250.471089	p113-caldav.icloud.com	17.250.84.36	443	TLSv1.2
301.339923	edge-102.sesto4.icloud-content.com	17.250.84.37	443	TLSv1.3
302.194211	p31-content.icloud.com	17.250.84.22	443	TLSv1.2
314.766744	setup.icloud.com	17.250.84.19	443	TLSv1.2
339.869951	backuprabbit.com	104.21.21.154	443	TLSv1.3
359.630968	gsa.apple.com	17.32.194.2	443	TLSv1.2
360.605764	backuprabbit.com	104.21.21.154	443	TLSv1.3
361.092903	pds-init.ess.apple.com	62.115.253.218	443	TLSv1.3
368.065719	cloudsponcer.com	104.21.79.172	443	TLSv1.3
377.414078	backuprabbit.com	104.21.21.154	443	TLSv1.3



2. File System Analysis



iTunes Backup

What does an iTunes backup save?

- **Media files:** photos, videos, and other media files.
- **Application Data:** App settings, preferences, data, documents and install profiles.
- **Settings:** Network settings (Wi-Fi hotspots, VPN settings, network preference), Paired Bluetooth devices.
- **Other Data:** Notes, Calendar events, ...

Encrypted backups include:

- Keychain data
- Wi-Fi settings
- Website history
- Health data
- Call,messages history

Encrypted backups don't include Face ID, Touch ID or device passcode data

iTunes Backup

How to create a backup ?

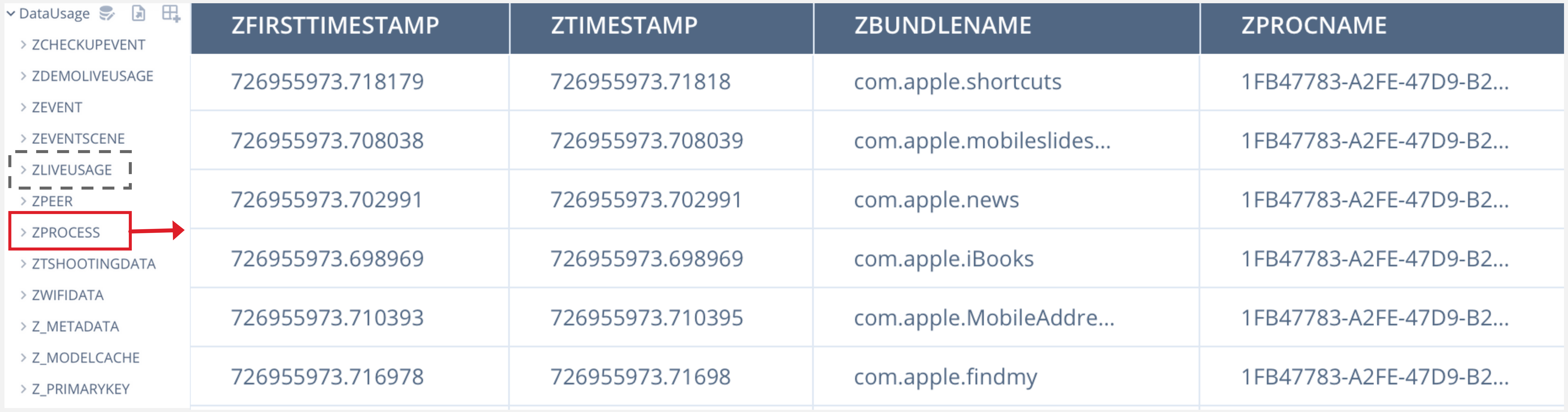
- Commercial Forensic Tools (Cellebrite, Elcomsoft, Magnet axiome, oxygen,...)
- iMazing
- iTunes (now Finder)
- iphone Backup Extractor
- libimobiledevice
- ...

Where to start ?

- Narrow down a timeline of events
- identify any applications that may be exhibiting odd behavior
- Do the same with any services (i.e. microphone, camera)
- Research avenues that data could get onto the device (messaging apps, email, bluetooth, web history/downloads)

iTunes Backup - Analysis

- **DataUsage.sqlite**



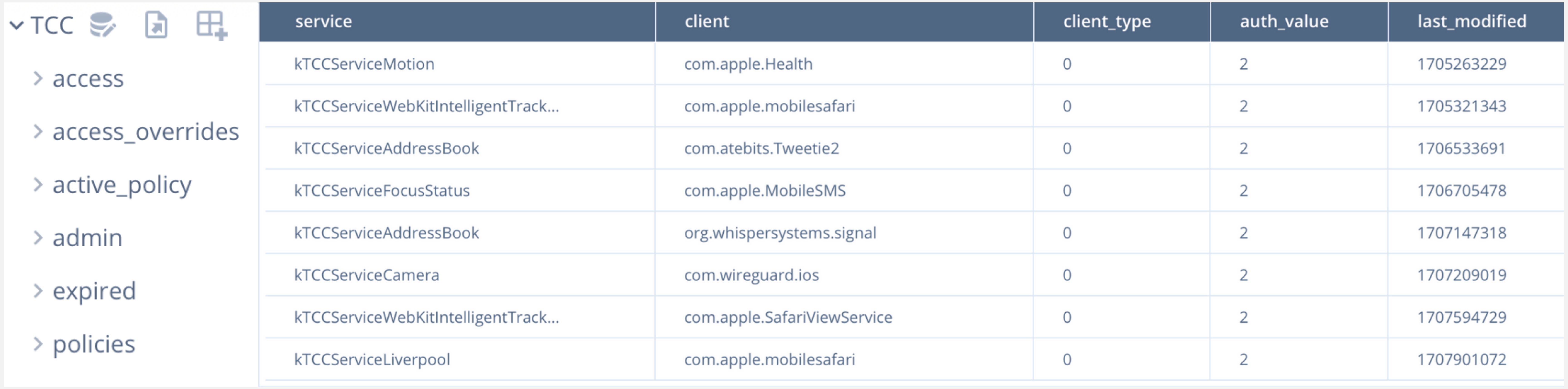
DataUsage
- ZCHECKUPEVENT
- ZDEMOLIVEUSAGE
- ZEVENT
- ZEVENTSCENE
- ZLIVEUSAGE
- ZPEER
- ZPROCESS
- ZTSHOOTINGDATA
- ZWIFIDATA
- Z_METADATA
- Z_MODELCACHE
- Z_PRIMARYKEY

ZFIRSTTIMESTAMP	ZTIMESTAMP	ZBUNDLENAME	ZPROCNAME
726955973.718179	726955973.71818	com.apple.shortcuts	1FB47783-A2FE-47D9-B2...
726955973.708038	726955973.708039	com.apple.mobileslides...	1FB47783-A2FE-47D9-B2...
726955973.702991	726955973.702991	com.apple.news	1FB47783-A2FE-47D9-B2...
726955973.698969	726955973.698969	com.apple.iBooks	1FB47783-A2FE-47D9-B2...
726955973.710393	726955973.710395	com.apple.MobileAddre...	1FB47783-A2FE-47D9-B2...
726955973.716978	726955973.71698	com.apple.findmy	1FB47783-A2FE-47D9-B2...

iTunes Backup - Analysis

- Can Artifacts tell the story: Check the app permissions

TCC.db - know which services your applications are using



- TCC
  - access
  - access_overrides
  - active_policy
  - admin
  - expired
  - policies

service	client	client_type	auth_value	last_modified
kTCCServiceMotion	com.apple.Health	0	2	1705263229
kTCCServiceWebKitIntelligentTrack...	com.apple.mobilesafari	0	2	1705321343
kTCCServiceAddressBook	com.atebits.Tweetie2	0	2	1706533691
kTCCServiceFocusStatus	com.apple.MobileSMS	0	2	1706705478
kTCCServiceAddressBook	org.whispersystems.signal	0	2	1707147318
kTCCServiceCamera	com.wireguard.ios	0	2	1707209019
kTCCServiceWebKitIntelligentTrack...	com.apple.SafariViewService	0	2	1707594729
kTCCServiceLiverpool	com.apple.mobilesafari	0	2	1707901072

iTunes Backups may take hours depending on the size of the files on the device
Limited amount of data is available

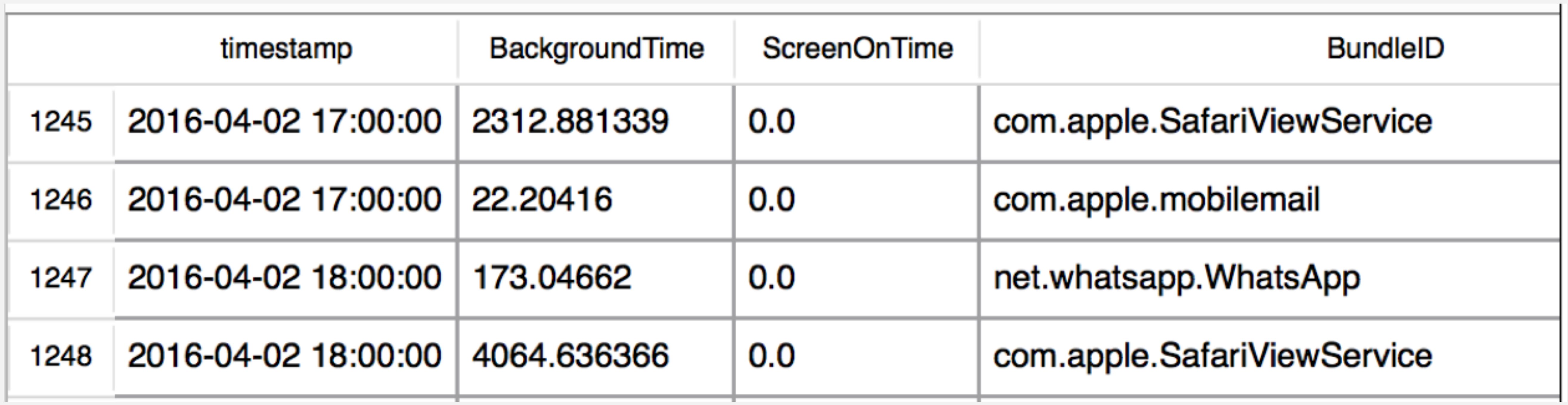
Full File System Extraction

More complete !

App Usage Time

➡ CurrentPowerlogs.plsql

- Size > 426 tables
- /private/var/containers/Shared/SystemGroup/<GUID>/Library/BatteryLife/CurrentPowerlog.PLSQL



	timestamp	BackgroundTime	ScreenOnTime	BundleID
1245	2016-04-02 17:00:00	2312.881339	0.0	com.apple.SafariViewService
1246	2016-04-02 17:00:00	22.20416	0.0	com.apple.mobilemail
1247	2016-04-02 18:00:00	173.04662	0.0	net.whatsapp.WhatsApp
1248	2016-04-02 18:00:00	4064.636366	0.0	com.apple.SafariViewService

Full File System Extraction

Detecting blocked OTA Update

➡ /var/mobile/Library/Preferences/com.apple.softwareupdateservicesd.plist

Settings -> General -> Software Update -> Automatic Updates

-> Download iOS Updates

SUDisableAutoDownload

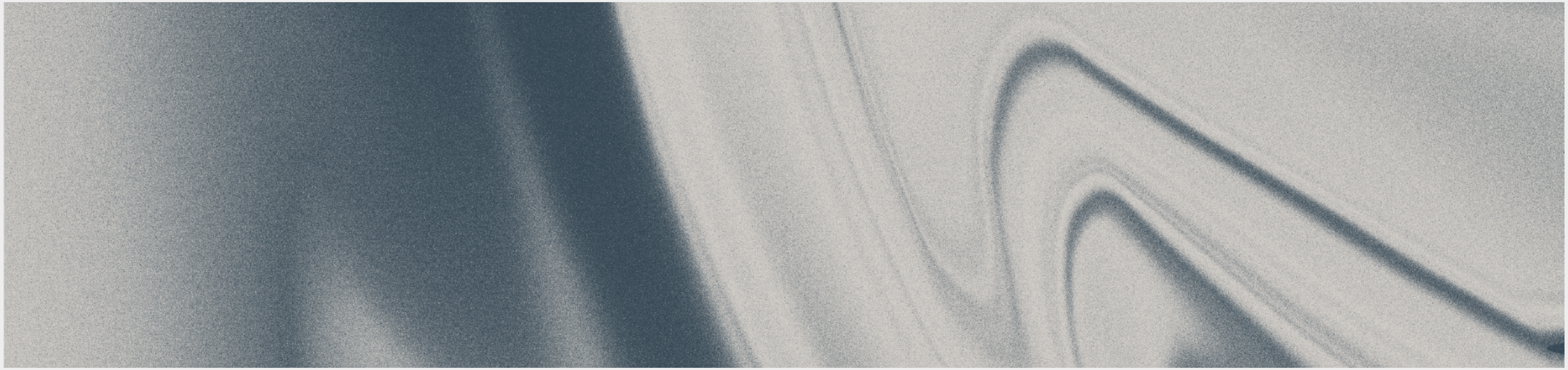
-> Install iOS Updates

SUAutomaticUpdateV2Enabled

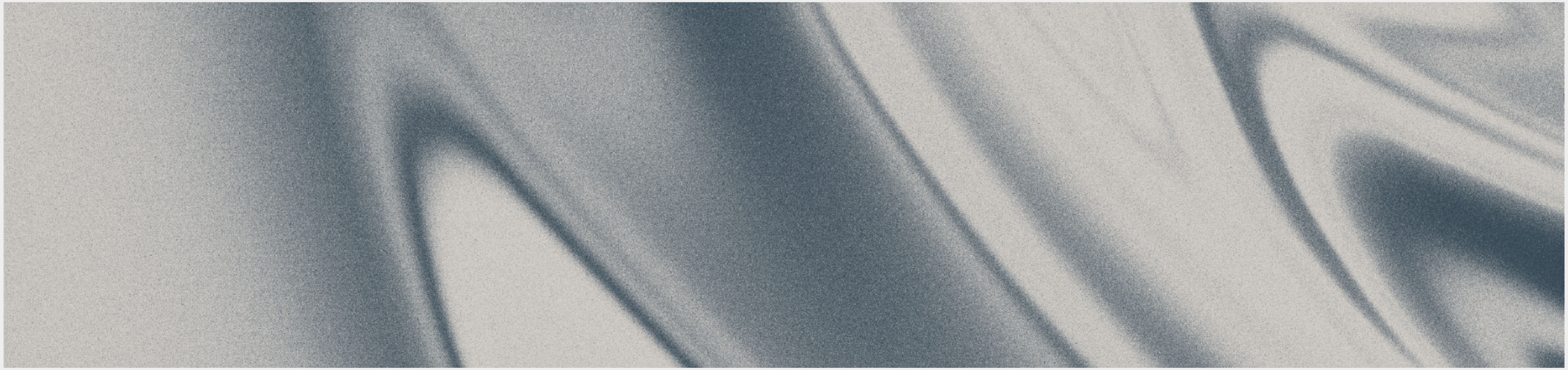
-> Security Responses & System Files

SUAutoInstallSystemDataFiles

⚠ **Need to jailbreak the device**

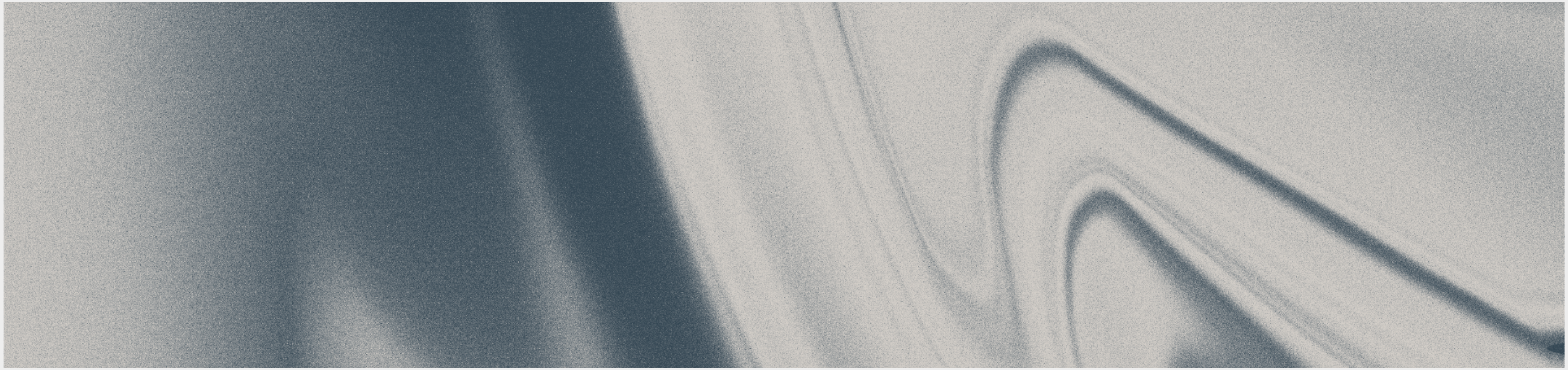


3. Diagnostic Information

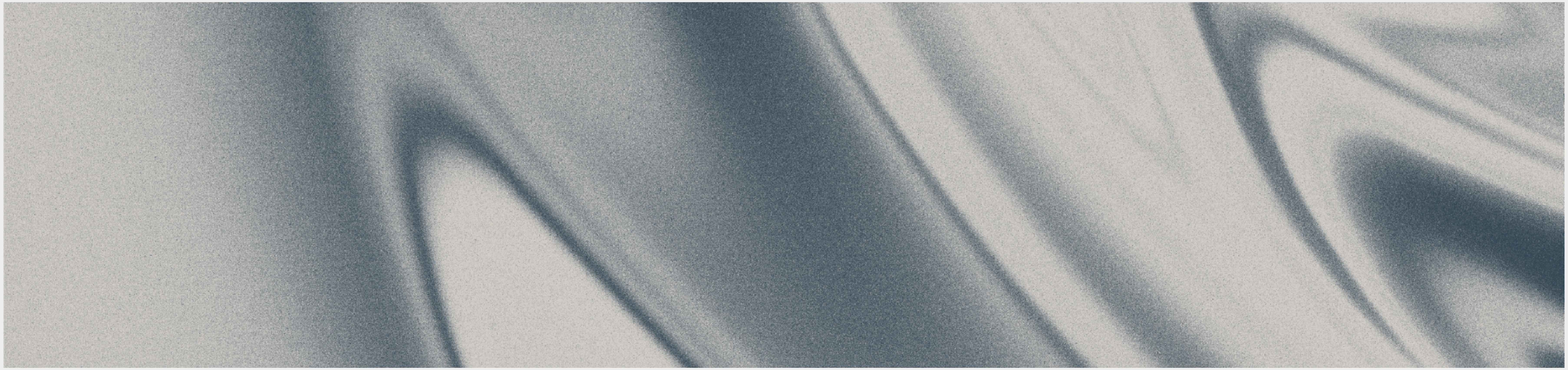


Diagnostic Information

- **Crashlogs:**
  - Investigate crashlogs to identify patterns or anomalies
  - Look for indications of malicious activities or vulnerabilities
- **Sysdiagnose**:
  - Analyze sysdiagnose reports for system-level information
  - Identify any irregularities that may point towards security breaches



III. Existing Projects



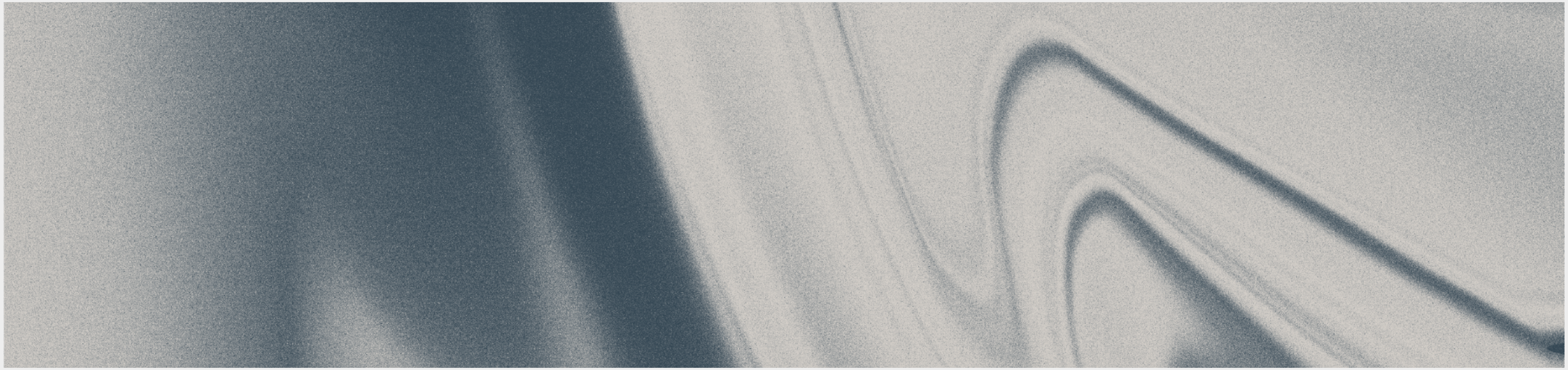
Network Analysis – TinyCheck

- Developed by Kaspersky
- Analyzes outgoing traffic from a device, using a Wi-Fi connection, and identifies interactions with known sources, such as servers linked to stalkerware
- **The project makes it possible to detect in certain cases the presence of more sophisticated implants implemented by malicious actors**

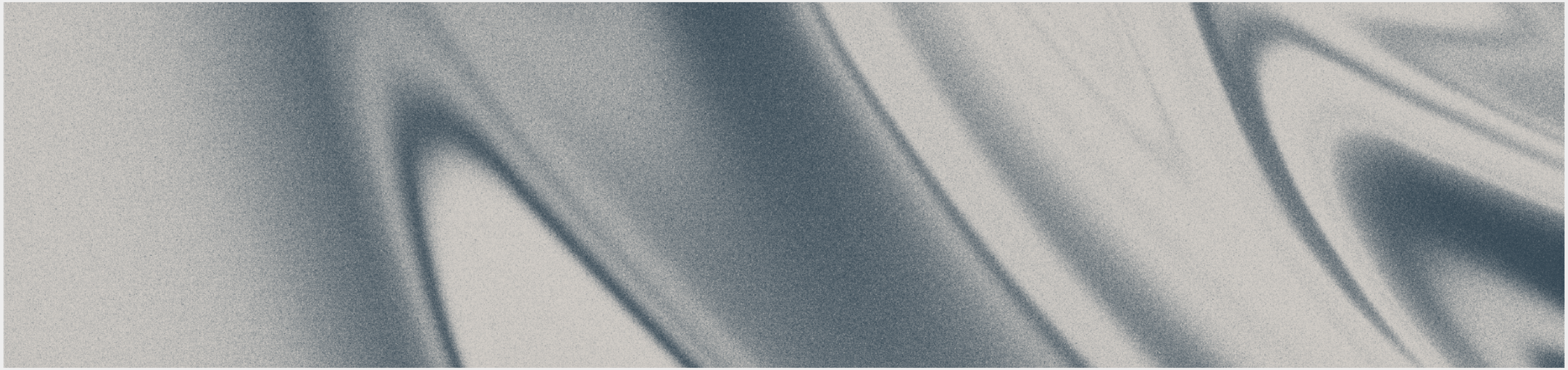
Backup Analysis – MVT

- Public project: https://github.com/mvt-project/mvt
- Developed by Amnesty International
- Processing and parsing records from numerous iOS system and apps databases, logs and system analytics
- Comparing extracted records to malicious indicators in STIX2 format
- Generating a unified chronological timeline of extracted records

⚠ **Has access to private user data**



IV. Sysdiagnose

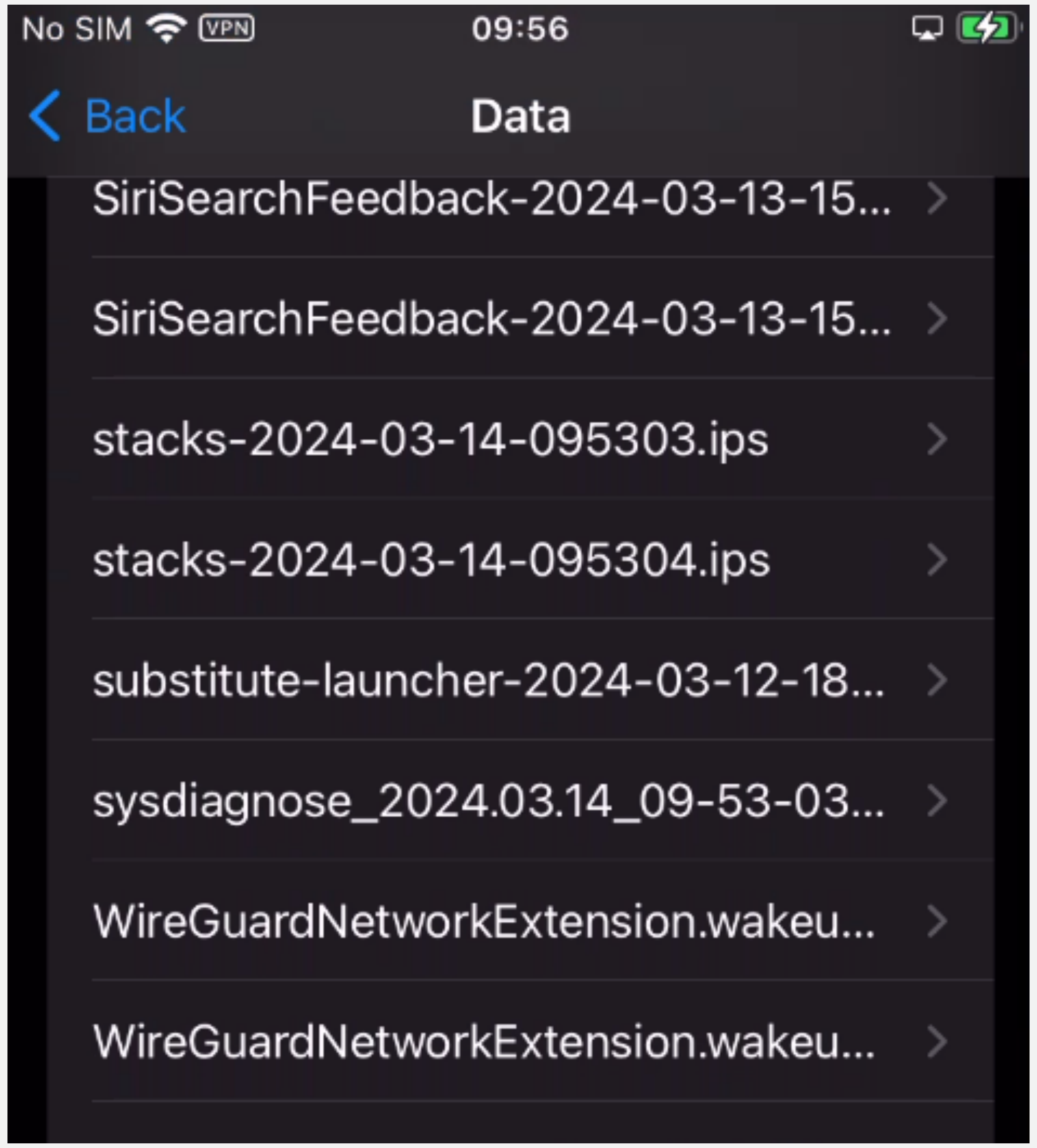


Diagnostic Information - Sysdiagnose

- The sysdiagnose tool gathers system diagnostic information helpful in investigating system performance issues

- **Generation**
  - Simultaneously pressing and releasing both volume buttons + the Side (or Top) button for 1 to 1.5 seconds.
  - Can take up to 10 min.
  - Locate it on settings > Privacy > Analytics & Improvements > Analytics Data

- **How to retrieve it:**
  - libimobiledevice: *idevicecrashreport* command
  - Finder/Airdrop
  - Commercial Tools:
    - Cellebrite
    - Magnet Forensics
    - ...



SiriSearchFeedback-2024-03-13-15...
SiriSearchFeedback-2024-03-13-15...
stacks-2024-03-14-095303.ips
stacks-2024-03-14-095304.ips
substitute-launcher-2024-03-12-18...
sysdiagnose_2024.03.14_09-53-03...
WireGuardNetworkExtension.wakeu...
WireGuardNetworkExtension.wakeu...

Diagnostic Information – Sysdiagnose

- **Think about privacy !**

Sysdiagnose contains no user data but lots of metadata

- Apps installed
- Hardware details
- Device configuration
- Network configuration & connections
- Logs
- Usage overview
- Results of commands run on the device
- ...

Different formats of files:

- SQLite
- Plist
- CSV
- ASCII Text
- GZIP Files

Diagnostic Information - Sysdiagnose

- **Interesting files**

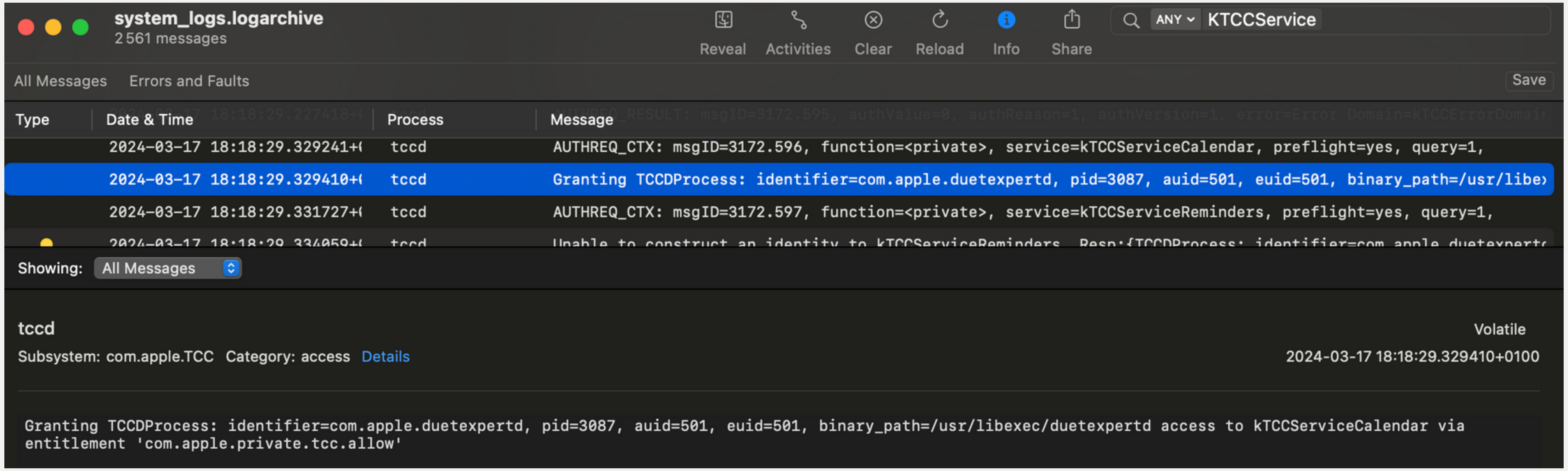
  - ./ps.txt
  - Ps_thread.txt
  - ./*/logs/MobileContainerManager
  - ./*/logs/powerlogs/powerlog_*: extracted from the CurrentPowerlog.PLSQL
  - logs/Networking
  - logs/MobileInstallation
  - Wifi, Airdrop, Bluetooth data in details

Diagnostic Information - Sysdiagnose

Unified Logs

- A collection of logs from the iOS device located in: **system_logs. logarchive** folder on a sysdiagnose
- Can be viewed with the native Mac OS Console
- Record as much informations as possible regarding the device's activity
- Have a limited duration

Example of a log emmited by **tccd**, this line tells that the process **duetexpertd** has been granted access to **kTCCServiceCalendar**



```
Type   Date & Time                    Process   Message
       2024-03-17 18:18:29.329241+(   tccd      AUTHREQ_CTX: msgID=3172.596, function=<private>, service=kTCCServiceCalendar, preflight=yes, query=1,
       2024-03-17 18:18:29.329410+(   tccd      Granting TCCDProcess: identifier=com.apple.duetexpertd, pid=3087, auid=501, euid=501, binary_path=/usr/libe›
       2024-03-17 18:18:29.331727+(   tccd      AUTHREQ_CTX: msgID=3172.597, function=<private>, service=kTCCServiceReminders, preflight=yes, query=1,
       2024-03-17 18:18:29.334059+(   tccd      Unable to construct an identity to kTCCServiceReminders  Resp:{TCCDProcess: identifier=com.apple.duetexpertc
```

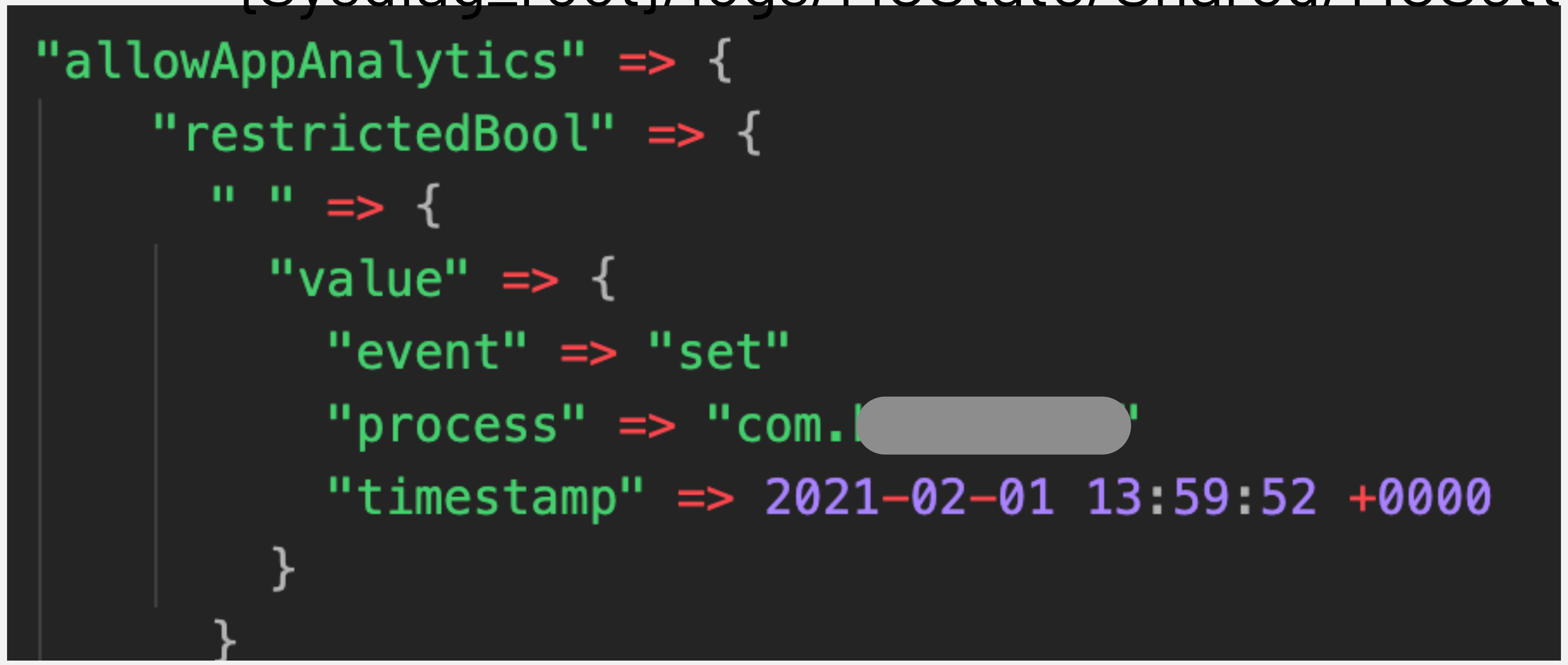
```
tccd                                                                              Volatile
Subsystem: com.apple.TCC  Category: access  Details                2024-03-17 18:18:29.329410+0100

Granting TCCDProcess: identifier=com.apple.duetexpertd, pid=3087, auid=501, euid=501, binary_path=/usr/libexec/duetexpertd access to kTCCServiceCalendar via
entitlement 'com.apple.private.tcc.allow'
```

Diagnostic Information - Sysdiagnose

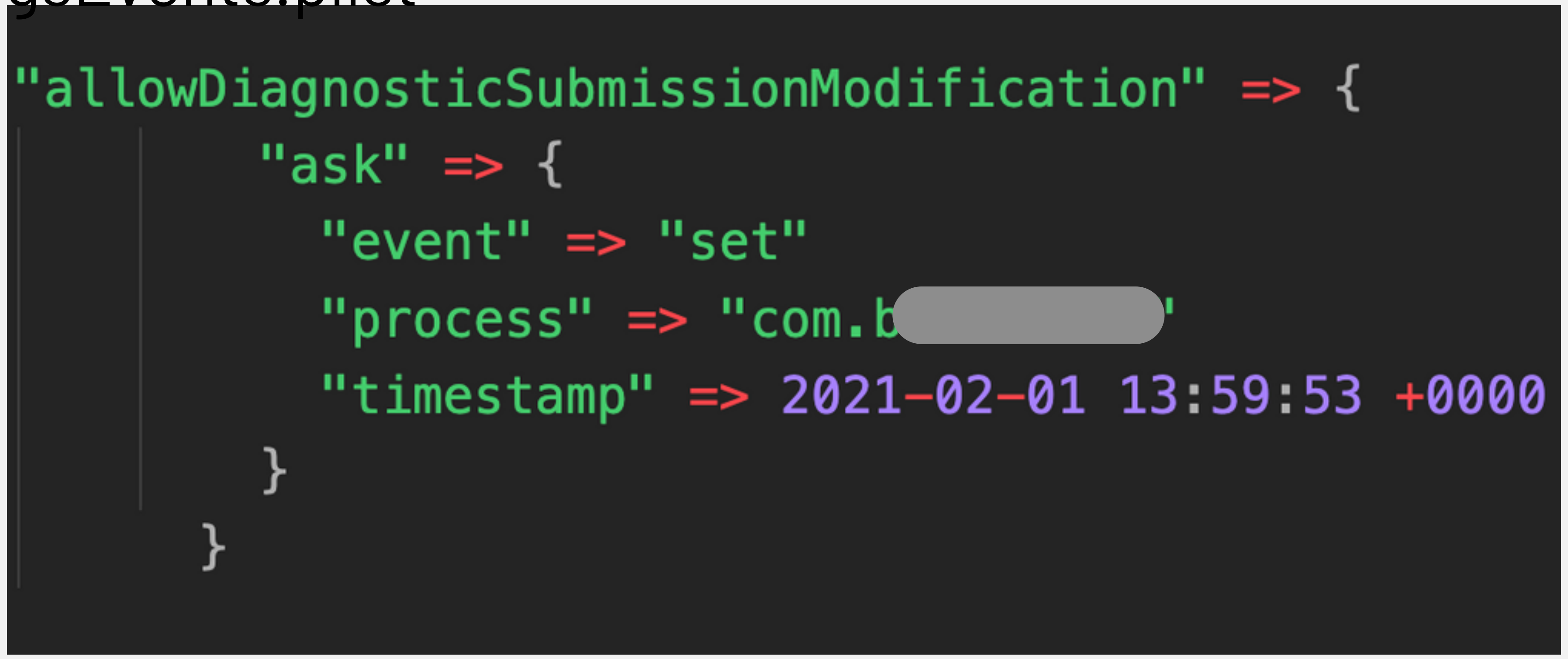
■ **Settings modified by malware**

MCSettingsEvents.plist

- Contains logs of settings changes
- Path: {Sysdiag_root}/logs/MCState/Shared/MCSettingsEvents.plist



```
"allowAppAnalytics" => {
    "restrictedBool" => {
      " " => {
        "value" => {
          "event" => "set"
          "process" => "com.[redacted]"
          "timestamp" => 2021-02-01 13:59:52 +0000
        }
      }
```



```
"allowDiagnosticSubmissionModification" => {
        "ask" => {
          "event" => "set"
          "process" => "com.[redacted]"
          "timestamp" => 2021-02-01 13:59:53 +0000
        }
      }
```

Diagnostic Information – Sysdiagnose

- **MCSettings.plist entries:**

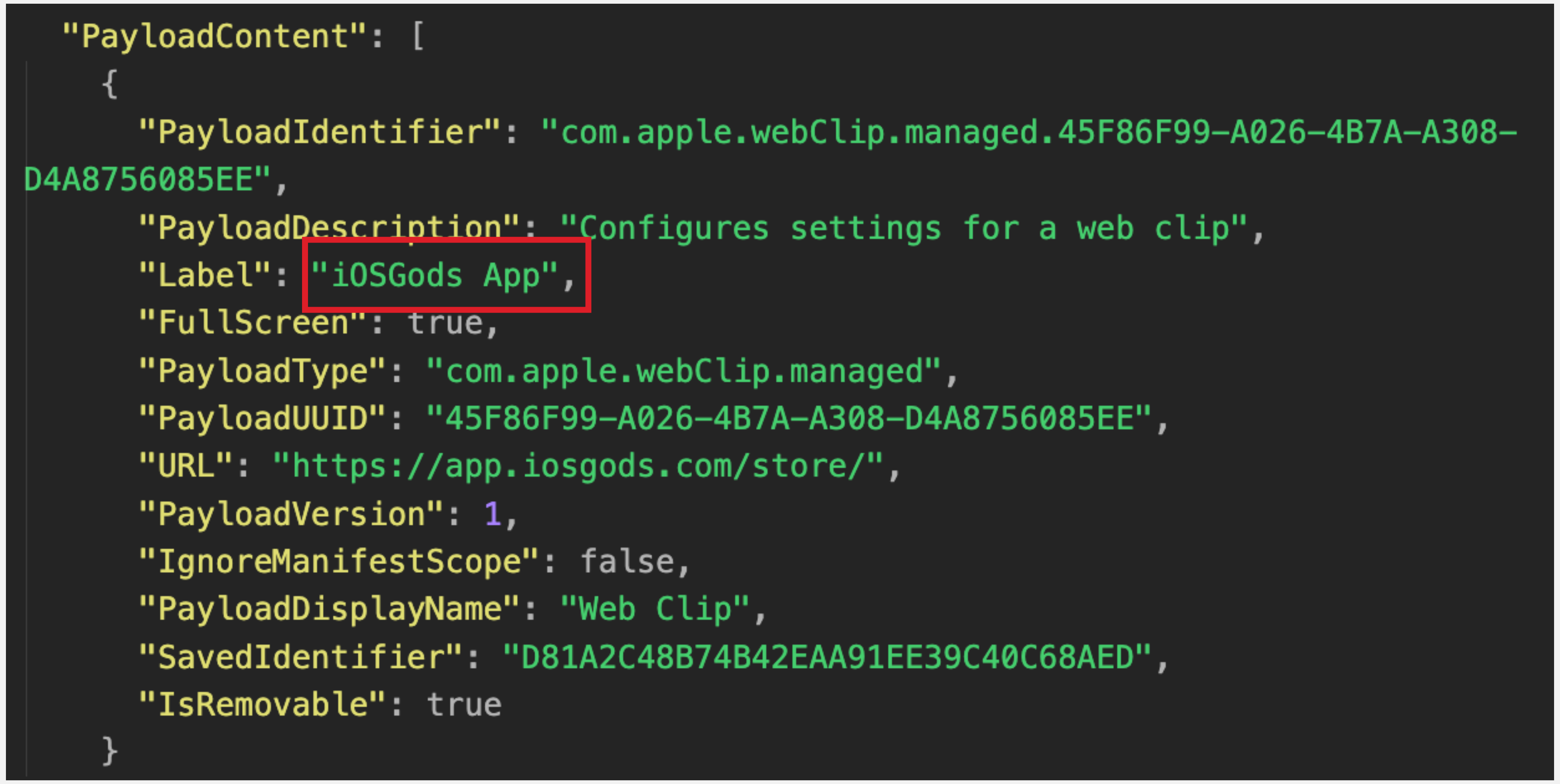
allowUntrustedTLSPrompt	Select to allow the device user to accept untrusted HTTPS certificates.
allowDiagnosticSubmission	Sends diagnostics to apple
allowAppRemoval	If false, disables removal of apps from iOS devices
allowUIConfigurationProfileInstallation	If false, the user is prohibited from installing configuration profiles and certificates interactively.
allowAutomaticAppDownloads	If false, it prevents automatic downloading of apps purchased on other devices.
allowSafetyDataSubmission	Check whom the device is sharing information with, restrict Messages and FaceTime to the iPhone, reset system privacy permissions for apps
allowSystemAppRemoval	If false, the system disables the removal of system apps from the device

Diagnostic Information – Sysdiagnose

- **Configuation Profiles**
  - Configuration profiles automate the configuration of settings, accounts, restrictions and credentials
  - These files can be created by an MDM solution or Apple Configurator for Mac or manually
    - Passcode and password policies
    - Restrictions on device features (for example, disabling the camera)
    - Network and VPN settings
    - Microsoft Exchange settings
    - Mail settings
    - Account settings
    - ...

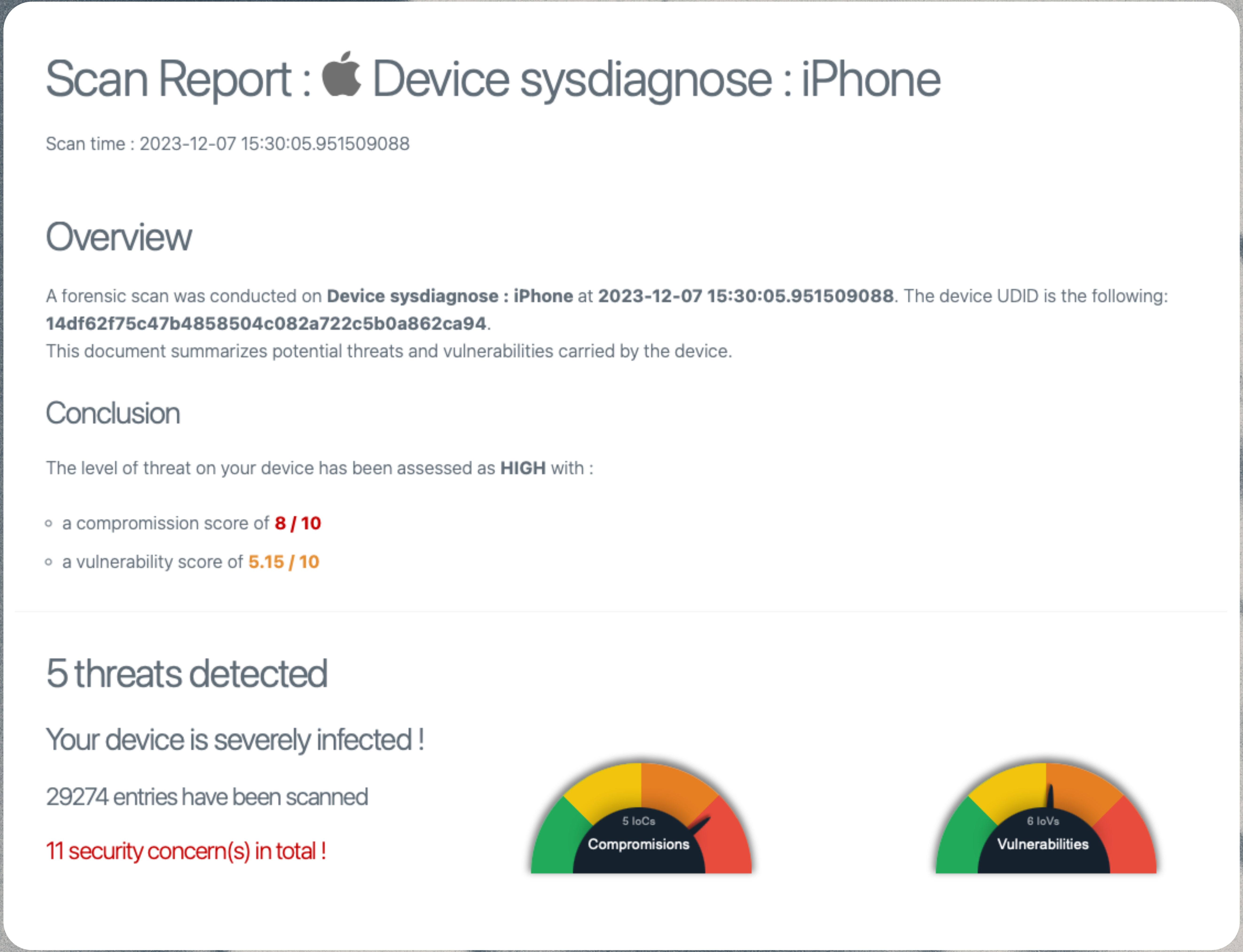
Diagnostic Information – Sysdiagnose

- **Detect Installation of Alternative App Store Apps**
  - Detection using webclip profiles
  - Web clips allow to add quick-access icons to the home screen of an iPad or iPhone that links directly to specified web pages.
  - Path: **{sysdiagFolder}/logs/MCState/Shared/profile-{ALNUMPSEUDORANDOM}.stub**



```
  "PayloadContent": [
    {
      "PayloadIdentifier": "com.apple.webClip.managed.45F86F99-A026-4B7A-A308-
D4A8756085EE",
      "PayloadDescription": "Configures settings for a web clip",
      "Label": "iOSGods App",
      "FullScreen": true,
      "PayloadType": "com.apple.webClip.managed",
      "PayloadUUID": "45F86F99-A026-4B7A-A308-D4A8756085EE",
      "URL": "https://app.iosgods.com/store/",
      "PayloadVersion": 1,
      "IgnoreManifestScope": false,
      "PayloadDisplayName": "Web Clip",
      "SavedIdentifier": "D81A2C48B74B42EAA91EE39C40C68AED",
      "IsRemovable": true
    }
```

SHINDAN
POWERED BY RANDORISEC



Scan Report : Device sysdiagnose : iPhone

Scan time : 2023-12-07 15:30:05.951509088

Overview

A forensic scan was conducted on **Device sysdiagnose : iPhone** at **2023-12-07 15:30:05.951509088**. The device UDID is the following: **14df62f75c47b4858504c082a722c5b0a862ca94**.
This document summarizes potential threats and vulnerabilities carried by the device.

Conclusion

The level of threat on your device has been assessed as **HIGH** with :

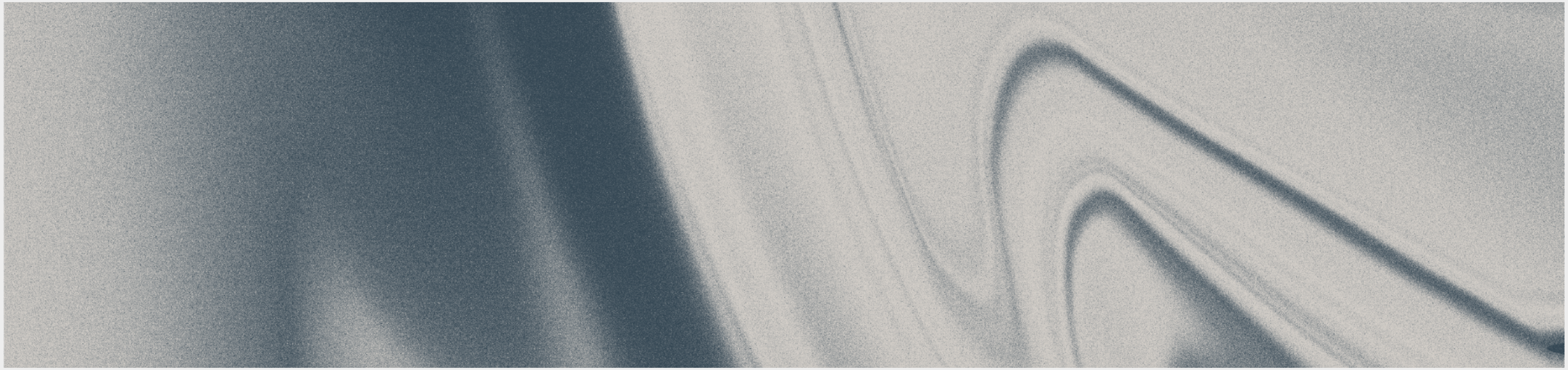
- a compromission score of **8 / 10**
- a vulnerability score of **5.15 / 10**

5 threats detected

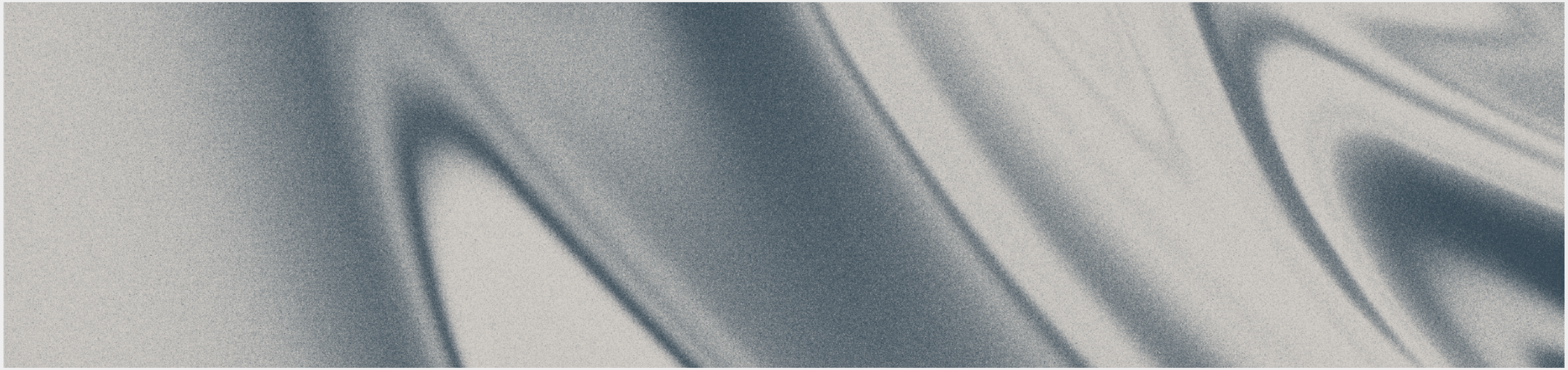
Your device is severely infected !

29274 entries have been scanned

11 security concern(s) in total !



V. Open source contribution



Jérôme == **Dev** != Cyber

Sysdiagnose is a Mess

An archive with :

- a lot of folders and files
- some are archives
- some are databases
- some are text
- some are structured (plist, xml, json ...)

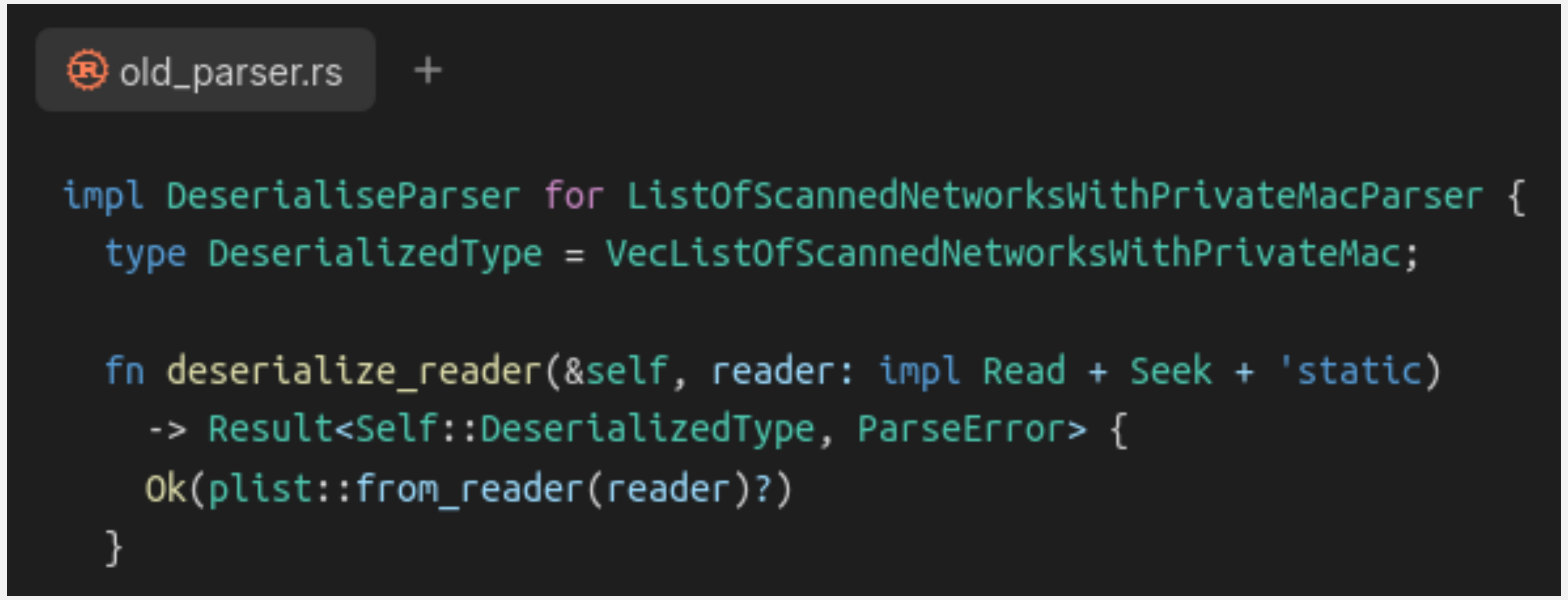
It's a real skill to get what we need

First attempt

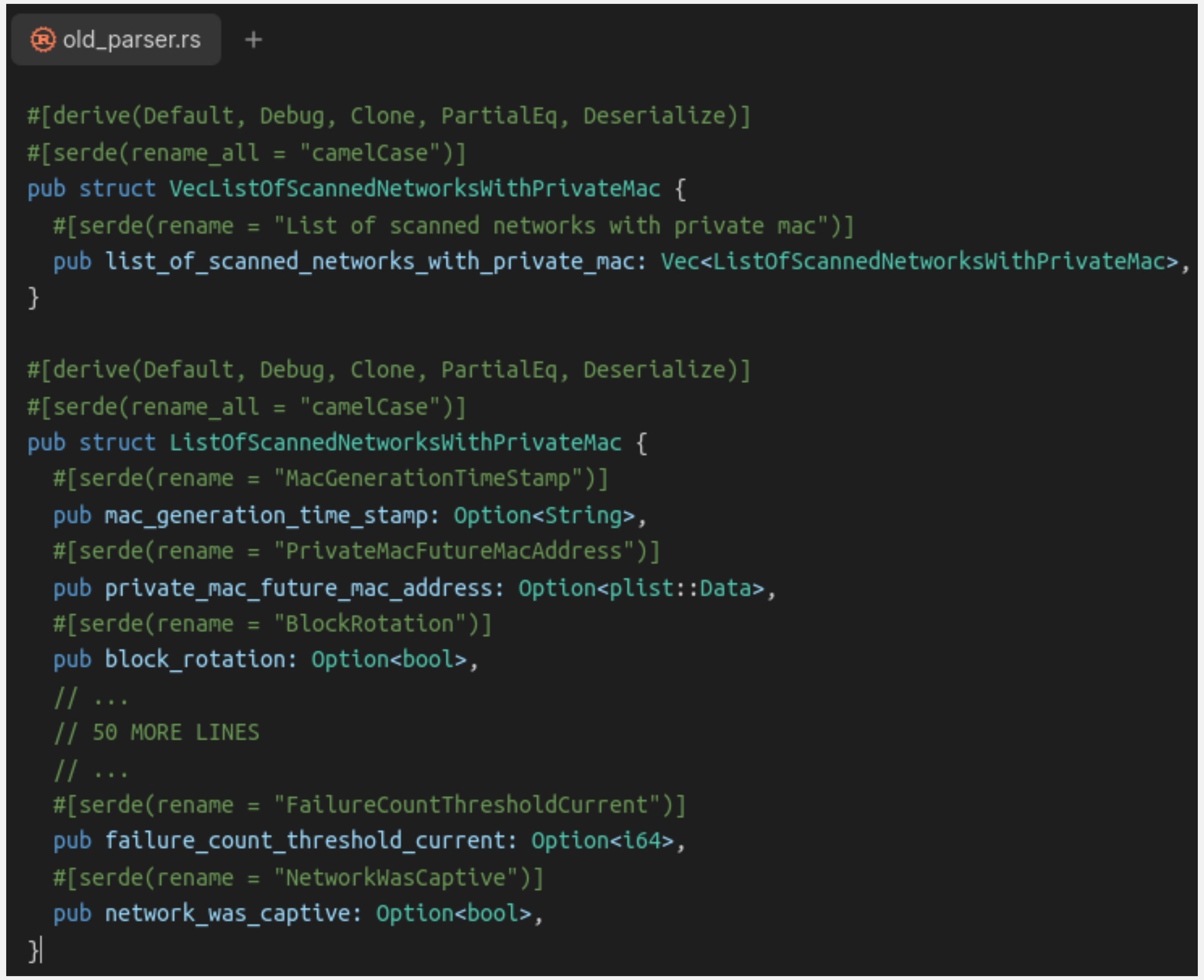
Ambitious :

- Specific parsers
- Strict data structures

Typed / Specific structures



```rust
impl DeserialiseParser for ListOfScannedNetworksWithPrivateMacParser {
  type DeserializedType = VecListOfScannedNetworksWithPrivateMac;

  fn deserialize_reader(&self, reader: impl Read + Seek + 'static)
    -> Result<Self::DeserializedType, ParseError> {
    Ok(plist::from_reader(reader)?)
  }
```



```rust
#[derive(Default, Debug, Clone, PartialEq, Deserialize)]
#[serde(rename_all = "camelCase")]
pub struct VecListOfScannedNetworksWithPrivateMac {
  #[serde(rename = "List of scanned networks with private mac")]
  pub list_of_scanned_networks_with_private_mac: Vec<ListOfScannedNetworksWithPrivateMac>,
}

#[derive(Default, Debug, Clone, PartialEq, Deserialize)]
#[serde(rename_all = "camelCase")]
pub struct ListOfScannedNetworksWithPrivateMac {
  #[serde(rename = "MacGenerationTimeStamp")]
  pub mac_generation_time_stamp: Option<String>,
  #[serde(rename = "PrivateMacFutureMacAddress")]
  pub private_mac_future_mac_address: Option<plist::Data>,
  #[serde(rename = "BlockRotation")]
  pub block_rotation: Option<bool>,
  // ...
  // 50 MORE LINES
  // ...
  #[serde(rename = "FailureCountThresholdCurrent")]
  pub failure_count_threshold_current: Option<i64>,
  #[serde(rename = "NetworkWasCaptive")]
  pub network_was_captive: Option<bool>,
}
```

Reality check :

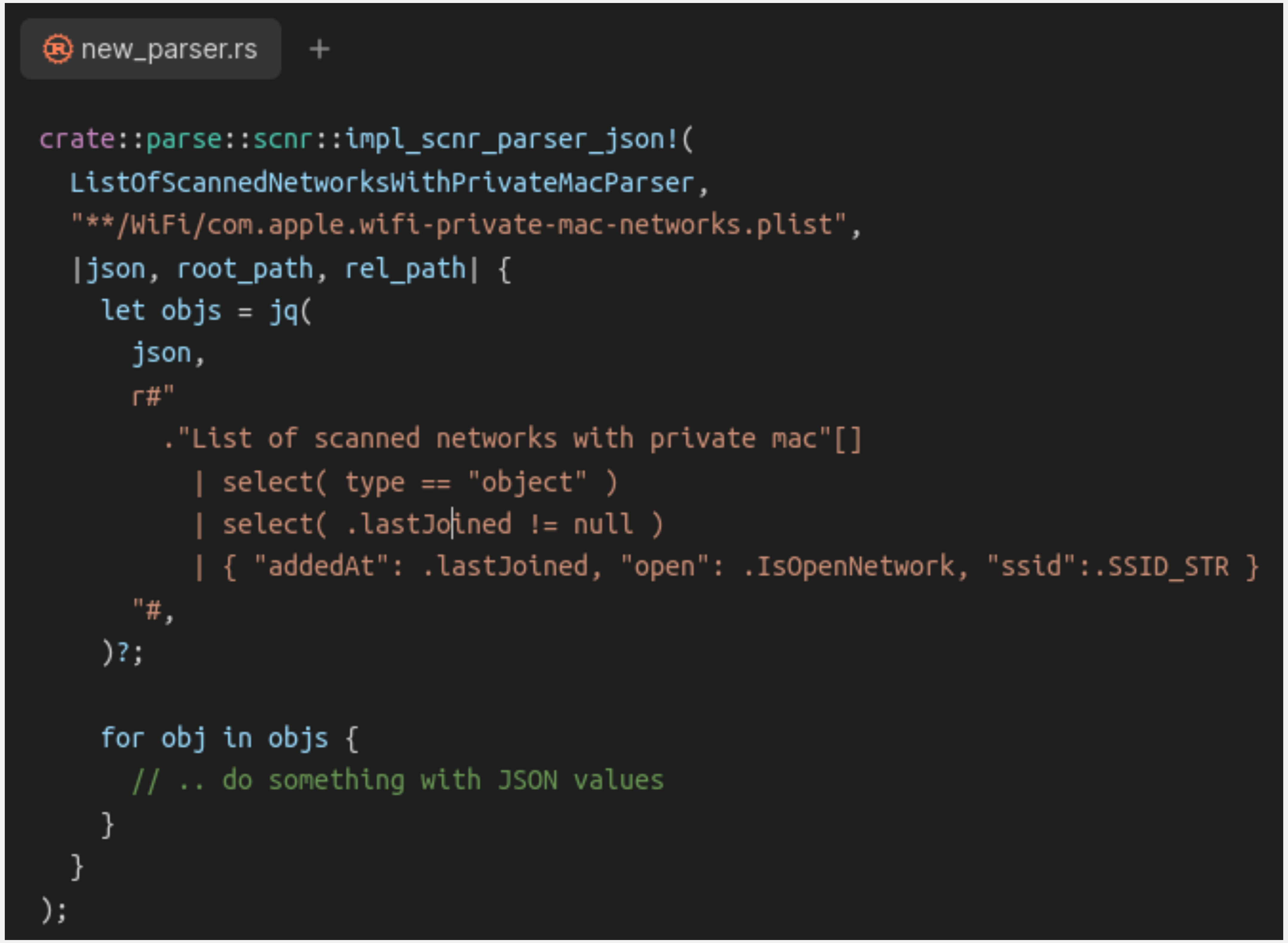
- Tedious !
- Fragile ! (versions !)
- Code duplication (not D.R.Y.)

New ways

Let’s talk **json** !

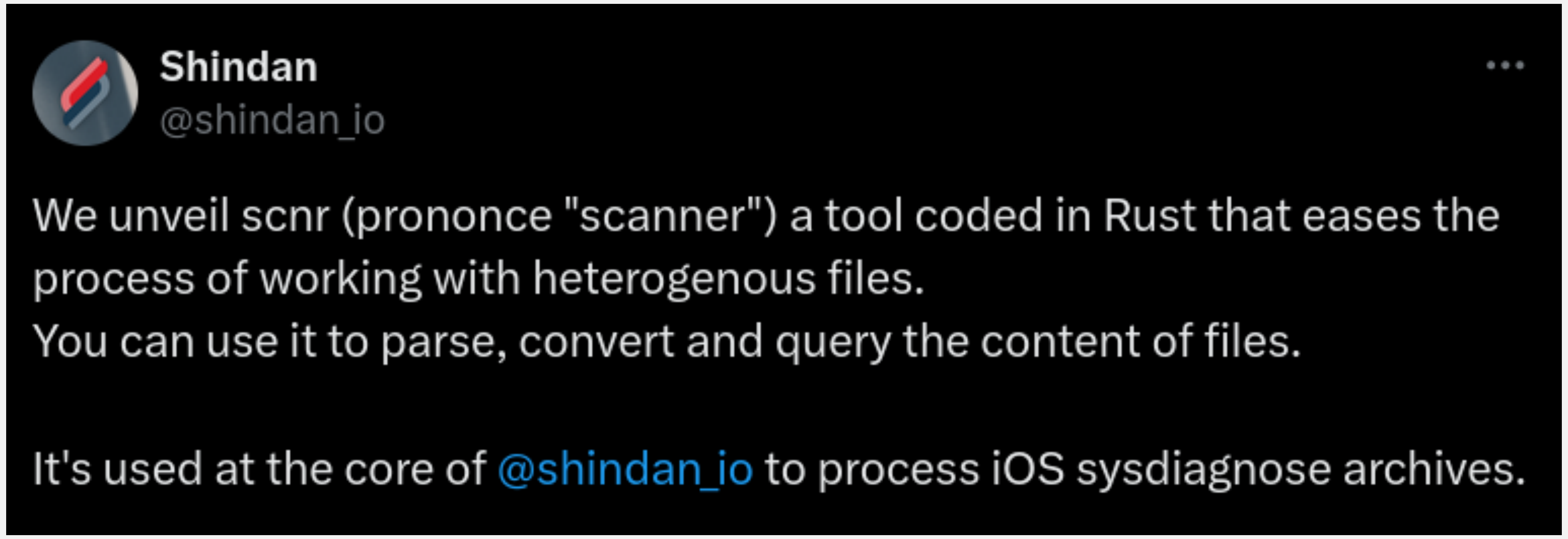
And

let’s query with **jq** !



```
crate::parse::scnr::impl_scnr_parser_json!(
  ListOfScannedNetworksWithPrivateMacParser,
  "**/WiFi/com.apple.wifi-private-mac-networks.plist",
  |json, root_path, rel_path| {
    let objs = jq(
      json,
      r#"
        ."List of scanned networks with private mac"[]
          | select( type == "object" )
          | select( .lastJoined != null )
          | { "addedAt": .lastJoined, "open": .IsOpenNetwork, "ssid":.SSID_STR }
      "#,
    )?;

    for obj in objs {
      // .. do something with JSON values
    }
  }
);
```

And so we open sourced our “digging” layer

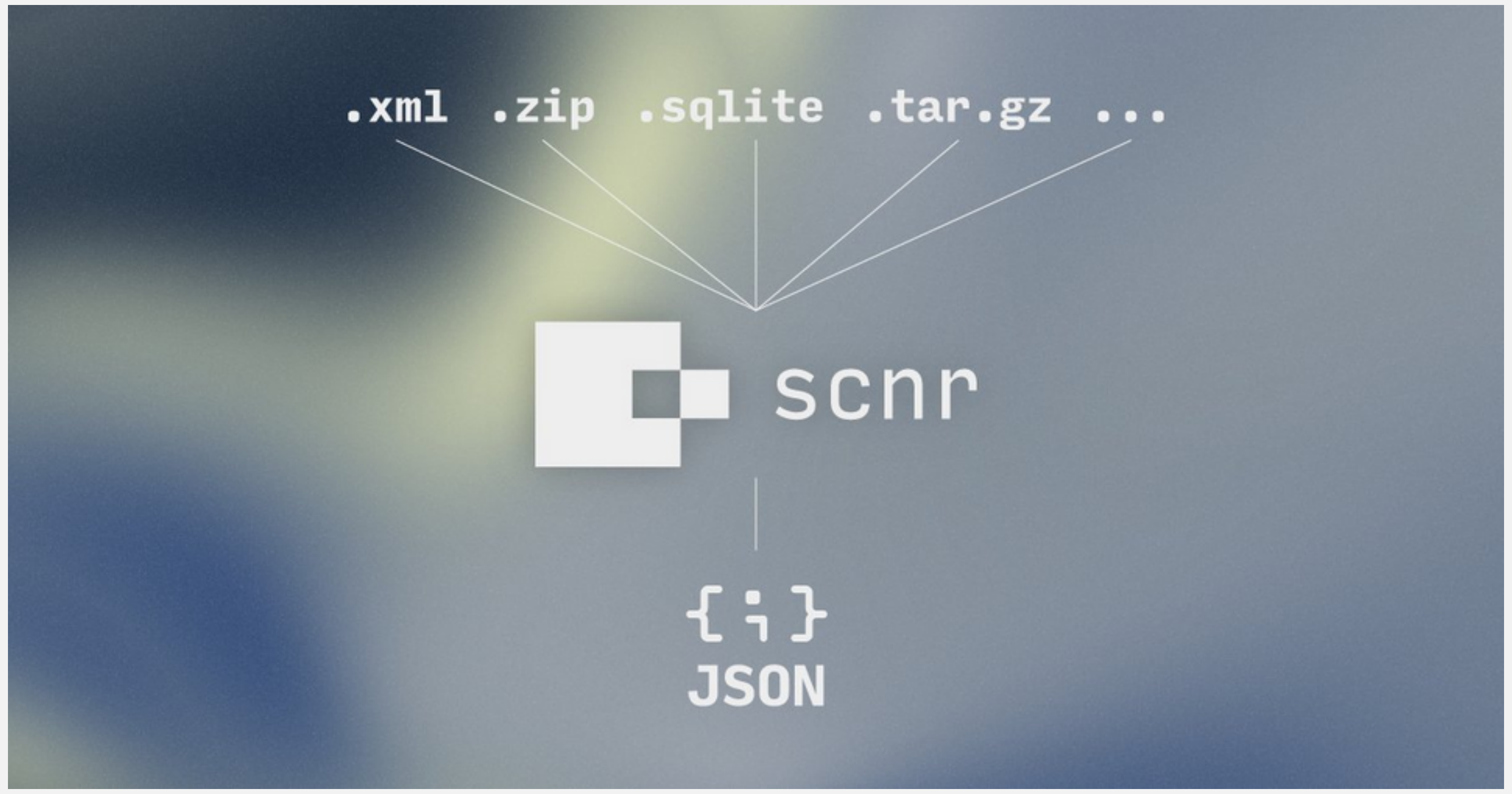


Shindan
@shindan_io

We unveil scnr (prononce "scanner") a tool coded in Rust that eases the process of working with heterogenous files.
You can use it to parse, convert and query the content of files.

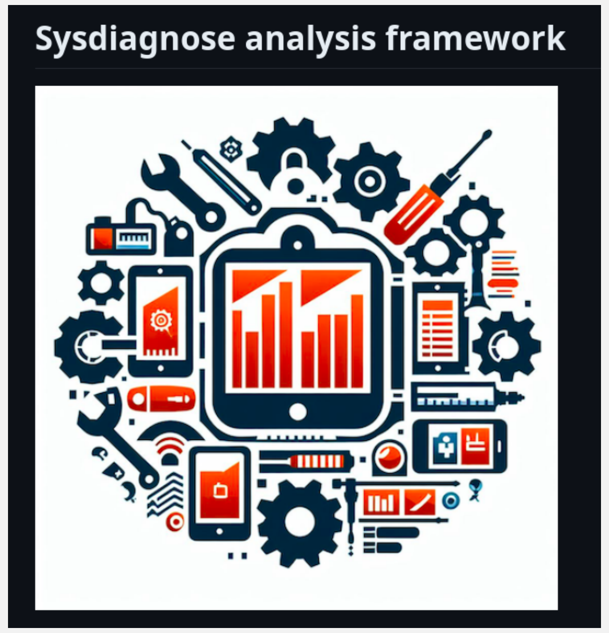
It's used at the core of @shindan_io to process iOS sysdiagnose archives.

https://github.com/shindan-io/scnr



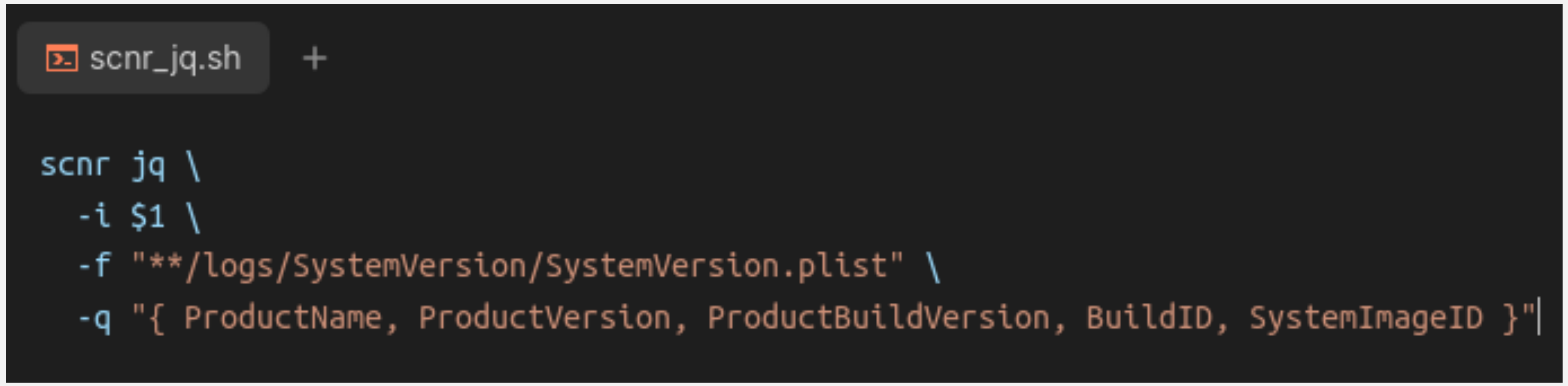
Yet another sysdiagnose digging tool ?

https://github.com/EC-DIGIT-CSIRC/sysdiagnose



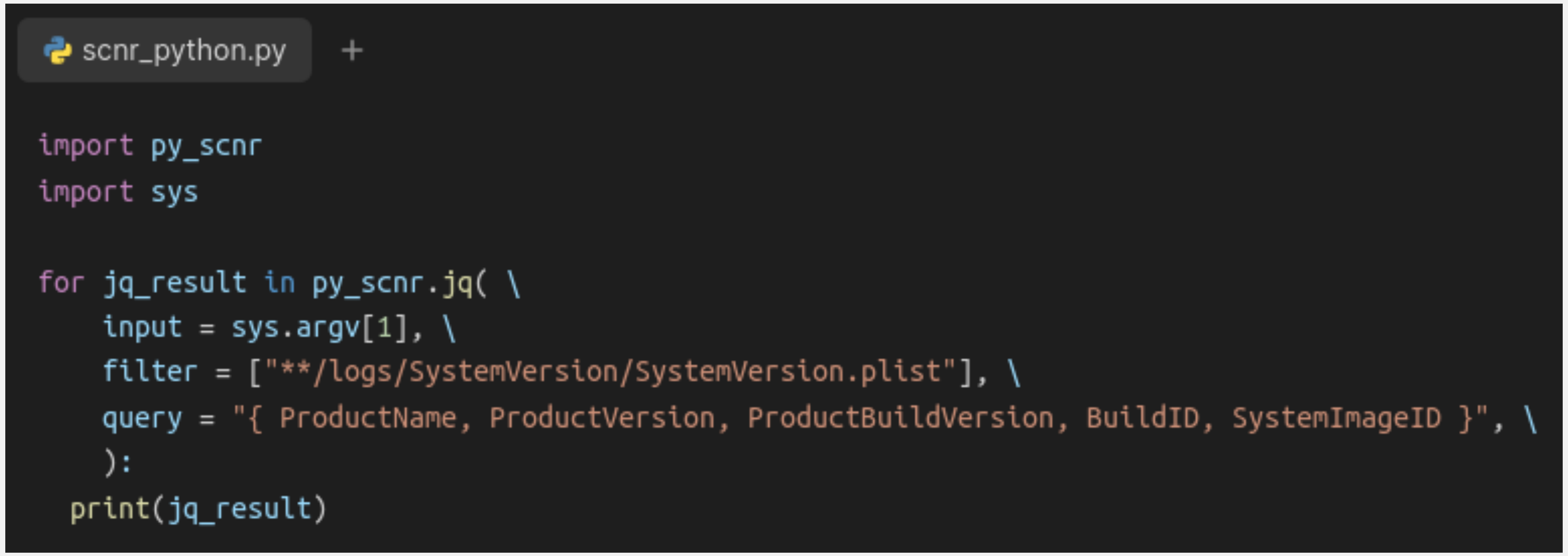
Choose your flavor

command line (as a rust lib too of course !)



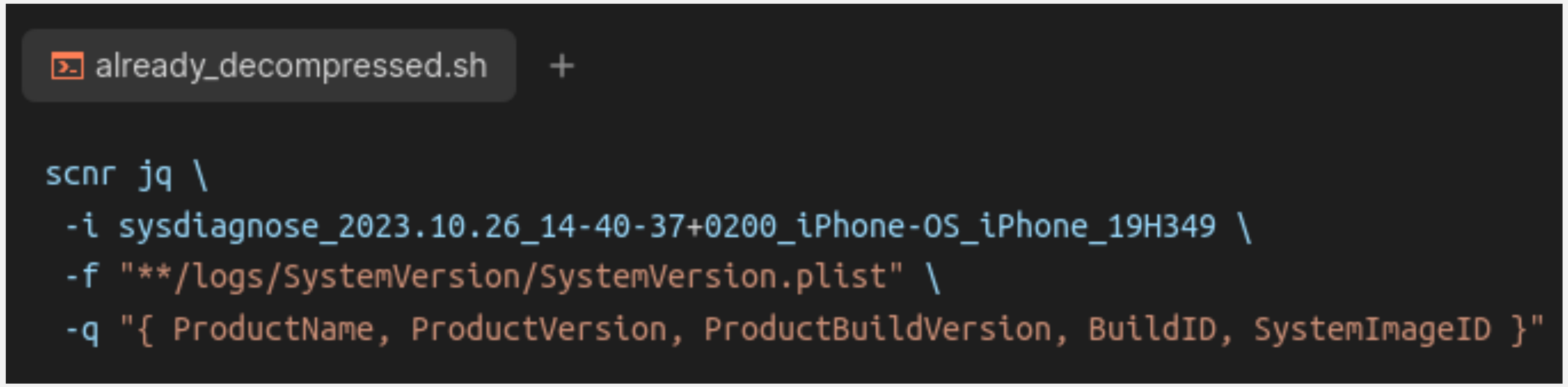
```
scnr jq \
  -i $1 \
  -f "**/logs/SystemVersion/SystemVersion.plist" \
  -q "{ ProductName, ProductVersion, ProductBuildVersion, BuildID, SystemImageID }"
```

python



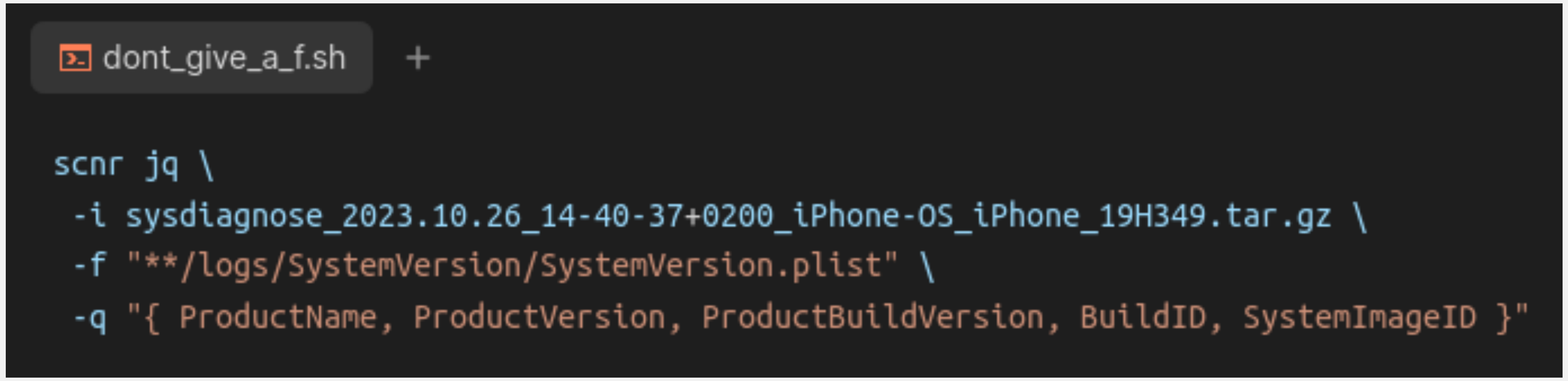
```python
import py_scnr
import sys

for jq_result in py_scnr.jq( \
    input = sys.argv[1], \
    filter = ["**/logs/SystemVersion/SystemVersion.plist"], \
    query = "{ ProductName, ProductVersion, ProductBuildVersion, BuildID, SystemImageID }", \
    ):
  print(jq_result)
```

In an archive ? no problem



already_decompressed.sh

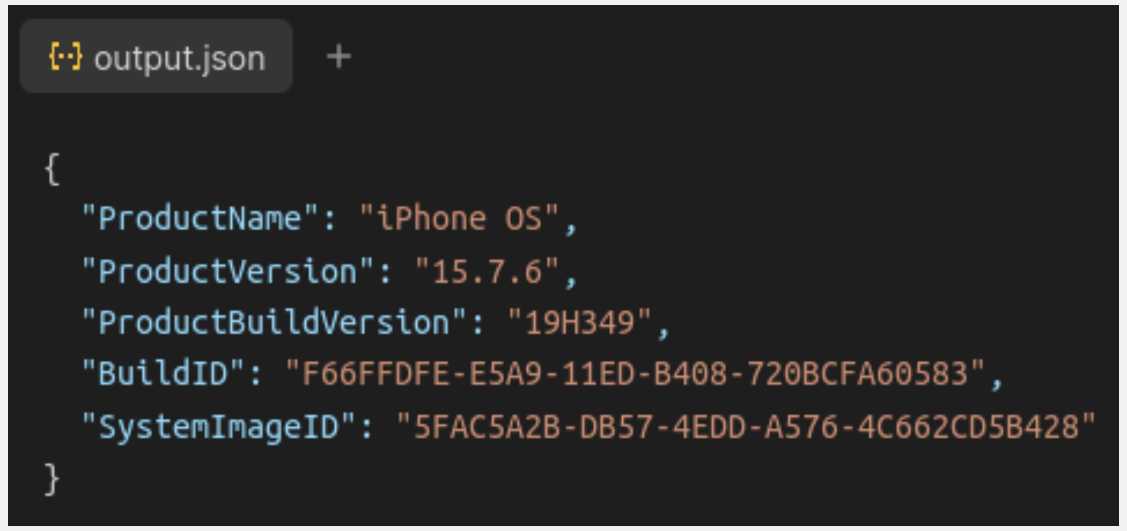
```
scnr jq \
 -i sysdiagnose_2023.10.26_14-40-37+0200_iPhone-OS_iPhone_19H349 \
 -f "**/logs/SystemVersion/SystemVersion.plist" \
 -q "{ ProductName, ProductVersion, ProductBuildVersion, BuildID, SystemImageID }"
```



dont_give_a_f.sh

```
scnr jq \
 -i sysdiagnose_2023.10.26_14-40-37+0200_iPhone-OS_iPhone_19H349.tar.gz \
 -f "**/logs/SystemVersion/SystemVersion.plist" \
 -q "{ ProductName, ProductVersion, ProductBuildVersion, BuildID, SystemImageID }"
```

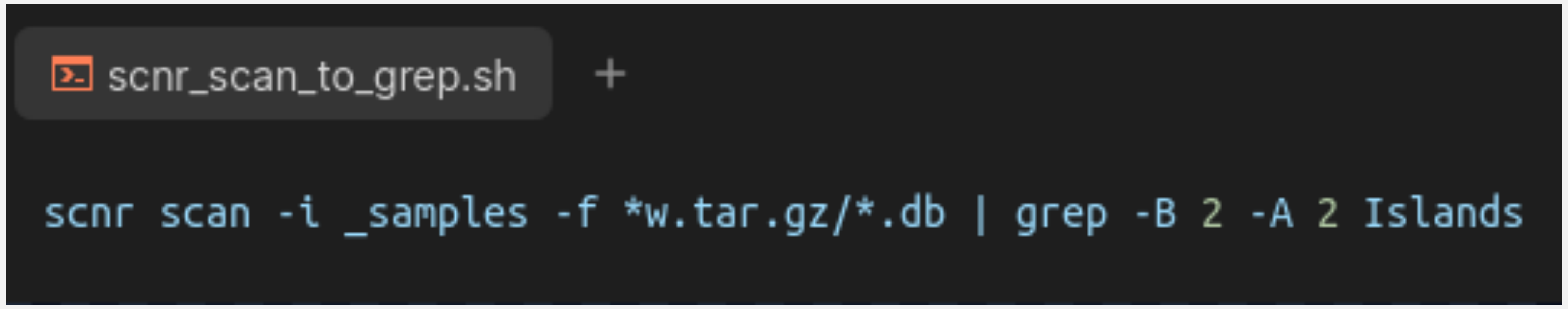
SAME RESULT =>



output.json

```
{
  "ProductName": "iPhone OS",
  "ProductVersion": "15.7.6",
  "ProductBuildVersion": "19H349",
  "BuildID": "F66FFDFE-E5A9-11ED-B408-720BCFA60583",
  "SystemImageID": "5FAC5A2B-DB57-4EDD-A576-4C662CD5B428"
}
```

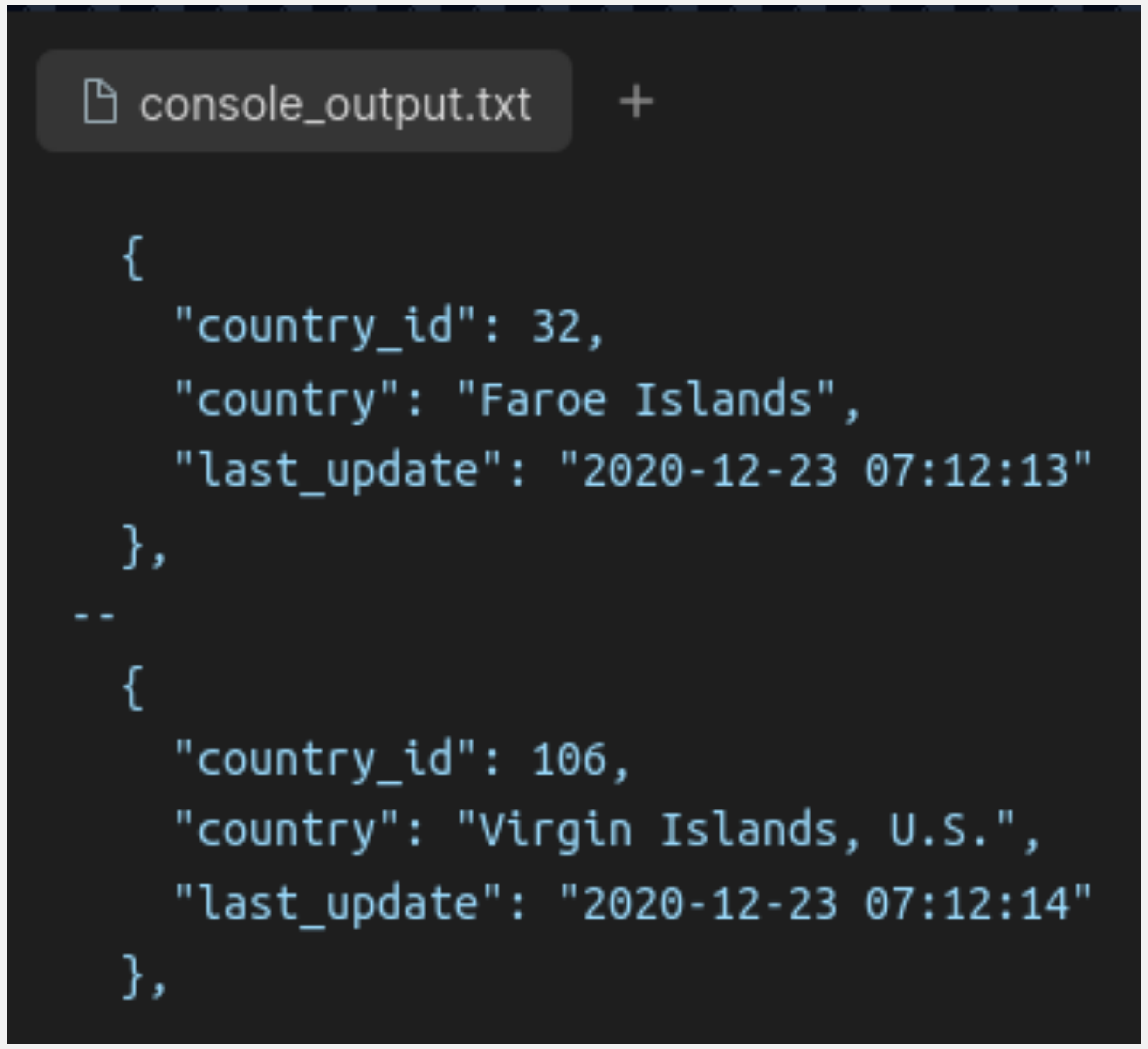
More examples



scnr_scan_to_grep.sh

```
scnr scan -i _samples -f *w.tar.gz/*.db | grep -B 2 -A 2 Islands
```

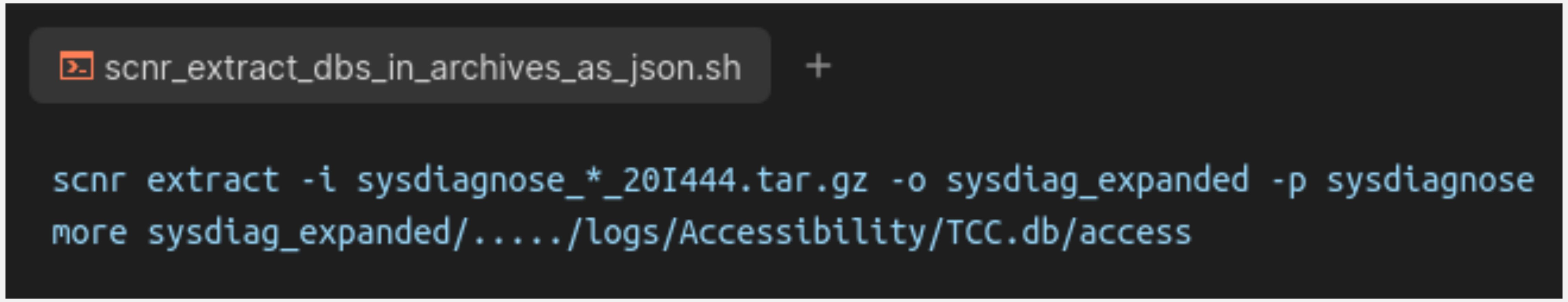
- grep through sqlite ?
- in an archive ?



console_output.txt

```
  {
    "country_id": 32,
    "country": "Faroe Islands",
    "last_update": "2020-12-23 07:12:13"
  },
--
  {
    "country_id": 106,
    "country": "Virgin Islands, U.S.",
    "last_update": "2020-12-23 07:12:14"
  },
```

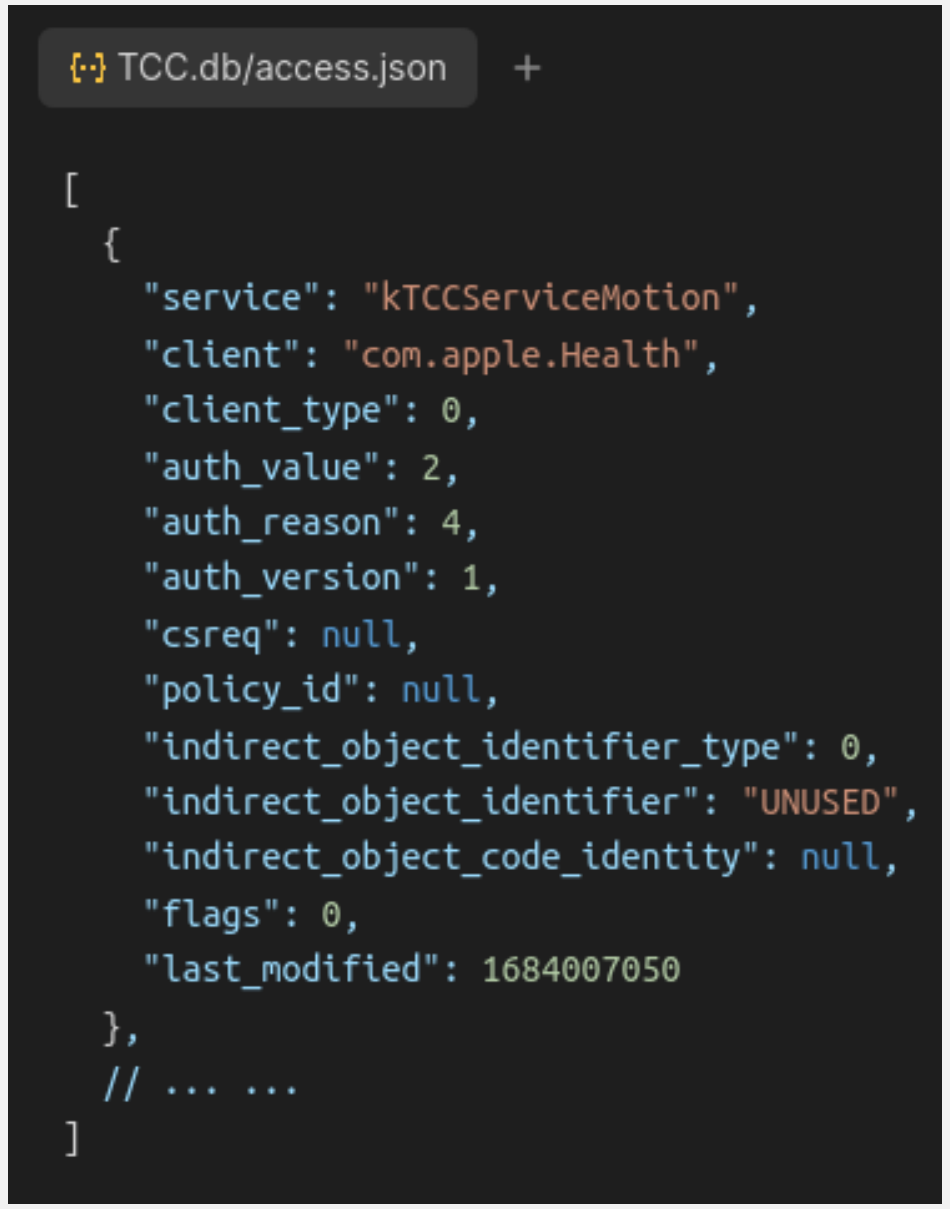
$ scnr extract



scnr_extract_dbs_in_archives_as_json.sh

```
scnr extract -i sysdiagnose_*_20I444.tar.gz -o sysdiag_expanded -p sysdiagnose
more sysdiag_expanded/...../logs/Accessibility/TCC.db/access
```

JUST EXTRACT PLIST OR DATABASE AS JSON
IT'S IN AN ARCHIVE, IN AN ARCHIVE ...
imgflip.com

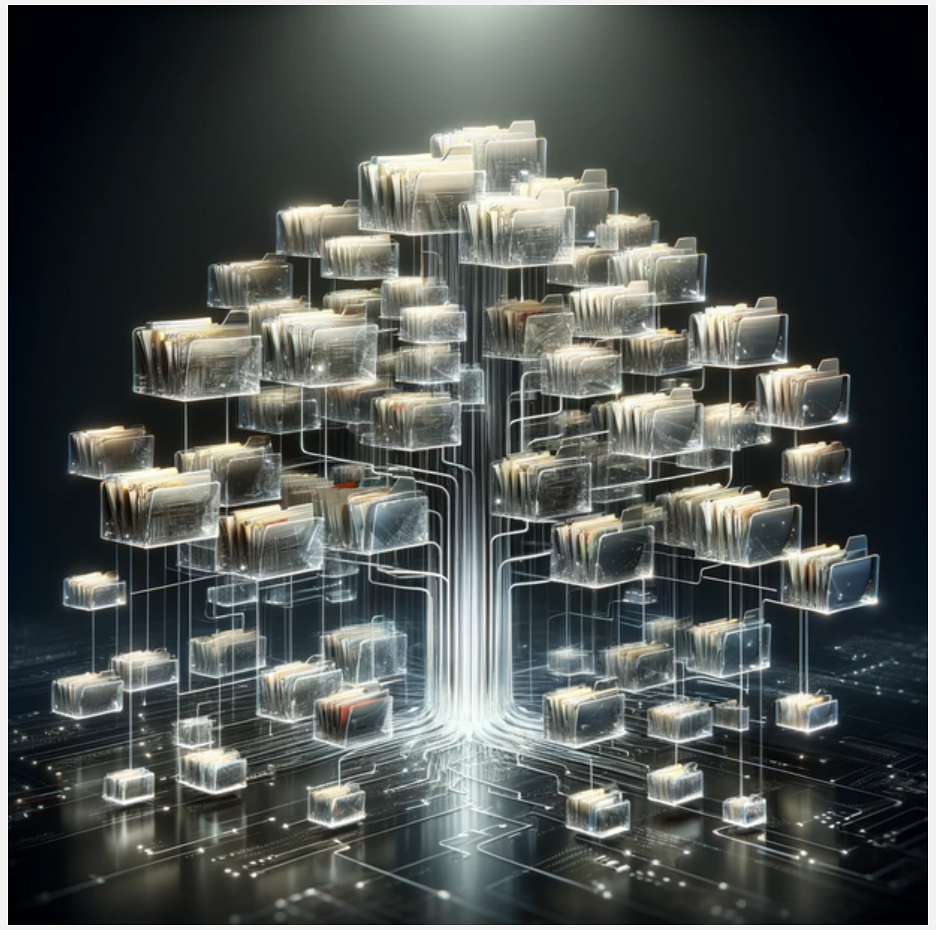


TCC.db/access.json

```
[
  {
    "service": "kTCCServiceMotion",
    "client": "com.apple.Health",
    "client_type": 0,
    "auth_value": 2,
    "auth_reason": 4,
    "auth_version": 1,
    "csreq": null,
    "policy_id": null,
    "indirect_object_identifier_type": 0,
    "indirect_object_identifier": "UNUSED",
    "indirect_object_code_identity": null,
    "flags": 0,
    "last_modified": 1684007050
  },
  // ... ...
]
```

$scnr wrap up

Archives transparency



Rust + Python libs

$ scnr scan ..

dumps json & txt to console

$ scnr jq ..

query each json and output the result

$ scnr extract ..

recursive extract & transform to json when possible

More to come ?

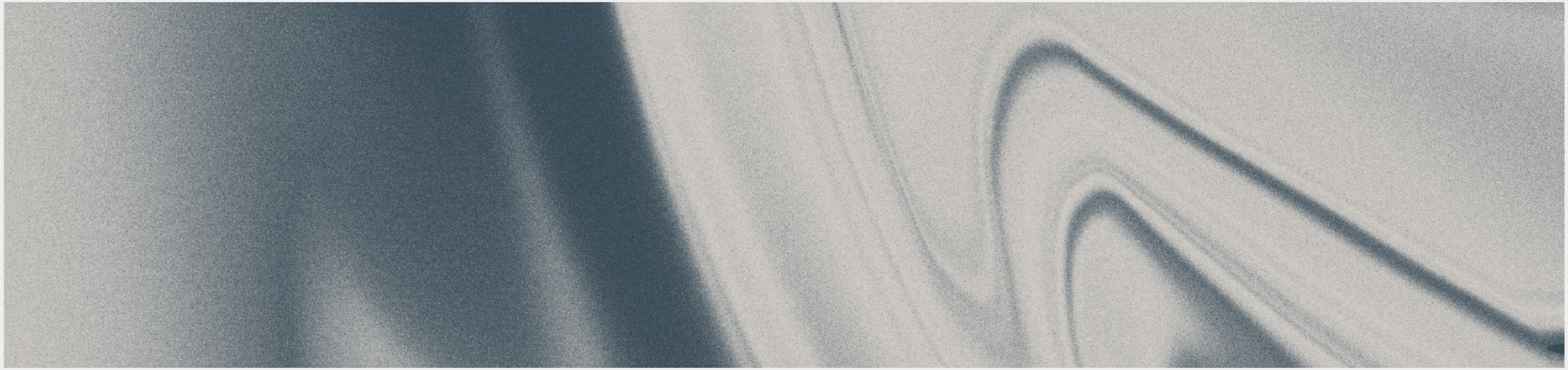
- performances ?
- more file types ?
- bindings in more languages ?
- more output types ?
- more query types ?

WTF section :

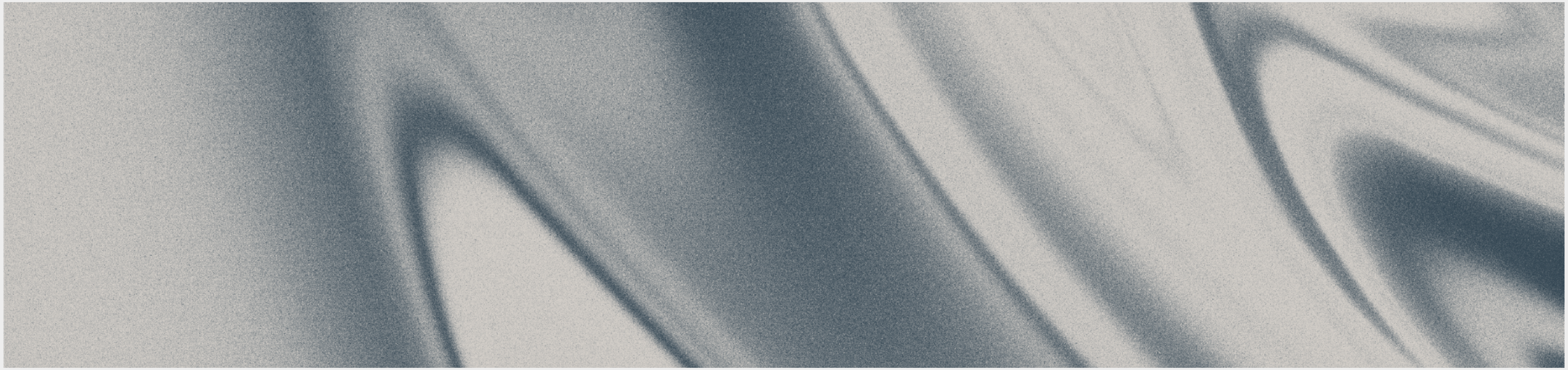
- DuckDB extension ?
- Graphql api ?
- in browser ? (WASM)

...and it's open, so **you** can contribute :

- use it, ask for use cases
- fork, improve, build your own ...
- issues & PR are welcome !

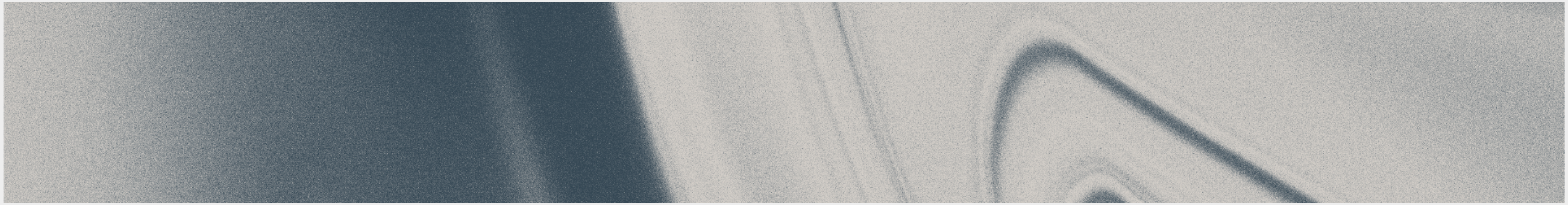


References & acknowledgments

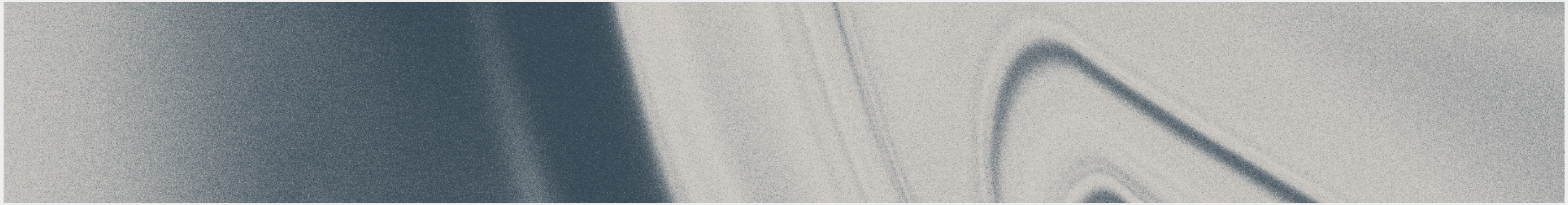


References & acknowledgments

Lib Mobile Device =>	https://libimobiledevice.org
iOSbackup =>	https://github.com/avibrazil/iOSbackup
SysDiagnose =>	http://www.for585.com/sysdiagnose
Operation Triangulation =>	https://securelist.com/?s=operation%20triangulation
Scnr =>	https://github.com/shindan-io/scnr
Shindan's blog =>	https://shindan.io/posts/
Tiny Check =>	https://tiny-check.com



Merci !



Questions ?

Retrouvez-nous sur notre stand ! le **A1**