Intro To Mobile Forensics

https://digitalcorpora.org/corpora/cell-phones/android-13-image/

[This is a set of cell phone images you can download to try out this workshop without a physical phone - Scan the QR to the right to be directed there.]

—

This workshop is designed to teach you how to check your phone for spyware and educate you on how much data is contained in your phone. **This workshop is not forensically sound and it will not hold up in legal proceedings or challenges.** This is an educational exercise.

This exercise will teach you to take an image of a device and use forensic tooling to analyze the device for spyware and other sensitive data leaks. If you follow along with these instructions, you'll be able to analyze your own phone and see your data footprint on the device and learn how to check for spyware!

> **What MVT is:** A command-line toolkit by Amnesty International Security Lab for consensual mobile forensics on iOS/Android. It can decrypt iOS backups, parse system/app logs, and compare findings against STIX2 Indicators of Compromise (IOCs). It's intended for trained technologists—not end-user self-diagnosis.
>
> **Ethics & consent:** Using MVT on devices without the owner's informed consent is prohibited by the project's license and contrary to the tool's purpose.
>
> **Keep in Mind:** This document is adapted using AI from the official documentation available at `docs.mvt.re`. If you run into issues, please check the documentation at this URL or find your workshop facilitator to ask questions.

Install MVT (macOS, Linux, or WSL)

Recommended: Install via `pipx` so MVT runs in an isolated environment and its commands are on your PATH.

macOS (Apple Silicon or Intel)

Install Xcode CLTs & Homebrew (if not already), then:

- `brew install python3 pipx sqlite3`
- `pipx ensurepath`
- `exec $SHELL -l`
- `pipx install mvt`

You should now have `mvt-ios` and `mvt-android` available.

Ubuntu/Debian (or WSL Ubuntu)

- `sudo apt update`
- `sudo apt install -y python3 python3-venv python3-pip sqlite3`
- `python3 -m pip install --user pipx`
- `python3 -m pipx ensurepath`
- `exec $SHELL -l`
- `pipx install mvt`

Windows users: MVT isn't fully supported natively; use **WSL** and follow the Linux steps above.

Acquire an iOS backup (encrypted recommended)

Encrypted backups include additional artifacts (e.g., Safari history/state) that improve analysis quality.

Option A — Finder (modern macOS)

- Connect iPhone via USB → open **Finder** → select device.
- General → Backups → "Back up all the data on your iPhone to this Mac" and check "Encrypt local backup."
- Set/remember the backup password → **Back Up Now**.
- Backups are stored in: `~/Library/Application Support/MobileSync/` (copy the latest folder for analysis).

Option B — iTunes (Windows or older macOS)

- Install **iTunes**, connect device, open iPhone view.
- Enable **Encrypted backup**, choose a password, start backup.
- Backup locations:
  - Windows: `%USERPROFILE%\Apple\MobileSync\`
  - Alternative: `%USERPROFILE%\AppData\Roaming\Apple Computer\MobileSync\`

Option C — libimobiledevice (Linux/macOS; CLI method)

- Install the utilities (may require HEAD build on macOS):
  - Linux: `sudo apt install libimobiledevice-utils`
  - macOS (via brew): `brew install --HEAD libimobiledevice`
- Ensure encryption is ON (recommended): `idevicebackup2 -i encryption on`
- Make the backup: `idevicebackup2 backup --full /path/to/backup/`

(If password issues arise, see `idevicebackup2 -i changepw`, or as a last resort Apple's **Reset All Settings**—note this removes some forensic traces; avoid if possible.)

Android Analysis (Limited Capabilities)

Important Note: MVT's Android capabilities are significantly more limited than iOS. Android backup analysis only supports SMS messages, not comprehensive device analysis.

Method 1: Android Backup (SMS Only)

1. **Install Android SDK Platform Tools:**
   - macOS: `brew install --cask android-platform-tools`
   - Ubuntu/Debian: `sudo apt install android-sdk-platform-tools-common`
2. **Enable USB Debugging:**
   - Go to **Settings** → **About phone** and tap **Build number** seven times
   - Go to **Settings** → **System** → **Developer options** and enable **USB debugging**
3. **Create SMS backup:**
   - Connect Android device via USB
   - Accept "Allow USB debugging" prompt
   - Create SMS backup only: `adb backup -nocompress com.android.providers.telephony`
   - For full backup (limited usefulness): `adb backup -all -nosystem`

Note: Some devices require a backup password. The `-nocompress` option avoids compression issues between Java and Python implementations.

Method 2: Direct Device Analysis (Recommended for Android)

For more comprehensive Android analysis, use direct device scanning:

- `mvt-android check-adb --output ./android_results`

This method provides more forensic data than backup analysis.

Decrypt iOS Backup (if encrypted)

Use MVT to decrypt into a separate directory:

Option 1: Provide password securely via environment variable

- `export MVT_IOS_BACKUP_PASSWORD="your-backup-password"`
- `mvt-ios decrypt-backup -d ~/ios/backup-decrypted "/path/to/original/backup"`

Option 2: Have MVT prompt for the password

- `mvt-ios decrypt-backup -d ~/ios/backup-decrypted "/path/to/original/backup"`

Optional: Extract and save a keyfile

- `mvt-ios extract-key -k ~/ios/backup.key "/path/to/original/backup"`
- **Then decrypt using the key:**
- `mvt-ios decrypt-backup -k ~/ios/backup.key -d ~/ios/backup-decrypted "/path/to/original/backup"`

Treat the key file as sensitive.

(Optional but recommended) Prepare IOCs

MVT can download public STIX2 IOCs and/or accept your own:

Pull public IOCs (Amnesty & community set)

- `mvt-ios download-iocs`
- `mvt-android download-iocs`

Use specific STIX2 files

- iOS: `mvt-ios check-backup --iocs ~/iocs/pegasus.stix2 --output ~/ios/results ~/ios/backup-decrypted`
- Android: `mvt-android check-backup --iocs ./iocs/pegasus.stix2 -o ./android_results ./android_backup.ab`

You can also apply IOCs later to previously extracted results with `mvt-ios check-iocs` and `mvt-android check-iocs`.

Analyze the Data

iOS Analysis

Run `check-backup` against the decrypted iOS backup:

- **Basic run (no IOCs)**: `mvt-ios check-backup --output ~/ios/results ~/ios/backup-decrypted`
- **With IOCs**: `mvt-ios check-backup --iocs ~/iocs/pegasus.stix2 --output ~/ios/results ~/ios/backup-decrypted`

Android Analysis

For Android Backup (SMS only):

- **Basic run**: `mvt-android check-backup -o ./android_results ./backup.ab`
- **With IOCs**: `mvt-android check-backup --iocs ./iocs/pegasus.stix2 -o ./android_results ./backup.ab`

For Direct Android Device Analysis (Recommended):

- **Basic scan**: `mvt-android check-adb -o ./android_results`
- **With IOCs**: `mvt-android check-adb --iocs ./iocs/pegasus.stix2 -o ./android_results`

Useful helpers

- `mvt-ios check-backup --list-modules`
- `mvt-ios check-backup --module SafariHistory --output ~/ios/results ~/ios/backup-decrypted`
- `mvt-ios check-backup --fast --output ~/ios/results ~/ios/backup-decrypted`

During analysis, matches to IOCs are highlighted and saved as `*_detected.json` alongside other JSON outputs.

Interpreting Results

iOS Results

`~/ios/results/` will contain structured JSON files such as: `safari_history.json`, `analytics.json`, `datausage.json`, `applications.json`, `sms.json`, `whatsapp.json`, etc.

Android Results

Android results are more limited but may include: `sms.json` (from backup), `packages.json`, `dumpsys.json`, and other system information depending on the analysis method used.

If IOCs were used, corresponding `*_detected.json` files list hits. A full catalog of what each JSON file contains and where data originates is documented at `docs.mvt.re`.

Important: Detection of IOCs indicates potential compromise, but absence of detections does not guarantee the device is clean. Advanced spyware may use techniques not covered by public IOCs.

Troubleshooting & Notes

- **`mvt-ios: command not found`**: Ensure you ran `pipx ensurepath` and opened a new shell.
- **Windows**: Use **WSL** with Ubuntu (native Windows has known issues, especially for Android).
- **Encrypted backup password forgotten**: You may try changing via `idevicebackup2 -i changepw` or disabling/re-enabling encryption; avoid "Reset All Settings" unless it's a last resort (it wipes some forensic traces).
- **Android backup limitations**: Remember that MVT Android backup analysis only covers SMS messages. For comprehensive Android analysis, use `mvt-android check-adb` instead.
- **Backup password for Android**: Some devices enforce backup passwords. Use `--backup-password` option or `MVT_ANDROID_BACKUP_PASSWORD` environment variable.

Important Limitations

iOS Analysis

- Requires encrypted backup for comprehensive analysis
- Most effective forensic capabilities
- Can analyze system logs, app data, and various artifacts

Android Analysis

- **Backup method**: Limited to SMS messages only
- **Direct ADB method**: More comprehensive but requires USB debugging
- Generally less forensic data available compared to iOS
- No root access means some advanced artifacts are inaccessible

Detection Limitations

- MVT only detects known indicators of compromise (IOCs)
- Public IOCs may not include the latest threats
- False negatives are possible - absence of detection doesn't guarantee safety
- For comprehensive analysis, consider professional forensic services

Attributions & Documentation

- **MVT docs home**: `docs.mvt.re`
- **Install MVT with pipx**: `docs.mvt.re/en/latest/install/`
- **iOS methodology & why encrypted backups help**: `docs.mvt.re/en/latest/ios/`
- **Back up with Finder/iTunes & paths**: `docs.mvt.re/en/latest/ios/backup/`
- **Decrypt, extract key, and check-backup usage**: `docs.mvt.re/en/latest/ios/`
- **Android methodology and limitations**: `docs.mvt.re/en/latest/android/`
- **IOCs (STIX2), sources, `download-iocs`, `check-iocs`**: `docs.mvt.re/en/latest/iocs/`
- **Outputs reference (what each JSON file contains)**: `docs.mvt.re/en/latest/`

License & Consent Reminder

Use MVT only for consensual forensic work. If you need expert assistance interpreting results or non-public IOCs, contact Amnesty Security Lab or Access Now's Digital Security Helpline.