Veil – Linux Filesystem Sandboxing Tool

Project Overview

Veil is a Linux command-line tool that allows users to run applications inside a temporary filesystem sandbox. Only explicitly allowed directories are visible to the application. Once Veil exits, the system is left completely unchanged.

Problem Statement

On Linux, most desktop applications have unrestricted access to the user home directory. Running untrusted binaries, installers, or scripts requires blind trust.

Veil provides per-run filesystem isolation without requiring system-wide policies, repackaging, or background services.

Core Design Principle

If a directory is not visible to the application, it is not accessible. Security is enforced through filesystem visibility, not runtime blocking.

How Veil Works

1. User launches an application using Veil.

2. Veil creates a temporary sandbox directory.

3. A FUSE filesystem is mounted at this location.

4. System directories are exposed as read-only.

5. One user-selected directory is exposed as read-write.

6. The application runs inside this sandbox.

7. Any access outside allowed paths is denied and logged.

8. When the app exits, the sandbox is unmounted and removed.

Filesystem View Inside Sandbox

Allowed read-only directories: /usr, /lib, /bin, /etc

Allowed read-write directories: /tmp, user selected folder

All other paths are invisible to the application.

Denied Access Handling

When an application attempts to access a restricted path, the operation fails safely. Veil logs the denied access and provides a summary after the app exits.

Post Run Suggestions

Veil explains which paths were denied and suggests which directories to allow if the user wants to retry.

Optional Interactive Mode

Veil includes an experimental interactive permission mode for CLI tools only. This mode allows users to approve filesystem access requests in real time.

Interactive mode is disabled by default and is not supported for GUI applications.

What Veil Does Not Do

Veil does not provide antivirus scanning, malware detection, or system-wide protection. It does not modify file permissions or install permanent policies.

Use Cases

Running untrusted binaries

Testing installers safely

Developer workspace isolation

Preventing accidental file corruption

Why This Fits FOSS Hack

Veil is fully open source, uses core Linux primitives correctly, avoids proprietary dependencies, and demonstrates strong systems engineering.

Future Work

Graphical interface

Policy presets

Improved logging visualization

Extended interactive permissions for CLI tools

Project Timeline

Preparation Phase: Learning and prototyping before March

Development Phase: Core implementation during March

Final Phase: Documentation, demo, and cleanup

End of Report