DeathBox – ChatGPT Security Audit

This document contains ChatGPT's verbatim output for a "light security audit" of both the practical and technical aspects of DeathBox and deathboxes.co.uk.

I've already addressed many of the recommendations it makes on the website, along with warnings and safeguards, yet it's still a worthwhile read if you wish to dig a little deeper into the potential vulnerabilities and their vectors.

Essentially, DeathBoxes as an application is technically and practically sound. The primary weakpoints are all external: passphrase hygeine, user error or carelessness, phishing attacks and malware on the user's computer or device (e.g. a keylogger).

**1. Data Handling and Storage**

**Client‑side only:** All data is entered in the browser. Nothing is sent to the server except for downloading the web app itself. This means there’s no cloud storage or server database to be compromised.

**Export options:** Users can export either encrypted JSON or plain text. This gives flexibility but also introduces risk if the exported files are stored insecurely on a shared or cloud drive.

**Implication:** The threat surface is mostly local — malware, lost devices, or careless file handling are the primary risks.

**2. Encryption**

**AES‑256 encryption:** The site uses strong symmetric encryption to secure user data client‑side.

**Passphrase‑based:** Security depends entirely on the strength and secrecy of the passphrase. Short or guessable passphrases make the encrypted file vulnerable.

**No server-side key recovery:** If the passphrase is lost, there is no backdoor — the data is unrecoverable.

**Implication:** Users must generate a strong, unique passphrase and keep it secure. Consider using a password manager or sharing with a trusted executor.

**3. Security Model Assumptions**

Assumes users will store exported files safely and not share them insecurely.

Assumes users will remember or entrust the passphrase.

Assumes the local machine is free from malware — if a keylogger or RAT is present, encryption does not protect the data.

**Implication:** The security guarantees are strong in ideal conditions, but user behaviour and endpoint security are the weak points.

**4. Potential Vulnerabilities**

**Local device compromise:** Malware, keyloggers, or unsecured devices could leak both the passphrase and data.

**Phishing / social engineering:** Users could be tricked into giving the file or passphrase to malicious actors.

**Data integrity:** If the exported file is accidentally modified or corrupted, it may become unrecoverable.

**Browser compatibility:** Some older browsers might handle crypto APIs differently; users could face errors or weaker encryption.

**5. Recommendations**

**Passphrase hygiene:** Encourage users to choose long, random passphrases, ideally generated with a password manager.

**Redundant backups:** Users should store encrypted files in multiple secure locations and avoid public cloud storage without additional encryption.

**User guidance:** Explicit warnings about losing passphrases, malware risks, and the need to combine DeathBox with legally recognised documents would reduce accidental misuse.

**Periodic testing:** Encourage users to decrypt their exported files periodically to confirm integrity.

**Summary**

DeathBox’s design is fundamentally strong: client-side encryption, no server storage, and offline operation give it a robust privacy posture. The main risks are human error (weak passphrase, lost file) and compromised local devices. It is suitable for its intended use, but should always be complemented with legal and physical safeguards.

**DeathBox Risk Map**

**1. Data Entry (in-browser)**

**Vulnerabilities:**

Keyloggers / malware on device

Browser compatibility issues

Accidental form abandonment

**User responsibility:**

Use a secure, up-to-date browser

Ensure device is free of malware

Save progress frequently

**2. Passphrase Creation**

**Vulnerabilities:**

Weak or guessable passphrase

Reusing passwords from other accounts

**User responsibility:**

Use a long, random passphrase

Consider a password manager

Do not share passphrase insecurely

**3. File Export / Storage**

**Vulnerabilities:**

Storing on shared or public drives

Accidental deletion or corruption

Unencrypted backups on cloud

**User responsibility:**

Keep multiple secure backups

Avoid storing unencrypted files online

Test decryption periodically

**4. File Sharing / Access by Others**

**Vulnerabilities:**

Sending passphrase via email or insecure channels

Untrusted recipients

**User responsibility:**

Share only with trusted individuals

Use secure communication for passphrase

Limit distribution

**5. Recovery / Decryption**

**Vulnerabilities:**

Forgotten passphrase → data unrecoverable

Corrupted or modified file

**User responsibility:**

Entrust passphrase to a responsible executor

Periodically test recovery

Keep backup copies

**Key Takeaway**

DeathBox itself provides strong client-side security, but the user is the critical link: device security, passphrase management, and backup discipline determine overall safety. Combining DeathBox with physical or legal safeguards is essential.