Name- Anmol Tripathi

Batch-T23

#### Lab Assignment 5

Aim: To explore hashdeep tool in kali linux for generating, matching and auditing hash of files.

LO2: Demonstrate key management, distribution and user authentication.

#### Theory:

What is the need of hashing? List different hashing algorithms.

### **Need for Hashing:**

- 1. Hashing is a process of converting input data (such as text, files, passwords) into a fixed-size value (hash value or hash code) using a hashing algorithm. Hashing serves several purposes in computer science and cybersecurity:
- 2. Data Integrity: Hashing can verify whether data has been altered or corrupted. If the hash value of the original data matches the hash value calculated from the received data, it's likely that the data hasn't been tampered with.
- 3. Password Security: Hashing is used to securely store passwords. Instead of storing plain-text passwords, systems store the hash values of passwords. When a user logs in, the input password's hash is compared with the stored hash.
- 4. Digital Signatures: Hashing plays a role in creating and verifying digital signatures. A hash value of a message is signed with a private key to create a digital signature, providing authenticity and non-repudiation.
- 5. Data Structures: Hashing is used in hash tables, which provide efficient data retrieval. Hash functions map keys to indices in an array, allowing quick access to data.

## **Different Hashing Algorithms:**

There are various hashing algorithms, each with its characteristics and use cases. Some commonly used ones are:

- 1. MD5 (Message Digest Algorithm 5)
- 2. SHA-1 (Secure Hash Algorithm 1)
- 3. SHA-256, SHA-384, SHA-512 (Secure Hash Algorithms 256, 384, 512)
- 4. bcrypt (Adaptive Hashing Algorithm)
- 5. scrypt (Memory-Hard Function)
- 6. Argon2 (Winner of Password Hashing Competition)

Write the commands used for generating hash values, matching them with stored hash values and auditing using hashdeep tool.

- 1. To check the version of Hashdeep hashdeep -V
- 2. To display help about hashdeep hashdeep -h or hashdeep -hh
- 3. To display the manual page of hashdeep- man hashdeep

4. To display the manual page of any specific hash algorithm supported

by hashdeep- man md5deep

By default, hashdeep generates MD5 n SHA256 hash values.

- 5. To hash a file hashdeep filename
- 6. If you don't want to display the full path of file in output hash

record- hashdeep -b filename

- 7. To supress any error messages- hashdeep -s filename
- 8. To apply multiple hash algorithms than defaulthashdeep -c md5,sha1,sha256,tiger filename
- 9. To hash multiple files (say all text files) using md5

hashdeep -c md5 \*.txt

10. To hash multiple files (say all text files) using md5 and sha1

hashdeep -c md5,sha1 \*.txt

- 11. Hashing block of files- hashdeep -c md5 -p 100 example.txt
- 12. To recursively calculate hash (all files and subdirectories in a

specified directory)

hashdeep c md5 -r /home/lab006/myfiles

13. To redirect the output of md5 hash of files to another file

md5deep \*.txt>hashset.txt

hashdeep \*.txt>hashtext1.txt

Check the content of output filecat hashset.txt

cat hashset1.txt

14. To display output in matching mode

md5deep -m hashset.txt \*

hashdeep -m -k hashset1.txt \*

15. To supress unwanted system msgs/error

md5deep -m hashset.txt \*

hashdeep -s -m hashset1.txt \*

No output is displayed if there is no matching hashed file is found.

16. To display all files which are negatively matching use -x option

Md5deep -s -x hashset.txt \*

hashdeep -s -x hashset1.txt \*

Forensic auditing can be done using hashdeep tool which means a check to determine if any files in the system are changed due to malware or any normal system operation like update patching.

17. To audit, first create a hashset file and then audit it against the files to be checked if they are modified.

hashdeep -c md5,sha1,sha256 -r /home/lab006/myfiles>hashset1.txt hashdeep -a -r -k hashset1.txt /home/lab006/myfiles

18. Add new file to the directory and audit. It fails.

touch /home/lab006/myfiles/newfile.txt

hashdeep -a -r -k hashset1.txt /home/lab006/myfiles

19. To get where it failed use the command with -v option

hashdeep -v -a -r -k hashset1.txt/home/lab006/myfiles

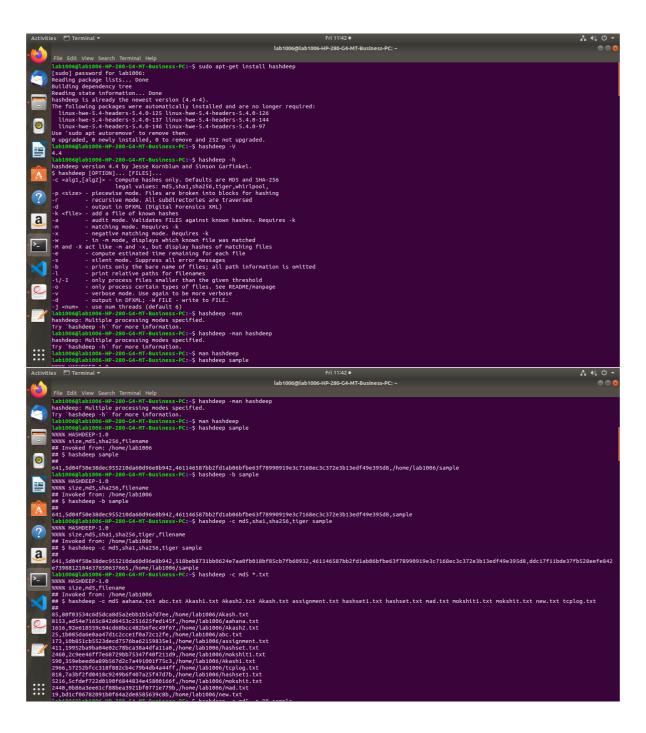
20. Move one of the files to another directory and audit n see output mv /home/lab006/myfiles/example.txt /tmp

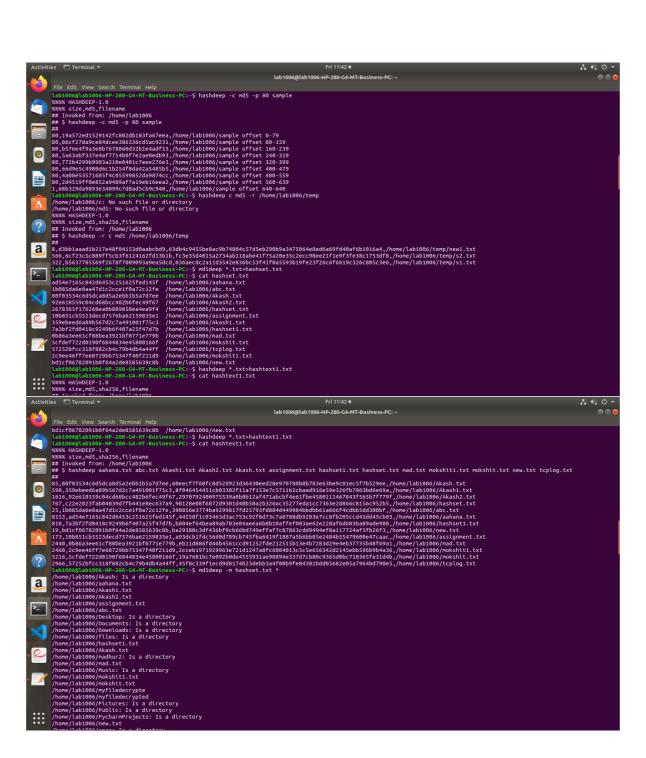
hashdeep -v -a -r -k hashset1.txt /home/lab006/myfiles

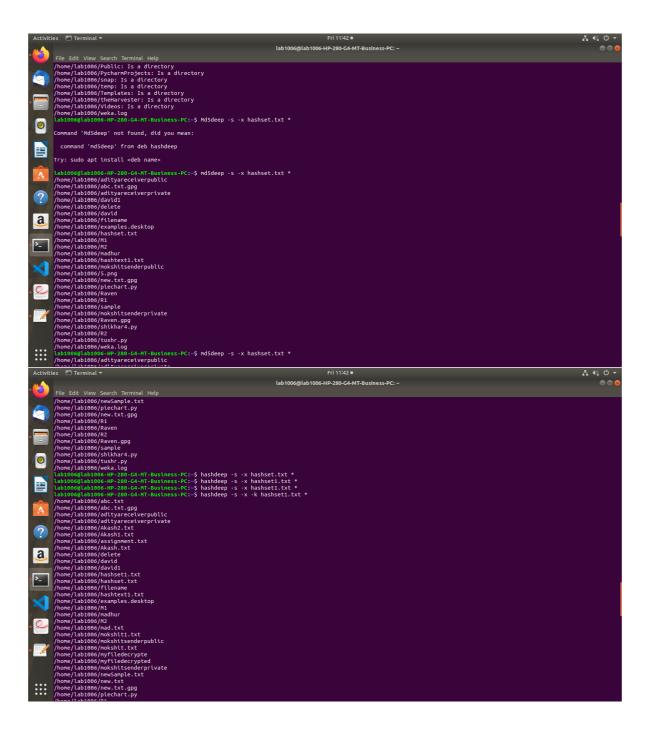
- 21. Rename one of the files and audit n see the output mv /home/lab006/myfiles/lab006.txt /home/lab006/myfiles/lab006.bak hashdeep -v -a -r -k hashset1.txt /home/lab006/myfiles
- 22. For verbose output of audit

hashdeep -vv -a -r -k hashset1.txt /home/lab006/myfiles

hashdeep -vvv -a -r -k hashset1.txt /home/lab006/myfiles







```
we_round: 1
--IPP-288-64-HT-Bustness-PC:-5 hashdeep -vv -a -r -k hashset2.txt /hor
emp/s132.txt: Moved from /home/lab1006/temp/s1.txt
emp/s2.txt: No match
emp/s2.txt: Known file not used
falled
                                              -round-1
-288-G-4MT-Business-PC:-$ hashdeep -vvv -a -r -k hashset2.txt /home/lab1006/temp
/s132.txt: Moved from /home/lab1006/temp/s1.txt
/new1.txt: Ok
/s2.txt: No match
/s2.txt: Known file not used
                                        S-HP-280-G4-NT-Business-PC:-$ mv temp/new1.txt /new
' temp/new1.txt' to '/new': Permission dented
s-HP-280-G4-NT-Business-Pc:-$ mv temp/new1.txt new
s-HP-280-G4-NT-Business-PC:-$ hashdeep -v -a -r -k hashset2.txt /home/lab1006/temp
a
```

# **CONCLUSION:**

In this experiment we studied the need for hashing and its significance in data verification, security, and data structure optimization. Also different hasdeep commands for generating, matching and auditing hash of files.