

User Guide

Digital Forensics & Investigation Platform
Version 1.7.0-beta
Developed by Rodel Plasabas
November 2026

Contents

Introduction
Installing VESTIGEX
- What you need
- Choosing an installer
- Install
- Upgrading
- Uninstalling
- Linux

Getting started
- Signing in
- Examiner accounts
- Creating or opening a case
- The main window

Adding and processing evidence
- Adding evidence
- Acquiring a drive
- Backing up an iPhone or iPad
- Collecting an Android phone
- Collecting remote computers
- Processing evidence
- Jobs
- Re-extracting artifacts

Browsing evidence
- Data Sources
- File System
- The file viewer
- File Types
- bulk_extractor
- Deleted & Carved
- Tables, sorting and filters

Searching
- Keyword Search
- Hash Sets
- YARA Rules

Analysis
- Timeline
- Connections
- Media Analysis
- AI Investigation

Artifacts
- Registry, OS Artifacts and Web History
- Messaging
- Email

- Memory
- Network
- Cloud: Facebook, Instagram and WhatsApp
- Mobile
- SQLite Databases
- Credentials

Threat hunt

Review and reporting

- Bookmarks and review status
- Refined Results
- Reports
- Audit Log and chain of custody

Settings

- Setting up the local AI
- Using an online AI service

Troubleshooting

Where things are kept

Introduction

VESTIGEX is a digital forensics and investigation platform for Windows. It opens disk images, triage collections, phone extractions, memory dumps and log files, extracts what the evidence holds (files, deleted files, registry, browser history, chats, e-mail, photos and more) and helps you search, review and report on it. Everything runs on your own computer: no evidence, search or AI question leaves the machine.

> Results produced automatically (detections, picture categories, AI answers, translations, scores) are leads for an examiner to review, never conclusions on their own. Always confirm them against the evidence before you rely on them.

Installing VESTIGEX

What you need

- Windows 10 (version 1809 or later) or Windows 11, 64-bit.
- About 12 GB of free disk space for the program and its models, plus room for your cases (a case folder can grow to a sizable fraction of the evidence it holds).
- 16 GB of RAM or more is recommended; 32 GB for large cases.
- An NVIDIA graphics card is optional. With one, photo scanning, OCR and speech-to-text run on the GPU; without one they run on the processor, more slowly.
- Windows 11 includes the Edge WebView2 runtime that VESTIGEX uses for reports. Windows 10 may need it installed from Microsoft.

Choosing an installer

Package	Use it when	What it does
Offline	The computer has no (or slow) internet, or you want everything in one download	Carries everything: the program, every model and library, and the Ollama installer. Nothing is downloaded during setup.
Online	The computer has internet and you want a smaller download	Carries the program; setup then downloads the speech and picture models, the CUDA libraries for speech on NVIDIA GPUs and Ollama in a window that shows each download (name, size, progress, speed, status). Untick what you do not want; anything skipped or failed can be fetched later under **Settings > Components > Setup downloads**.
Online updater (VESTIGEX-Online-Updater.zip)	VESTIGEX is already installed, at any earlier version	Compares the installation with the current release and downloads only the files that differ. From 1.5.0-beta VESTIGEX does this itself (see Upgrading).

Every package has a **SHA256SUMS.txt** (and MD5SUMS.txt). Setup checks the package against it before installing, and every file the online installer downloads is checked against the SHA-256 recorded when the installer was built; a file that does not match is downloaded again.

Install

1. Unzip the package and keep its files together in one folder.
2. Close VESTIGEX if an earlier version is open.
3. Double-click **Install VESTIGEX.cmd** and answer **Y** to install.
4. VESTIGEX unpacks into `%LOCALAPPDATA%\Programs\VESTIGEX` (several minutes). No administrator rights are needed.
5. Offline package: choose whether to install Ollama for the local AI (it is in the package). Online package: the **VESTIGEX Setup** window lists what will be downloaded; press **Start**, then **Finish** when it is done. Then choose and download an AI model under **Settings > AI assistant** (see Setting up

the local AI).

Setup adds **VESTIGEX** to the Start menu and the desktop, a **VESTIGEX logs** shortcut and an entry in Windows **Settings > Apps**. All packages work with Windows Smart App Control turned on.

Upgrading

From inside VESTIGEX (versions with the built-in updater): when VESTIGEX starts it asks the update server which version is current. If there is a newer one, a dialog names it, shows what is new in it and asks whether to update. An administrator presses **Update now**: VESTIGEX compares every program file with the new release, downloads only the files that differ, then closes, puts the new files in and opens again. After the restart a dialog shows the new version and the changelog of everything that came with the update. **Later** leaves VESTIGEX as it is until the next start. During the download, **Continue working** sends it to the background; it then waits at the bottom right with **Restart to update**, so you can finish what is running first (the open case is closed by the restart). If tasks are still running when the download is done, VESTIGEX asks before stopping them. Every downloaded file is checked against the release's signed SHA-256 list, the replaced files are kept in the `.update` folder of the installation, and a failed update puts the earlier version back. **Settings > Updates** has **Check for updates**, **Check files and repair** (downloads program files that are missing or damaged) and the switch **Check for a newer version when VESTIGEX starts**; turn it off on a workstation that must not contact the internet. The check only asks for the current version number; nothing about the installation or its cases is sent.

With the online updater (any earlier version, on Windows and Linux): download **VESTIGEX-Online-Updater.zip**, unzip it and double-click **Update VESTIGEX Online.cmd** (Linux: `./update-online.sh`). It does the same comparison and download from a console window, from whichever version is installed, and also repairs an installation an earlier update left incomplete. Add the installation folder if it is not the default one, for example `"Update VESTIGEX Online.cmd" D:\Apps\VESTIGEX`.

Both leave alone the data you or **Settings > Components** changed, and the large optional components (Whisper, CLIP, the CUDA 12 libraries), which keep coming from **Settings > Components**.

Or run a newer version's **Install VESTIGEX.cmd** (offline or online). Either way your user accounts, AI and processing settings, hash sets and organization logo are kept. Cases are stored wherever you created them and are never touched by setup.

Uninstalling

Use **Settings > Apps > VESTIGEX > Uninstall** or the **Uninstall VESTIGEX** Start menu shortcut. You are asked whether to keep or delete the user accounts and settings. Cases are never deleted.

Linux

VESTIGEX also comes as Linux packages (64-bit Intel / AMD): **VESTIGEX-<version>-Linux-Offline.tar.gz** and **-Linux-Online.tar.gz**. They hold the same program with Linux builds of its Python, window and tools, and install for your user only, without root.

- **What you need**: a 64-bit desktop distribution from 2020 or later (glibc 2.28+, for example Ubuntu 20.04+, Debian 11+, Fedora 32+), about 12 GB of free space, and the same memory as on Windows. An NVIDIA GPU needs the NVIDIA driver installed.

- **Install**: unpack the package (`tar -xzf VESTIGEX-<version>-Linux-Offline.tar.gz`), then run `./install.sh` in its folder and answer **y**. VESTIGEX goes to `~/.local/share/vestigex` (or the folder given to install.sh) and appears in the applications menu; the command `~/.local/bin/vestigex` starts it too. The online package then shows the same download window as on Windows.
- **Upgrade**: from inside VESTIGEX (1.5.0-beta and later), or unpack VESTIGEX-Online-Updater.zip and run `./update-online.sh` (or install a newer Offline / Online package over the old one). Accounts, settings and hash sets are kept, as on Windows.
- **Uninstall**: run `~/.local/share/vestigex/desktop/uninstall.sh`.
- **Local AI**: install Ollama from ollama.com (`curl -fsSL https://ollama.com/install.sh | sh`), then choose a model under **Settings > AI assistant**. The Linux packages do not carry Ollama.
- **Not in the Linux version**: Velociraptor remote collectors and drive acquisition, which are Windows-only. **bulk_extractor**: install your distribution's `bulk-extractor` package (VESTIGEX finds it), or import a Linux build under **Settings > Components**.
- On distributions that do not allow Chromium's sandbox for programs installed per user (Ubuntu 23.10 and later), the VESTIGEX window runs without it. It only ever shows VESTIGEX's own pages from 127.0.0.1.

Getting started

Signing in

VESTIGEX opens on a sign-in screen. A new installation has one administrator account:

- User name: **admin**
- Password: **admin**

You must choose a new password the first time you sign in. Sessions end after 12 hours without activity, and five wrong passwords lock an account name for one minute.

Use the **user menu** (your name in the top bar) to change your password or sign out. Signing out closes the open case and asks first if background work is still running.

Examiner accounts

Administrators manage accounts in **Settings > Users** (or the **Users** button on the open-case screen):

- **Create** an account with a temporary password; the examiner sets their own at first sign-in.
- Change the display name or the role (**Examiner** or **Administrator**).
- **Reset** a password (signs that user out everywhere), or **disable** and enable an account. At least one active administrator is always kept.
- **Sign-in activity** lists successful and failed sign-ins and account changes with the source address.

The signed-in examiner's name fills the Examiner field of new cases and is written into every chain-of-custody entry.

Creating or opening a case

After signing in you choose a case:

- **Create New Case**: give the case a name, the examiner, organization, description, the **case type** and a folder. VESTIGEX creates `case.vestigex` and the case database `vestigex.db` in that folder.
- **Open Existing Case**: pick a case folder or its `case.vestigex` file.

The **case type** decides which checks **Refined Results** runs: Compromise assessment / incident response, Insider data theft, Employee misconduct, Fraud & financial crime, Harassment, threats & cyberbullying, Sexual exploitation & voyeurism, Violence & threats to life, Litigation & e-discovery, or General. You can change it later in **Case Management > Edit Case**.

When a case opens, VESTIGEX compares each data source's files, sizes and modified times with what was recorded at intake and warns if evidence changed or is missing. If you moved or renamed the evidence, click **Relocate...** on its row in the Evidence register and pick where it is now (the file, or the folder for folder evidence). VESTIGEX hashes it there and uses the new location only if the MD5 and SHA-256 match the ones taken at intake; otherwise the recorded path stays. The audit log records the old and new path and the result. Nothing has to be processed again.

The **chain of custody log** is hash-chained: each entry includes the hash of the one before, so changing or removing an entry breaks the chain from there on. The number and hash of the latest entry are also kept outside the case database, in the case folder (audit_anchor.json) and in this workstation's VESTIGEX data folder; the log's end must match them, so removing the latest entries, or rewriting the whole log, shows as a failed check too. The report prints the latest entry's number and hash: keep it, and a later log that does not reach that entry with that hash was changed.

The main window

- **Sidebar**: every screen, grouped as Overview, Evidence, Search, Analysis, Artifacts, Threat Hunt and Reporting. The ☰ button collapses it.
- **Top bar**: the open case, counters (artifacts, events, findings, critical/high, bookmarked), the **date range**, the **data source filter**, **★ snapshots**, your user menu and **Close Case**.
- **Status bar** (bottom): backend health, index and hash status, running jobs, the AI model, and the version. Click the version to open **About VESTIGEX**.

The **date range** limits findings, the timeline, artifacts, lateral movement and processes to an inclusive range, in the display time zone. The **data source filter** shows only the evidence of the ticked sources. A **snapshot** saves the current date range and source filter under a name (for example "Initial access window") so anyone on the case can return to it.

Adding and processing evidence

Adding evidence

Go to **Evidence > Data Sources** and press **+ Add evidence**. The wizard asks three things:

1. **Category**: Computer, Mobile, Memory, Network or Logs.
2. **Platform**: Windows, Mac, Linux, Android or iOS.
3. **Evidence type**, for example:

Category	Evidence types
Computer	Disk image (E01, raw/dd, VHD/VHDX, VMDK, DMG ...), files and folders, KAPE / Velociraptor / UAC collection, Mac logical .zip
Mobile	Full file system (zip / tar / tar.gz / folder, from Cellebrite, GrayKey, Magnet, Elcomsoft ...; a Cellebrite .ufd / .ufdx or its output folder leads to the archive and adds the device details), iTunes / Finder backup, Cellebrite UFDR
Memory	RAM dump (.mem, .raw, .vmem, .dmp), page file, swap file, hiberfil.sys
Cloud	Facebook or Instagram "Download Your Information" export (JSON or HTML format, .zip or unpacked folder), Meta warrant return (Meta Platforms Business Record: .zip or folder with records.html), WhatsApp chat export (.zip with media, or the .txt) and account information report, or a folder of them
Network	Packet capture (.pcap, .pcapng) from Wireshark, tcpdump, dumpcap, a sensor or a firewall, or a Windows .etl trace from pktmon or netsh trace (capture=yes)
Logs	Windows event logs, web server logs, Plaso timelines and other log files

Web server access logs inside disk images and triage folders are found and parsed during Process Evidence, with no need to export them: IIS (inetpub, W3SVC), Apache (including XAMPP: `xampp\apache\logs`, and `/opt/lampp/logs` on Linux), Red Hat httpd (`access_log`), nginx, their rotated copies (`access.log.1`, `access_log-20240101`) and gzip-compressed rotations (`access.log.2.gz`). A file is kept only when it holds web requests.

The file picker only shows matching files. Before adding, VESTIGEX opens the evidence and tells you what it found (for example "macOS 15.4 (24E248) · Alexs-MacBook-Pro") and warns if it is not the platform you chose. Give each source a clear label, usually the host or device name.

Encrypted **iTunes / Finder backups** ask for the backup password when added; it is checked and kept in memory only. A disk image with a locked **BitLocker** volume stops within seconds of starting Process Evidence, before the image is hashed, with an unlock box on the data source: enter the recovery key (eight groups of six digits) or the password, and VESTIGEX checks it and queues the image again.

A data source can be **renamed** at any time and **removed** only before it is processed (or after processing failed).

Acquiring a drive

+ Acquire (Data Sources) images a drive or USB drive connected to this computer into the case, backs up an iPhone or iPad, collects an Android phone, or builds a collector for remote computers (below).

1. **Authority**: the legal basis (search warrant, court order, consent, company policy), its reference and who authorized it are required; custodian, owner and notes are optional. They are stored with the acquisition and in the audit log.
2. **Source**: a whole drive or one of its volumes. Drives that report themselves read-only (as a hardware write blocker makes them) are marked; for others VESTIGEX warns that no write blocker was detected. The Windows system drive and the drive holding the case ask for confirmation.
3. **Options**: E01 (compression none / fast / best, segments of 650 MB, 1.5 GB or 2 GB, with case number, evidence number, description, examiner and notes in its header) or raw (dd, with a hash file); where to save it (not on the drive being imaged).
4. **Start**: Windows asks for administrator permission, which reading a drive directly requires; only the imaging process gets it. Progress, speed and time left show on Data Sources and in Jobs.

MD5, SHA-1 and SHA-256 are computed while reading. Sectors that cannot be read are retried, then written as zeros and listed. When imaging ends, the image is read back and its structure and hashes checked; only a verified image is added to the case as a data source, ready to process. Each acquisition gets a plain-text report beside the image (authority, device model and serial, write protection, hashes, verification, unreadable sectors) and a line under **Acquisitions** on Data Sources, in the HTML / PDF report (Chain of custody) and in the Excel report.

Backing up an iPhone or iPad

+ Acquire → iPhone / iPad makes an encrypted backup of the device over USB, as iTunes or Finder would, and adds it to the case for iLEAPP. It holds messages, call history, contacts, photo records, app data that apps allow into backups, health, Wi-Fi and location records; not a full file system or the keychain. Apple's Mobile Device service must be installed on this computer (the Apple Devices app from the Microsoft Store, or iTunes).

1. **Authority**, as for drives.
2. **Device**: connect it with its cable and keep it unlocked. Devices that do not trust this computer yet will ask the person holding them to tap **Trust** and enter the passcode.
3. **Backup password**: iOS puts saved passwords, health and Wi-Fi data only into encrypted backups, so the backup is always encrypted. If the device has no backup password, enter a new one; the device asks for its passcode to switch encryption on, and by default VESTIGEX switches it off again afterwards (one more passcode prompt), leaving the device as it was. If the device already has a backup password, enter it; an unknown one can only be removed on the device with Reset All Settings, which changes the device.
4. **Start**: progress and what the device is waiting for show on Data Sources and in Jobs.

Every file of the backup is hashed (MD5 and SHA-256) into a file list saved beside it (EV-….files.csv); the hashes of that list stand for the backup. The files are then read again and compared, and the password is tested on the finished backup; only then is it added to the case. The acquisition report lists the device

(model, iOS, serial, UDID, IMEI, phone number) and every change made to it: pairing with this computer, and backup encryption switched on and off. The backup password is kept with the case, because iLEAPP needs it each time the backup is processed; it is never written to the report or the audit log.

Collecting an Android phone

+ Acquire → **Android** makes a logical collection of the phone over USB with Android's adb, and adds it to the case for ALEAPP.

1. **Authority**, as for drives.
2. **Phone**: switch on USB debugging (Settings › About phone: tap Build number 7 times; then Developer options › USB debugging), connect the cable, unlock the phone and tap **Allow** on "Allow USB debugging?". The wizard shows these steps; a phone still waiting for Allow is marked, and the collection waits for it.
3. **Options**: evidence number, label, where to save it, whether to skip Android/data and Android/obb (app caches and game data, often most of the size), and whether to also request an Android backup (the phone asks to confirm; leave its password empty).

What is collected: shared storage (photos, downloads, WhatsApp media and backups …) with its timestamps; the phone's properties, installed apps with their versions and install times, accounts, usage stats, Wi-Fi, Bluetooth, location, battery history, notifications and settings; the call log, SMS and contacts where the phone lets them be read. Most app databases (chats, and call logs on many phones) are private to their apps and need root or a full file system extraction with a commercial tool; VESTIGEX imports Cellebrite UFDR for those.

Nothing is installed on the phone. Every command sent to it is written, with its time and result, to the adb log beside the collection (EV-….adb-log.txt). Shared storage is copied as one file, shared_storage.tar, made by the phone's own tar: files the phone will not let it read (other apps' private folders) are skipped instead of stopping the copy, and long or unusual file names do not matter. Every collected file is hashed (MD5 and SHA-256) into a file list (EV-….files.csv) and read again to check; files the phone listed but that are not in the copy are counted in the report and listed in the adb log, and a copy with no files fails.

Collecting remote computers

+ Acquire → **Remote computers** builds a Velociraptor offline collector for this case; the client's IT team runs it on each computer, and the results come back to the case.

1. **Authority**, as for drives.
2. **What to collect**: a triage preset (Incident response, SANS triage, Ransomware, Insider case, Basic) and/or single items (event logs, registry hives, $MFT, $UsnJrnl, Prefetch, Amcache, SRUM, browsers, scheduled tasks, PowerShell history, LNK files, Recycle Bin, USB logs …), and optionally a memory capture.
3. **Where the results go**: beside the collector (brought back on a drive), a Windows share, SFTP or S3. Upload details are stored inside the collector, so use an account that can only write there.
4. **Build**: the collector is saved in the case folder under remote\<name>, with a zip of it. It is Velociraptor's official, signed velociraptor.exe plus a collector file; Run-Collector.cmd runs them (right-click › Run as administrator). README.txt has the command lines for PsExec, Group Policy, Intune and SCCM.

Results are always encrypted to this case's certificate, so a collection intercepted on the way cannot be read, and only this case can open it. The collector, its targets, the authority and the hashes of every file are recorded under **Remote collectors** on Data Sources and in the audit log.

When collections come back, use **Import remote collection** (Data Sources) on each Collection-….zip, with the .log of the same name beside it. VESTIGEX checks the zip was made for this case and that its SHA-256 matches the hash the collector wrote in its log; a zip that does not match is refused. It then decrypts it (checking its integrity code), keeps the received files, and adds the collection as a data source named after the computer, ready to process. A memory capture in it is added as its own memory source.

Processing evidence

Press **Process Evidence**. A dialog lets you choose what runs, showing only the steps that apply to the waiting sources:

- File hashing and hash set lookup, deleted files, the keyword index
- Event and web logs, threat hunt, YARA
- Registry, OS artifacts, web history, e-mail, photo EXIF, messaging apps, encrypted files, stored credentials
- Volume Shadow Copies, page files, memory analysis
- Media: speech-to-text and OCR
- macOS: mac_apt artifacts and (optionally) Unified Logs

Presets: **Everything**, **Suggested for this case type** and **Quick triage**. Your last choice is remembered and recorded in the audit log. Phone extractions always run in full.

Before reading a source, VESTIGEX hashes it (MD5 and SHA-256) for the chain of custody. Disk images are hashed as one stream over all segments, matching imaging tools. If the evidence cannot be read to the end for this hash, the source is not processed: it is marked as an error and the failure is written to the audit log.

For an **E01** (from FTK Imager, EnCase, libewf or VESTIGEX), VESTIGEX also reads the acquisition hash the imaging tool stored in the image (MD5, and SHA-1 where present) and recomputes it over the media inside the E01. The evidence register shows **E01 acquisition hash: ✓ matches the media**, **MISMATCH**, or **none stored in the image**, and the result is in the audit log and the report. The E01 files' own MD5 and SHA-256 (the intake hash) differ from this media hash: they prove the files are unchanged, while the acquisition hash ties the image to the device it was made from. E01s processed before this check have it done the next time you press **Verify**.

Processing runs in the background. Several data sources are processed at the same time, and within each source files are hashed and have their text extracted several at a time (8 by default), also inside disk images and other containers: each worker opens the evidence itself. Both numbers are set in **Settings > Processing**.

Jobs

Overview > Jobs (also from the status bar) lists every background job with a progress bar, the current step, elapsed time and a log of its steps. Processing shows one sub-job per data source. **Cancel** stops a job; cancelling one data source sends just that source back to pending.

Re-extracting artifacts

Every artifact screen has a **Re-extract artifacts** button for sources processed by an earlier version, so improved parsers can be applied without processing everything again.

Browsing evidence

Data Sources

Each source shows its type, platform, status, file counts and evidence hashes. **Verify** re-hashes the evidence and compares (verified, HASH MISMATCH or missing). **System information** shows what was read from the source: computer name, operating system and build, time zone, users, network interfaces and known Wi-Fi networks (Windows, macOS and Linux).

File System

A folder tree of every volume in every source, with the files of the selected folder in a table. Volume Shadow Copies (when processed) appear as their own volumes labelled with their creation date.

The file viewer

Selecting a file opens its detail pane. Depending on the file, it has these tabs:

- **Hex** and **Text**: the raw content.
- **Extracted text**: the text indexed for keyword search.
- **Metadata**: names, sizes, MACB times, hashes, signature, hash set and YARA matches, and the review panel.
- **Document**: PDF, Word (with tracked changes and comments), Excel (with hidden sheets flagged), PowerPoint (with speaker notes), CSV, HTML, RTF and property lists. Everything except PDF is rendered by VESTIGEX in a sandbox that blocks scripts and network access.
- **Preview**: pictures, including HEIC/HEIF and AVIF. Photos with GPS show their location; the map loads only when you ask, because loading it contacts OpenStreetMap.
- **Video** / **Audio**: a player that streams the file read-only from the evidence.
- **Video check**: what the video container records (camera, editing software, creation time, GPS, structural oddities).
- **Registry**: browse a registry hive's keys and values.

Export file saves a copy; the export is recorded in the audit log. ☆ **Bookmark** adds the file to the case's bookmarks.

File Types

Files grouped by what they are (pictures, videos, audio, documents, archives, executables …), identified by their content signature, not their extension. A mismatch between extension and content is flagged. Pictures and videos have a **Gallery** view: thumbnail sizes S / M / L, arrow keys move, **B** bookmarks the selected item.

bulk_extractor

On a disk image, memory image or folder in **Data Sources**, **bulk_extractor** runs bulk_extractor (install it under Settings > Components) with a profile:

- **Network & IOCs**: e-mail addresses, domains, URLs, search terms, IP and MAC addresses, carved network packets, HTTP logs, JSON.
- **Personal data**: credit card and telephone numbers and other personal data, e-mail addresses, EXIF and GPS data, vCards.
- **Encryption keys**: AES keys in memory (BitLocker, TrueCrypt / VeraCrypt ...).
- **Search terms & wordlist**: what was searched for, and every word in the evidence (for password cracking).
- **Full**: every scanner bulk_extractor runs by default.

It reads every byte, unallocated space and slack included, and decodes ZIP, GZIP, PDF, base64 and Office XML as it goes. Results are on **Artifacts** → **Bulk Extractor**: each feature with its offset, or its forensic path (`20971520-GZIP-312`: found 312 bytes into GZIP data at offset 20,971,520), and context, plus histograms of how often each value appears. Carved network packets are added as a capture data source; process it for the Network page. bulk_extractor's own output (feature files, histograms, report.xml) stays in the case folder under `bulk_extractor`, and its log under `logs`.

Deleted & Carved

- **Deleted**: files whose MFT records (NTFS) or directory entries (FAT/exFAT) are marked deleted.
- **Carved**: press Carve to recover files from unallocated space (all unallocated space by default, or free space in file systems only), from a memory image, or from an image with no readable file system: - pictures: JPEG, PNG, GIF, BMP, WebP, HEIC / AVIF - video: MP4, MOV, 3GP, M4V, AVI, MKV, WebM, WMV, FLV - audio: MP3, WAV, M4A, AMR, OGG / Opus, WMA - documents: PDF, DOCX, XLSX, PPTX, ODT / ODS / ODP, DOC, XLS, PPT, RTF - archives: ZIP, 7z, RAR, GZIP; SQLite databases; Outlook PST / OST and MSG - Windows: event logs (whole .evtx files, and loose event log chunks rebuilt into a log), registry hives, LNK shortcuts, Prefetch; programs (EXE, DLL, SYS, ELF, stored as .bin)

Each file's length is read from its own structure, and anything inside a carved file (a thumbnail, a ZIP member) is not carved twice. Every carved file is checked (does it open, play or parse) and listed under OS Artifacts > Carved files with its type, size, where it was found (area and disk offset) and the result of the check. Carved files become a "(carved)" data source and are hashed, typed and indexed like any other file. A file whose clusters were scattered cannot be put back together: it is left out, or marked as not opening. Deleted NTFS MFT records found in free space are listed under OS Artifacts > Carved MFT records.

Tables, sorting and filters

All file and artifact tables work the same way:

- Click a column header to sort; click again to reverse, a third time to clear.
- Type in the filter row under the headers to filter that column. Filters run on the whole case, not only the rows on screen.
- Drag a header edge to resize a column; double-click it to reset. Widths are remembered.

Filter	Examples
Text	`invoice` (contains)
Size	`> 1 MB`, `1MB-5MB`
Date	`2024-05`, `> 2023`, `2024-01..2024-03`
Deleted	Yes / No

Searching

Keyword Search

Search every file's text and every log event. Choose **Words**, **Exact phrase** or **Regex**. **Keyword lists** run a saved list of terms (for example the built-in e-mail address pattern) across the case; **Save and run** stores a list for later. If the index has not been built yet, **Index now** builds it.

Hash Sets

Import known-file and alert hash sets (MD5, SHA-1 or SHA-256 lists, NSRL RDSv3 SQLite databases, and Project VIC / CAID VICS JSON exports with their categories). Matching files are listed per set and marked in their metadata. Hash sets are shared by every case on this installation.

YARA Rules

A rule library shared by all cases: import a `.yar` file or a folder of rules, or write rules in place (each is checked before saving), and switch rule files on or off. Enabled rules run during Process Evidence and on carved files; **Scan case files** rescans the whole case. Matches show the rule, severity, matched strings and offsets.

Analysis

Timeline

Every log event, parsed artifact, network record and file system time in one table, built at the end of processing. Filter by source type, host, text and "detections only", or with the column filters; the histogram shows activity per day (or per hour) and clicking a bar zooms in. Rows open the event, artifact or file in place, and log events can be bookmarked from their pop-up.

A **USB storage device** has a row for each time Windows recorded for it in the registry: **USB first installed**, **USB first connected**, **USB last connected** and **USB last removed** (the last two from Windows 8 on); times that are the same share one row. Type `USB` in the Type column filter to list them. Windows keeps only the first and the last connection of a device; connections in between are found in the event logs, where they were kept.

Times on the timeline are UTC: a value stored with an offset (an Apache log line at 13:55:36 +08:00) is converted (05:55:36 UTC), and a value stored without one is shown as UTC **by assumption**: such rows carry an orange **assumed UTC** badge until you decide that source's zone, and the timeline's **All times / Only assumed UTC / Hide assumed UTC** filter shows or hides them. Above the table, a notice lists the sources with assumed-UTC entries, how many each has, and the zone that source's own system was set to (on Windows, the registry's TimeZoneInformation) as a suggestion. **Decide zone…** opens the same settings as **Data Sources** → **Times**:

- **Time offset (clock skew)**: added to every time of that source, file system times included (for example +00:03:12 when the device clock ran 3 min 12 s slow).
- **Treat times without a zone as**: the zone for that source's local times written without an offset (IIS in local mode, a Plaso CSV in local time, EXIF dates). Times that carry Z or an offset, and file system times, are not changed. The zone the system itself was set to is shown as a suggestion with **Use suggested zone** and **Keep UTC**; it is never applied by itself.

Only the timeline uses these settings (its rows, range filter and histogram); the parsed records keep their original times. Each save is your **decision** and needs a reason: VESTIGEX keeps it with the source (who, when, zone, offset, reason, the suggested zone and whether you accepted or overrode it) and writes it to the audit log, then rebuilds that source's timeline rows. Keeping UTC with a reason is a decision too: it removes the assumed UTC badges. The Data Sources page and the report show each source's decision. An adjusted row has a ± next to its time; its pop-up shows the stored value and the adjustment, and the report lists each source's settings. The histogram counts each event in the display time zone with the offset of its own date, so daylight-saving changes do not shift bars.

Connections

E-mail addresses, phone numbers, accounts, IP addresses and web domains collected from every parsed artifact and linked when they appear in the same records. Pick an entity to centre it in a graph with the data sources it appears in and everything it shares records with. Click a linked entity or a line to see the records behind the link, with the shared values highlighted. Drag nodes to untangle the graph; **Reset layout** puts them back.

Media Analysis

Scan Photos runs offline over every photo and five frames of every video:

- **Skin / nudity** (NudeNet, double-checked by CLIP), **Weapons** (CLIP) and **AI-generated** (labels inside the file such as C2PA content credentials or generator settings, plus a UniversalFakeDetect score).
- **Picture categories**: documents and receipts, ID cards and passports, screenshots, cash and bank cards, drugs, vehicles and plates, maps and locations, children.
- **Near-duplicate pictures**: the same image resized, recompressed or lightly edited.

Each list has a **minimum score** filter and a **List** / **Gallery** view. With **Blur** on, thumbnails stay blurred until you select one; click the open photo again to blur it again.

Check with local AI shows each hit at or above the minimum score to the local vision model with one narrow question and records its answer: confirmed (AI ✓), rejected (AI ✗) or unsure (AI ?), with a one-line description. The **AI check** filter hides rejected hits by default. Nothing is deleted and the detector score is kept.

Read text & speech transcribes audio and video (Whisper, in the original language with timestamps) and reads text in pictures and scanned PDFs (OCR). Both are added to Keyword Search and the IOC sweep.

AI Investigation

Ask questions about the case in plain language. The model answers from the case data and cites what it used: findings, events, files and artifact records. Click a citation to open the exact record; citations the model could not back with evidence are marked unverified. Conversations are kept with the case. Configure the model in **Settings > AI assistant**.

With a local model (Ollama, LM Studio) everything stays on the workstation. With an online service (ChatGPT, Claude, Gemini or another) the header shows **ONLINE** and the page first asks whether this case may use it: your questions and what the model reads to answer them (findings, events, file names and paths, chat and email text, artifact records) are sent to that company; the evidence files themselves are not. Tick the box that you are authorised and press **Allow for this case**. The permission is per case and per service, **Stop allowing online AI** withdraws it, and until it is given nothing is sent, also not by **Read chats with AI** or **Draft with AI**. The decision and every use are entries in the audit log ("Case data sent to online AI", with the service, model and what it was for). Do not use an online service for material you may not share with a third party.

Artifacts

Registry, OS Artifacts and Web History

Parsed records, grouped by artifact with counts. Windows examples: run keys, services, UserAssist, ShellBags, USB devices, ShimCache, Amcache, BAM, Prefetch, LNK files, Jump Lists, Recycle Bin, scheduled tasks, PowerShell history and script blocks, SRUM, Windows Timeline, Defender, remote-access tools, encrypted and password-protected files. macOS (through mac_apt and VESTIGEX's own parsers): KnowledgeC, Biome, quarantine events, TCC permissions, FSEvents, Unified Logs, Notes, iMessage, Wi-Fi and more. Linux: users, shell history, authentication logs, logins, cron, services, syslog and the journal. Web History covers browsers (per browser) and cloud storage clients (OneDrive, Google Drive, Box Drive, Dropbox).

Each record's detail pane shows every field, the review panel, ☆ **Bookmark** and ✦ **Ask AI**.

Messaging

Chats from computers (Skype, Viber, Teams, Slack, WhatsApp Desktop, Discord) and from each phone (SMS, iMessage, WhatsApp, Telegram, Signal ...), shown as conversations with message bubbles or as a table. **Read chats with AI** gives each message its language, an English translation and topics (threats, weapons, drugs, money, sexual content, minors, meetings, self-harm, fraud, hiding evidence and your own topics); topic chips filter the conversations.

A long conversation loads more as you scroll; **Load all** loads it to the end. A message stored more than once with the same time, direction, sender, text and attachment (WhatsApp can keep one contact's chat under two ids) is shown once with "2 copies"; open it to see each stored record. **Export whole chat** saves the conversation as HTML, PDF or CSV with each message, its time in the display time zone and its record number (a message stored twice is shown once with both record numbers; tick **every stored copy** for one bubble per stored record; the CSV always has one row per record); the file is listed under Reports and its SHA-256 is written to the audit log. **RSMF** next to those buttons saves the same conversation in Relativity Short Message Format (below).

Export RSMF (top of the Messaging page) saves chats in **Relativity Short Message Format 2.0** for review in Relativity and other eDiscovery platforms that read RSMF. Pick any conversations from phones, computer apps (Skype, Viber, Teams, Slack, WhatsApp Desktop, Discord, Signal Desktop, Telegram exports) and cloud exports (Facebook Messenger, Instagram Direct, WhatsApp chat exports); the filter narrows a long list. Each conversation becomes an .rsmf file: an e-mail file whose text is the conversation (searchable as extracted text) and whose single attachment, rsmf.zip, holds rsmf_manifest.json (participants, the conversation, and every message with its time in UTC, sender, sent / received, replies, reactions, edits and deleted messages) and the message attachments.

- **Include attachments** reads each attachment from the evidence (or from the media unpacked from a cloud export) by its path. An attachment that is not in the evidence, or a bare file name that matches several different files, is not guessed: the message names it under "Attachments not included".

- Every message keeps its VESTIGEX record number and source file as custom fields, so a reviewer can trace it back to the case. Times that the source stored without a time zone are written as UTC and say so.
- Relativity recommends at most 10,000 messages per file, so longer conversations are split into numbered parts that share one event collection id. **Split long conversations** can also make one file per month or per day.
- One conversation is saved as a single .rsmf; several are saved as one .zip with the .rsmf files and an index.csv (file, conversation, app, source, messages, attachments, first and last time, SHA-256 of each file). The file is listed under Reports, and the export with the SHA-256 of the result and of every .rsmf is written to the audit log.

Email

Messages from `.eml`, `.mbox`, Outlook PST / OST and `.msg` files, with attachments.

Memory

RAM dumps and hiberfil.sys are analysed offline with Volatility 3: processes (including unlinked and exited ones), command lines, DLLs, network connections, services, injected code (Volatility's malfind rule) and registry hives. The enabled YARA rules are run over every process's memory (a match names the process and address) and over the whole image. Every memory file's text goes into Keyword Search, and its URLs, IP and e-mail addresses are listed with counts. The image's text, URLs and raw YARA scan, and the YARA scan of the processes' memory, run in several worker processes while Volatility works, so a multi-core computer is used fully.

For Windows images VESTIGEX also lists:

- **Files in memory** (windows.filescan): every file object Windows had open or cached, including files later deleted from disk. Select one and press **Extract file from memory** to save the parts Windows still held (windows.dumpfiles): the data section (the file as read), the image section (a program as loaded) and the file cache, each with its size, MD5 and SHA-256. Pages that were not in memory are zeros, so a copy can be partial. Copies are kept in the case folder under memory-files, programs with ".bin" added, and each extraction and download is in the audit log.
- **Handles (files, keys, mutexes)** (windows.handles): which files, registry keys and mutexes each process had open. **Mutexes** (windows.mutantscan) lists every named mutex; many malware families create a fixed, well-known one.
- **Process cross-view (psxview)**: each process as seen by the kernel's process list, the pool scan, the thread scan and csrss. A process that is still running but missing from the process list or the pool scan is marked hidden. Exited processes are normally missing from the thread scan and csrss.
- **Process tampering**: findings of Volatility's process hollowing, process ghosting and PEB masquerading checks, DLLs mapped into a process but missing from its module lists (ldrmodules), and threads that start in memory not backed by a file or in an unexpected region (suspicious_threads, at most five minutes on a large image).
- **Registry hives**: every hive loaded in memory. Select one and press **Browse hive** to walk it like regedit, as it was in RAM (windows.registry.printkey): click subkeys or type a path, see each key's last write time and its values (strings, numbers and lists decoded; binary data as hex with the text found in it, so RecentDocs and MRU entries show their file names). Keys marked **volatile** exist only in memory, never in the hive file on disk.

- **UserAssist, Amcache, ShimCache and Scheduled tasks (from memory)**: program-execution and persistence evidence read from the registry hives and the kernel's AppCompatCache in RAM, with their times on the timeline. They can include entries newer than the hives on disk.
- **Kernel modules** (windows.modules, modscan, unloadedmodules): the loaded kernel drivers, the ones only a scan of memory pools finds, and recently unloaded ones with their unload time. **Kernel callbacks** (windows.callbacks): every routine drivers registered to be told about new processes, threads, image loads, registry changes, shutdown and bug checks.
- **Token privileges** and **Token SIDs** (windows.privileges, windows.getsids): each process's privileges (present, enabled, enabled by default) and its user, groups and integrity level. **SeDebugPrivilege** switched on by the process itself, and a process running as **SYSTEM** whose parent runs as an ordinary user (a stolen or duplicated token, or a tool such as PsExec -s), are added to Process tampering.

Rootkit findings and **Rootkit checks run** (Windows, Linux and macOS images): VESTIGEX looks for kernel code that no loaded module explains and for objects hidden from the kernel's own lists.

	Checks
Windows	System call table (SSDT); interrupt table of every processor (read by VESTIGEX from each processor's KPCR, as Volatility 3 has no Windows IDT plugin; device interrupts are followed to their driver's service routine); kernel callbacks; driver IRP handlers; drivers with no module behind them (drivermodule); kernel modules found only by the pool scan; system threads and kernel timers outside every module; ETW functions patched in processes; ntdll system call stubs that differ between processes; threads with hardware breakpoints; the Skeleton Key patch in lsass.exe
Linux	System call table; interrupt table; network protocol handlers (afinfo); terminal input handlers; keyboard notifiers; netfilter, ftrace and tracepoint hooks; kernel threads; modules missing from the module list but in sysfs or a memory scan (modxview); processes sharing one credential structure; processes named like kernel threads, running a deleted program, or renamed in a temporary or hidden folder; processes a memory scan finds that the task list does not hold. eBPF programs are listed under **eBPF programs**
macOS	System call, Mach trap and sysctl tables; kauth listeners and scopes; socket filters; TrustedBSD policy hooks; kernel timers

A finding is either strong (a handler outside the kernel and every module, a hidden driver, module or process) or a **Lead**: something rootkits do but legitimate software does too, such as a system call hooked by a named driver on 32-bit Windows (security products did this before 64-bit Windows forbade it), a kernel timer left by an unloaded driver, or a keyboard notifier in a module. Strong findings raise threat hunt rule ART-028. **Rootkit checks run** has one row per check: what it looked at, how many entries, how many findings, or that it could not run on this image (the reason is in the analysis log), so "nothing found" is never confused with "not checked". Memory sources processed before these checks existed need processing again.

Windows console commands lists recovered console history separately from process launch arguments. Reprocess an existing RAM source with memory analysis enabled to populate it. Clear date filters: command history does not supply execution timestamps. Select a command to inspect its console host PID, application, history slot, history address and command address. `ProcessHandle` is a handle in the

host, not the command's PID; `AssociatedPID` stays empty when unresolved. Repeated commands at different slots are retained. Legacy slots beyond the recorded command count are marked `residual`; they may be stale.

Search an individual process's memory: select a Windows process in Memory → Processes and press **Search memory** in the record's header (next to Bookmark and Ask AI), or choose **Search memory** beside it in Process Inspector. The search panel opens over the page; enter 2–512 characters, such as `hacker` or `net user`. Matching is literal and case-sensitive, in ASCII and UTF-16LE. The search reconstructs that process's virtual memory from its RAM source; a reused PID is checked against the recorded creation time, and ambiguous matches are refused. Windows RAM sources and previously decompressed Windows hibernation sources are supported; Linux process search is not yet supported.

The search enumerates Volatility's process translation mappings independently of VADs. It joins adjacent readable virtual bytes even across mapping boundaries, but never fills missing pages with zeros or joins bytes across gaps. By default only the process's own (user-mode) memory is searched. Kernel memory is mapped into every process, so the same kernel bytes appear in every process's address space; tick **Include kernel memory** to search it as well. Each match is labelled **process memory** or **kernel memory (shared by all processes)**, and the result counts each kind: a kernel match does not show that this process held the text. Each match shows its virtual address, encoding, observed contiguous span, and surrounding text and hex bytes. These are virtual addresses, not raw-image file offsets. A retained request or command fragment does not prove execution.

URL-encoded context is automatically decoded into labelled derived views alongside the original evidence. Percent decoding preserves literal `+`; HTTP form decoding interprets original `+` as a space. Nested percent encoding is decoded up to three passes, and invalid UTF-8 is flagged. Searching still matches original bytes: search for `hacker` to recover context such as `cmd=net+user+hacker`, or search for the encoded spelling itself. Decoded views do not change match addresses or original hex bytes.

Matches are listed in a table (address, encoding, process or kernel memory, and a short preview); sort it by any column, filter it to process or kernel matches, and click a match (or step with ↑/↓) to see its full context, decoded views and hex on the right. The ⓘ next to the result summary explains what was searched. Closing the panel (✕ or Esc) does not stop a search: it keeps running in Jobs, and pressing Search memory again, or **Open results** on the search in Jobs, returns to the same search and match. Earlier searches of the process are under Recent searches.

The limits row sets matches (1–5,000), search size (1–16,384 MiB), scanning time (1–900 seconds), and context bytes per side (0–4,096). Defaults are 500 matches, 4 GiB, five minutes and 64 bytes per side. Context reads are additional to the search-byte budget; the overall worker limit is fifteen minutes including symbol discovery. Coverage separates failed reads from mapping gaps (which may be unallocated or nonresident) and known mapped bytes skipped. Unenumerated mappings remain unknown. Complete means the readable mappings were searched, not that all process memory was resident. No match in a partial search does not establish absence. Each search (examiner, text, process, source, limits, whether kernel memory was included, and the outcome with its match counts) is written to the case's audit log.

Results are saved under the case's `process-search` folder: reopen them through **Recent searches** for the same process or export JSON with the query, source, source evidence hash when available, process identity, search time, coverage, decoded views and match bytes. Run a new search to obtain these fields

for older saved results. Leaving the panel does not cancel a running job; use **Cancel** or Jobs. No execution timestamp is inferred from a search hit.

Console history recovery reports coverage, errors and partial scans. Legacy decoding covers NT 5.1/5.2/6.0/6.1 on x86 and x64 (including Vista / Server 2008), scanning `csrss.exe` before Windows 7 and `conhost.exe` on Windows 7. It searches the default history size of 50 and sizes recovered from the registry, with a 60-second budget per host and 300 seconds overall. Other custom sizes may be missed. Windows 10-family x64 images use the bundled Volatility `cmdscan` plugin and its console symbols; unsupported layouts are reported. These paths have synthetic regression coverage; recovery from a particular acquisition still depends on its surviving memory. History alone does not identify the actor or prove a command succeeded. This feature does not reconstruct screen output or commands submitted directly through a web application.

Linux memory images (raw, LiME, VMware .vmem, QEMU / KVM ELF dumps) are recognised by their kernel banner and analysed with Volatility's Linux plugins: processes and command lines, bash history, sockets and network interfaces, kernel modules (including hidden ones), injected code, hooked system calls, environment variables, open files, mounts, the kernel log, YARA over each process's memory, and the rootkit checks above. **macOS** memory images (Intel Macs running macOS 10.x) are recognised by their Darwin banner and analysed the same way: processes and command lines, bash history, network connections and interfaces, kernel extensions, open files, mounts, the kernel log, injected code and the macOS rootkit checks. Volatility 3 cannot analyse Apple silicon memory, and the symbol collection only keeps symbols for macOS 10.x. Volatility needs the symbol file of the exact kernel: install the **Volatility symbols (Linux & macOS) index** under Settings > Components, and VESTIGEX takes the matching file from its cache, from the **symbol folders** you add there (a copy of the Abyss-W4tcher volatility3-symbols collection, or your own files), or downloads just that file once (switch this off for offline work). Without a symbol file the image's text, URLs and raw YARA are still searched, and Memory image says what is missing.

Analysis log on a memory source (Data Sources) shows what was run: each Volatility step with the `vol` command that does the same, its time, the rows it gave, and any error with its traceback and Volatility's warnings. Network connections are not available for Windows XP / Server 2003 images (Volatility 3 has no plugin for them).

Network

Packet captures (.pcap and .pcapng, and the packets in Windows .etl traces made with pktmon or "netsh trace start capture=yes"; read in one pass however large, without Wireshark) give these tabs:

Tab	What it holds
Conversations	Every TCP, UDP and ICMP conversation: client and server with ports, start and end, duration, packets and bytes each way, whether the TCP handshake completed, and the names the server was looked up by
DNS	Each query with its type, answers, response code, client and server
HTTP	Requests and responses: method, host, URL, user agent, referer, status, content type and sizes
TLS	Server name (SNI), negotiated version, JA3, JA3S, JA4 and JA4S fingerprints, ALPN, also after STARTTLS (SMTP, IMAP, POP3, FTP, LDAP, XMPP), and the server certificate's subject, issuer, validity and SHA-1 where it is sent in the clear (TLS 1.2 and older); self-signed and expired certificates are marked. Every certificate of the chain is saved once as a .cer file under Files transferred
Files transferred	Files carried over cleartext HTTP (downloads and uploads), FTP, TFTP, SMB2, email (each message as .eml and its attachments, from SMTP, POP3 and IMAP), LPR print jobs, and TLS certificates, rebuilt from the packets with MD5 / SHA-256 and whether any part was missing
Hosts	Per IP address: MAC addresses (ARP, DHCP), host names with where each came from (DHCP, NetBIOS name registrations, Windows browser announcements, the Kerberos machine account, the NTLM workstation name), workgroup or domain, Windows version, an operating system guess with its evidence (TCP SYN fingerprint, DHCP vendor class, browser announcements, HTTP user agents), the user accounts that signed in from it, DNS names, traffic, and IP addresses claimed by more than one MAC
Email messages	Messages sent with SMTP and read with POP3 or IMAP (unencrypted, or before STARTTLS): from, to, cc, subject, date, the text and the attachment names; the .eml and attachments are under Files transferred
Chat and text messages	IRC (PRIVMSG / NOTICE with the sender, channel or user and text) and SIP MESSAGE texts
VoIP calls and audio	SIP calls (caller, callee, start, answer, end, duration, phones) with a recording you can play in VESTIGEX (caller on the left channel, callee on the right) and each direction as its own WAV; RTP audio streams without SIP are found and played too. G.711 (PCMU / PCMA) is decoded; other codecs (G.722, G.729, Opus …) are kept as raw files with the codec named
Accounts	Account names seen in Kerberos (AS-REQ, TGS-REQ, replies and errors), NTLM sign-ins (SMB, HTTP, LDAP, RPC …) and LDAP binds: account, domain, the person's full name when the domain controller sent it (SAMR user information over SMB), user / machine / service, the computer it came from, servers and services, first and last seen, and Kerberos errors such as unknown user or wrong password. Only names are read: tickets, pre-authentication data, NTLM responses and passwords are never kept
Remote access and lateral movement	SMB, RDP, SSH, WinRM, RPC, Telnet, VNC, TeamViewer and AnyDesk sessions with both ends; SMB shares, files and named pipes opened (service control, task scheduler, remote registry …), SSH banners and the RDP user-name cookie
Scans and sweeps	Port scans, port sweeps across hosts and ping sweeps, with how many probes were answered
Beaconing candidates	Connections repeated to the same server at a steady interval, with the interval, jitter and a score out of 100

Tab	What it holds
Cleartext logins and protocols	FTP, Telnet, POP3, IMAP, SMTP AUTH, LDAP simple bind and HTTP Basic logins sent without encryption. That a login happened and where is recorded; user names and passwords are not extracted or stored

Tunnelled and encapsulated traffic is taken apart and the packets inside are analysed like any other: 802.1Q VLAN and QinQ, MPLS and Ethernet over MPLS, PPPoE, GRE (with Ethernet over GRE and ERSPAN), VXLAN, IP-in-IP, IPv6 in IPv4 (6in4), Teredo, OpenFlow packet-in and packet-out, and SOCKS 4 / 4a / 5 proxies (the Conversations record names the proxied destination, and HTTP through the proxy is read as HTTP). IPv6 is handled throughout, with IPv6 addresses matched to MAC addresses by neighbour discovery.

The rebuilt files are added as their own data source ("… (network files)"), so they are hashed, matched against hash sets, typed and scanned with YARA, and appear under File System. Network records go into the Timeline as **Network**, into Keyword Search, the IOC sweep (IP addresses, domains, URLs, hashes and JA3), Connections, and Lateral Movement (as "Network captures" sessions). Encrypted traffic (HTTPS, SMB 3 encryption, SSH, RDP) shows who talked to whom and when, but not what was sent.

> **Programs are stored safely.** Files VESTIGEX writes into the case folder that Windows would run when double-clicked (programs, scripts, installers, shortcuts, macro-enabled Office files, disk images such as .iso) are stored with **.bin** added, for example `runner_ilove.exe.bin`, so they cannot be started by accident from the case folder. VESTIGEX itself shows and reports the original name (`runner_ilove.exe`) and the same hashes. Files downloaded from VESTIGEX (Export, email attachments) get .bin added the same way; remove it only if you mean to run the file, and only in an isolated analysis machine.

Cloud: Facebook, Instagram and WhatsApp

Evidence you already have from these services, added under Data Sources (Add evidence → Cloud):

- **Facebook "Download Your Information" exports** (JSON or HTML): every file of the export becomes records on **Cloud** → **Facebook**, one kind per file, such as profile information, logins and logouts, account activity and IP addresses, recognised devices, posts, comments, likes and reactions, friends and friend requests, groups, searches, Marketplace, payments and locations. Messenger conversations are on **Messaging** (app "Facebook Messenger", with sent / received, reactions and attachments), and photos, videos and other files are unpacked into the case and indexed as their own data source ("… (media)"), so they are hashed and appear under File System and Media Analysis. Facebook's garbled accented text ("Ã±") is shown correctly ("ñ").
- **Instagram "Download Your Information" exports** (JSON or HTML): the same, on **Cloud** → **Instagram** (profile and username, login activity with IP addresses and devices, followers and following, posts, stories, comments, likes, searches and the rest). Direct messages are on Messaging (app "Instagram Direct").
- **HTML-format exports** (Facebook and Instagram): each entry of each page becomes a record with its fields. The times on HTML pages are the local time of whoever downloaded the export and say no time zone, so they are kept as written; the Timeline marks them "assumed UTC" until you set the source's time zone under Data Sources → Times. JSON exports carry UTC times and need nothing.

- **WhatsApp chat exports** ("Export chat" on the phone, with or without media): add the .zip, the .txt (WhatsApp Chat with <name>.txt, or _chat.txt from an iPhone) or a folder holding many of them. Every message is on Messaging (app "WhatsApp (chat export)") with its sender, attachments, deleted and edited messages and media left out of the export; each chat is one entry on **Cloud** → **WhatsApp** with its participants, number of messages, first and last message. The export's photos, videos and voice notes are unpacked and indexed like the Facebook media. A chat export does not say whose phone it came from, so messages are not marked sent or received. Its times are the phone's local time with no zone (set it under Data Sources → Times), and the date order (day / month or month / day) is worked out from the file: a day above 12, then the order that keeps the messages in sequence, then the dates in attachment names; when nothing decides it, day / month is assumed and each message says so ("date_order").
- **WhatsApp account information reports** ("Request account info"): each section of the report's pages (phone number, profile, devices, groups, contacts, settings) becomes records on Cloud → WhatsApp.
- **Meta warrant returns** (Meta Platforms Business Record): each section of records.html (request parameters, registration date and IP, IP addresses with their times, emails, phone numbers, names, linked accounts, locations, payment methods and any others in the return) becomes records on Cloud → Facebook, one per entry; files under linked_media are indexed like the export's media.

Everything with a date goes into the Timeline, and Keyword Search finds text in these records (the Network & cloud tab). Nothing is downloaded from Facebook, Instagram or WhatsApp: VESTIGEX reads what you add.

Mobile

One tab per phone with its device details (name, model, OS, number, IMEI, serial), then the artifacts in ALEAPP / iLEAPP's own categories: accounts, apps, locations on a map, Wi-Fi, browser, photos metadata and more. Cellebrite UFDR imports appear the same way.

SQLite Databases

Every SQLite database in the evidence, found by signature, grouped by the app it belongs to. Browse tables with sorting and search. ♻ **Recovered records** lists deleted rows recovered from free pages, free space inside pages and the write-ahead log.

Credentials

Local account password hashes (SAM), LSA secrets and saved browser passwords. Values stay masked until revealed. Local hashes export in pwdump format for your own offline tools, and the export's SHA-256 is written to the audit log.

Threat hunt

- **Findings**: detection rules run against the log events and against the artifacts parsed from the evidence, with MITRE ATT&CK techniques and the matching events. The top of the page sums up what was found: how many findings of each severity, which hosts have critical or high ones, and the time span. Three tabs: - **Incidents** (the default): one card per host, the most serious host first, holding that host's findings of medium severity or worse as bursts of activity in time order (a quiet gap of more than two hours starts a new burst). Findings that come back like a schedule (the same findings at about the same time of day on 3 or more days, or on 7 or more days at any time) are one **Recurring** entry showing the days, the usual hours and the events per day, so a routine job does not fill the list; check one day and mark it Benign or suppress it if it is harmless. Open an entry for its findings in the order they first fired, the accounts and addresses involved, **View events** for each step and **Ask AI**. - **All findings**: every finding, filtered by severity and by your verdict. Open the evidence viewer to page through every event. Under each finding, **Examiner verdict and notes** records Confirmed, Benign or Needs review with a note; findings marked Benign leave the incidents and are counted separately. **Suppress…** stops the rule firing for a host, account, address or other value you have checked (for example a backup server or an administrator's tool): the matching events are removed at once and on every later threat hunt. - **Suppressions**: every suppression with who added it and why; **Remove** undoes one (run the threat hunt again to bring the events back). Verdicts and suppressions are recorded in the audit log.

Rules over parsed artifacts (ART-…) need no logs: they look at execution traces (Prefetch, Amcache, ShimCache, BAM, UserAssist, SRUM, Jump Lists, shortcuts), Run keys, services, scheduled tasks, Defender quarantine and exclusions, reassembled PowerShell script blocks and history, browser downloads and visits, memory (injected code, script hosts talking to the internet, kernel rootkit findings, token stealing), local accounts with blank passwords, and Linux / macOS shell history, SSH logins and cron. The threat hunt checks a data source's artifacts again whenever they change.

- **Lateral Movement**: logons and remote connections between hosts drawn as a graph, notable connections first (failed logons, RDP, explicit credentials, admin shares, service installs). Click a line to see its accounts, logon types and time span. The panel beside the graph is its key and its filters: lines are coloured by type (remote execution, RDP, explicit / cleartext credentials, network logon, other) and drawn solid (successful logon recorded), dashed (failed attempts only) or dotted (outcome not recorded); click a type or outcome to hide or show it, and use the switches to hide computer-account logons or public sources, or keep only hosts in threat hunt findings. A node whose operating system VESTIGEX knows shows a Windows, Linux or macOS mark: from the system information of your evidence (matched by computer name or IP address, with the computer name shown under the IP) or because its own Windows event logs were read. A computer's short name and full domain name count as one host. An IP address counts as its computer when the system information of the evidence lists it, the RDP client resolved the computer's name to it, or its logons all gave the workstation name of a computer whose logs are in the case; other workstation names are only noted, since the client chooses them. **Outbound RDP** (1024, from the RDP client log) draws connections a computer made to other servers, with the account (matched from the hash the log keeps) and whether it connected. **Timeline** lists every logon (500 at a time, with a filter and **Load more**), and **Chains** shows each multi-hop path with a table of its hops. **Export HTML** saves

the connections the Network Graph shows (its mode, Max and filters) as one self-contained HTML file in the case's reports folder (it opens offline in any browser): a case header, the movement graph (click a host, arrow or row to isolate its connections), a numbered connection register with accounts, times in the display time zone, the event IDs behind each arrow and whether a successful logon was recorded, and the findings as cards. Hosts are shaded by what the analysed events show (whose logs are in the case, successful logons, flagged by a high or critical finding, public addresses); that shading is a lead, not a conclusion. The file's SHA-256 goes into the audit log.

- **Process Inspector**: processes as a tree, from process-creation events (Sysmon 1, Security 4688) and memory images (with full command lines; processes found by the memory scan but missing from the process list are marked), plus Prefetch runs and BAM entries, which show what ran on hosts without process logging but not what started it. **Source** switches between them. Chains such as Office starting PowerShell are flagged by the inspector's own rules, and processes covered by a threat hunt finding carry it (**Threat hunt** filter). An encoded PowerShell command is shown decoded under the command line.
- **Findings across the case**: hosts, accounts, IP addresses and data sources that appear in threat hunt findings (medium or worse, not marked Benign) get a dashed ring or ⚑ in the colour of the worst finding in **Lateral Movement** and **Connections**; hover for the findings. The **Timeline** marks the events and records behind findings (**Detections only**). Reports show each finding's verdict and notes, and can leave out findings marked Benign or include only Confirmed ones.
- **Detection Rules**: the built-in rules (68) the threat hunt uses (Windows event logs, web server logs, registry exports, parsed artifacts), with an editor to add or change your own and Sigma import. Each built-in rule is tied to the log that produces its event (an event ID means nothing without its log), checks the fields that matter, leaves out known harmless cases, and uses a threshold where one event proves nothing (for example 10 failed logons for one account from one source within 10 minutes). Web rules read the request URL-decoded, so encoded SQL injection and similar attacks are recognised, and claims that a file was served ignore a site's catch-all page. Expand a rule to see its conditions. Built-in rules are updated with VESTIGEX unless you edited them; switching a rule off is kept. Search and the **Built-in / Sigma / Your own** filters narrow a long list. - **Import Sigma** takes .yml files or the whole SigmaHQ rule pack (sigma_all_rules.zip or sigma_core.zip from the Releases page of github.com/SigmaHQ/sigma). Choose the statuses (stable, test, experimental) and the minimum level to keep. VESTIGEX evaluates each rule's own selections, modifiers and condition on Windows event logs (process_creation reads both Sysmon 1 and Security 4688) and web server logs; rules for other logs (Linux, macOS, cloud) or with aggregations are skipped and counted by reason. Importing again updates the rules in place. **Remove Sigma rules** takes them all out. Run the threat hunt again to apply new rules to a case.
- **ATT&CK & IOCs** has three tabs: - **ATT&CK matrix**: findings laid out by ATT&CK tactic and technique, with the IOC sweep results. **Possible threat actors** ranks the ATT&CK groups reported to use the techniques and tools seen in the findings: rare techniques and named tools (Mimikatz, PsExec, Rclone ...) count most, and each group shows what matches, what it is known for that the case could have shown but did not, and a link to its ATT&CK page. It is a lead for the investigation, not attribution: many groups share techniques and tools, and ATT&CK only knows what has been publicly reported. With few techniques seen the list cannot tell groups apart, and the page says so. - **Coverage**: every technique the enabled rules look for, marked Seen (a finding), Could be seen (the case holds what the rules read; nothing was found) or Cannot be seen (the events or artifacts they need are missing). Click a technique for its rules and what each reads. **What is missing** lists the absent logs, those that would open up the most techniques first, and the auditing that produces them. "Cannot be seen" says nothing about whether the technique happened. - **IOC Hunting**: IPs,

domains, URLs, e-mail addresses and file hashes typed or imported from text, CSV, STIX 2 or MISP. **Sweep** looks for every IOC across file hashes and names, artifacts, chats, memory, file contents and log events.

Review and reporting

Bookmarks and review status

Bookmark files, artifact records, chat messages, phone records and log events with ☆ **Bookmark**, and add a note to each. **Reporting > Bookmarks** lists them all. Any record or file can also be marked **Relevant**, **Follow up** or **Not relevant** with comments for the team; each change records who and when.

Refined Results

For the case type, a set of checks pulls together the evidence that usually matters in that kind of case (for example USB devices and cloud uploads for insider data theft). Results follow the date range and source filter; **Show all** loads every item and ↻ **Refresh** recomputes.

Reports

- **Case report** as HTML, PDF (printed by Microsoft Edge so it looks the same), Word or Excel. Choose the findings and sections to include; the report follows the date range, source filter and display time zone.
- **Draft with AI** writes an executive summary from the report's data only. You edit it, and the report states that AI assisted unless you mark it as rewritten.
- **Portable case**: a zip for a lawyer, investigator or supervisor without VESTIGEX. It opens in any browser with no internet: case summary, evidence hashes, findings, bookmarked and reviewed items with copies of the files. The reviewer marks items and comments, then sends back a small review file that you import with **Import review**.

Every report is saved in the case's reports folder and its SHA-256 is written to the audit log.

Audit Log and chain of custody

Reporting > Audit Log is the case's hash-chained activity log: every examiner action with time (UTC), examiner and details (case opened and closed, sources added, jobs and their results, bookmarks, exports, reports). Each entry stores the hash of the previous one, so editing or deleting an entry is detected. Filter by text, action, examiner and date, and export as CSV.

Settings

- **Backend**: the address of the VESTIGEX backend (normally left as it is).
- **Appearance**: themes (Graphite, Dim, Light, Blue, Green, Dark Coffee, Coffee, Pink or Match Windows).
- **Users**: examiner accounts (administrators only).
- **Organization logo**: replace the VESTIGEX mark with your organization's PNG, JPEG, WebP or GIF logo (up to 2 MB).
- **Processing**: how many data sources are processed at the same time, and how many files within a source are hashed and have their text extracted at the same time (default 8).
- **Time**: the display time zone for every timestamp (UTC by default).
- **AI assistant**: the model server, the model and its options. **On this PC or network**: **Ollama**, or LM Studio, llama.cpp or another OpenAI-compatible local server, with the context window and reasoning. **Online**: **ChatGPT (OpenAI)**, **Claude (Anthropic)**, **Gemini (Google)** or another OpenAI-compatible online service, with an API key (see Using an online AI service). Each server keeps its own settings. **Test connection** checks the saved one. Picture checks always use a local Ollama model that can see images (for example Gemma 3 or 4, Qwen-VL or LLaVA).
- **Updates**: new versions of VESTIGEX itself: check for updates, check the program files and repair them, and whether VESTIGEX looks for a newer version when it starts (see Upgrading).
- **Components**: the data, rules, tools and models VESTIGEX works with: MITRE ATT&CK, SigmaHQ rules, Volatility symbols, GeoNames places, ALEAPP and iLEAPP, the CLIP, UnivFD and Whisper models, and your Ollama models. Each shows the version installed and its source. - **Check** asks the official source whether a newer version exists. - **Update** downloads it (see Jobs for progress), checks it and puts it in place. - **Import…** does the same with a file you bring in, for a lab without internet (for example enterprise-attack.json, sigma_all_rules.zip, windows.zip, cities1000.zip or a zip of a model folder). - **Roll back** returns to the version used before. - Ollama models can be updated or downloaded by name (for example gemma3:12b). - The picture and speech models only change when you import one, so results stay reproducible. - Only administrators can change components. Every change is kept in the update history and written to the open case's audit log, and reports list the component versions in use.

Setting up the local AI

1. Install Ollama (offered by setup, or from ollama.com).
2. Download a model, for example in a Command Prompt: `ollama pull gemma3:12b`. Choose a size that fits your graphics card's memory.
3. In **Settings > AI assistant**, choose Ollama, pick the model and press **Test connection**.

Using an online AI service

For cases whose data may be shared with a third party, a large online model answers faster and more thoroughly than one that fits a workstation.

1. Create an API key in your account with the service: OpenAI (platform.openai.com), Anthropic (platform.claude.com) or Google (aistudio.google.com). The service bills that account for what VESTIGEX sends; VESTIGEX adds nothing.

2. In **Settings > AI assistant**, choose the service under **Online**, paste the key and press **Save**. The list of models then loads; pick one and save again. For **Other online service**, also enter its OpenAI-compatible address (for example `https://openrouter.ai/api/v1`).
3. Open the case, go to **AI Investigation** and allow the service for that case (see AI Investigation).

The key is kept in VESTIGEX's settings on this workstation, encrypted for your Windows account; it is never written to a case, a report or an installer, and another Windows account has to enter its own. To go back, choose Ollama again: the local model and the online key are both remembered. What the service does with the data it receives (how long it keeps it, whether staff may see it) is set by its terms and your account, so check them against your own rules before using it on a real case.

Troubleshooting

Problem	What to do
Starting VESTIGEX again does nothing new	Only one VESTIGEX can run; starting it again brings the open window to the front. Setup refuses to run while VESTIGEX is open.
Setup says the package is damaged	Its SHA-256 does not match SHA256SUMS.txt: download it again.
A download failed during online setup	Press **Retry** in the setup window, or later **Settings > Components > Setup downloads**.
VESTIGEX does not start	Open the **VESTIGEX logs** shortcut and check `backend.log`.
Photo scanning or OCR is slow	Without an NVIDIA GPU these run on the processor. Close games or other GPU-heavy programs.
AI features say the model is not reachable	Start Ollama, then **Settings > AI assistant > Test connection**.
An online AI service says it did not accept the API key, or is out of credit	Check the key and the billing of your account with that service, then **Settings > AI assistant > Save**.
"... is an online service and has not been allowed for this case"	Open **AI Investigation** and allow it for this case, or choose a local model in Settings.
"This image has an encrypted volume that is still locked"	Enter the BitLocker recovery key or password on the data source.
Processing on a slow external or hard disk	Copying the evidence to an SSD first makes processing much faster.

Where things are kept

What	Where
Program	`%LOCALAPPDATA%\Programs\VESTIGEX`
Accounts, settings, hash sets, YARA rules	`%LOCALAPPDATA%\Programs\VESTIGEX\backend\data`
Logs	`%LOCALAPPDATA%\Programs\VESTIGEX\logs`
Linux: program, data and logs	`~/.local/share/vestigex`, `~/.local/share/vestigex/backend/data`, `~/.local/share/vestigex/logs`
A case	the folder you chose: `case.vestigex`, `vestigex.db`, reports and caches