Forensic Auditing using Hashdeep Tool

Forensic auditing can be done using the 'hashdeep' tool, which helps check if any files in the system have been altered due to malware or normal system operations like update patching. Below is a detailed explanation of the steps involved:

# 17. Create Hashset and Audit:

## Create Hashset:

`hashdeep -c md5,sha1,sha256 -r /home/user/myfiles > hashset1.txt`  
Explanation:  
- `-c md5,sha1,sha256`: Computes MD5, SHA1, and SHA256 checksums.  
- `-r`: Recursively checks all files in the directory.  
- `/home/user/myfiles`: The directory path to audit.  
- `> hashset1.txt`: Saves the output to `hashset1.txt`.  
  
Purpose: A hashset is created to store checksums, capturing the state of the files.

## Audit Files:

`hashdeep -a -r -k hashset1.txt /home/user/myfiles`  
Explanation:  
- `-a`: Perform an audit (compare against the hashset).  
- `-r`: Recursively check all files.  
- `-k hashset1.txt`: Use the hashset for auditing.  
  
Purpose: Audits the current state of the files against the hashset.

# 18. Audit with New File (Fails):

## Create New File:

`touch /home/user/myfiles/newfile.txt`  
Explanation: Creates a new empty file named `newfile.txt`.  
  
Purpose: Simulates adding a new file to the directory.

## Audit Again:

`hashdeep -a -r -k hashset1.txt /home/user/myfiles`  
Explanation: Re-runs the audit, which fails due to the added file (`newfile.txt`).  
  
Purpose: Detects new files not present in the original hashset.

# 19. Check Failed Points (Verbose):

## Audit with Verbose:

`hashdeep -v -a -r -k hashset1.txt /home/user/myfiles`  
Explanation: The `-v` flag enables verbose mode, providing detailed output.  
  
Purpose: Shows detailed information on audit failures (like added files).

# 20. Audit After Moving File:

## Move File:

`mv /home/user/myfiles/example.txt /tmp`  
Explanation: Moves `example.txt` to the `/tmp` directory.  
  
Purpose: Simulates a file being moved out of the directory.

## Audit Again:

`hashdeep -v -a -r -k hashset1.txt /home/user/myfiles`  
Explanation: The audit fails because `example.txt` is missing.  
  
Purpose: Detects missing files compared to the original hashset.

# 21. Audit After Renaming File:

## Rename File:

`mv /home/user/myfiles/shreya.txt /home/user/myfiles/backup.txt`  
Explanation: Renames `shreya.txt` to `backup.txt`.  
  
Purpose: Tests how the audit behaves when files are renamed.

## Audit Again:

`hashdeep -v -a -r -k hashset1.txt /home/user/myfiles`  
Explanation: The audit detects the renamed file.  
  
Purpose: Shows how renaming files affects the audit.

# 22. Verbose Audit Output:

## More Verbose:

`hashdeep -vv -a -r -k hashset1.txt /home/user/myfiles`  
Explanation: `-vv` shows more detailed output.  
  
Purpose: Increases verbosity for more detailed file changes.

## Very Verbose:

`hashdeep -vvv -a -r -k hashset1.txt /home/user/myfiles`  
Explanation: `-vvv` provides the highest level of detail.  
  
Purpose: Shows maximum information on how files differ from the original hashset.