

Implementation Report

Implementation Report: Safe `rm` and `restore` System

This report provides detailed instructions on how to **install**, **set up**, and **use** the safe deletion system implemented in this project. The system replaces the default `rm` command with a safer alternative and introduces a `restore` command to recover deleted files.

11

Project Components

- `rm` – Safe deletion script that moves files to `/tmp/trash/` instead of permanently deleting them.
- `restore` – Script that recovers files from trash and places them back to their original location.
- `/tmp/trash/` – Trash directory where deleted files are stored temporarily.
- `/tmp/trash/trash.log` – Log file in CSV format that tracks all deletion events.

Installation and Setup Instructions

1. Prepare the Trash System

Open a terminal and run:

```
mkdir -p /tmp/trash
touch /tmp/trash/trash.log
```

This creates the trash directory and an empty log file.

```
(user1kali@vbox)-[~]
$ mkdir -p /tmp/trash

(user1kali@vbox)-[~]
$ touch /tmp/trash/trash.log

(user1kali@vbox)-[~]
$ tree /tmp/trash
/tmp/trash
├── trash.log

1 directory, 1 file

(user1kali@vbox)-[~]
$
```

2. 📁 Move Scripts into a Personal Bin Directory

Create a `bin` directory in your home folder (if not already there):

```
mkdir -p ~/bin
```

Move the `rm` and `restore` scripts into `~/bin/`:

```
mv rm restore ~/bin/
```

Make both scripts executable:

```
chmod +x ~/bin/rm ~/bin/restore
```

```
(user1kali@vbox)-[~]  
$ mkdir -p ~/bin  
  
(user1kali@vbox)-[~]  
$ cd ~/bin  
  
(user1kali@vbox)-[~/bin]  
$ vim rm.sh  
  
(user1kali@vbox)-[~/bin]  
$ vim restore.sh  
  
(user1kali@vbox)-[~/bin]  
$ chmod +x rm.sh restore.sh  
  
(user1kali@vbox)-[~/bin]
```

3. 🛤️ Update Your PATH

To override the default `rm` command with your script, add `~/bin/` to the **beginning** of your `PATH`.

Edit your `.bashrc`, `.bash_profile`, or `.zshrc` (depending on your shell):

```
(user1kali@vbox)-[~/bin]  
$ echo $SHELL  
/usr/bin/zsh  
  
(user1kali@vbox)-[~/bin]  
$ vim ~/.zshrc
```

```
export PATH="$HOME/bin:$PATH"
```

```
File Edit View Bookmarks Plugins Settings Help  
# ~/.zshrc file for zsh interactive shells.  
# see /usr/share/doc/zsh/examples/zshrc for examples  
export PATH="$HOME/bin:$PATH"
```

Then apply the change:

```
source ~/.zshrc # or source ~/.bashrc
```

```
File Edit View Bookmarks Plugins Settings Help
(user1kali@vbox)-[~]
$ source ~/.zshrc
(user1kali@vbox)-[~]
```

✓ Now, when you type `rm`, your safe script will be used instead of the system one.
