Evidence Search Techniques

Find command

Objectives

- Understand the goal of search
- Search for files/directories base on names
- Search for files/directories base on their metadata
- Understand various timestamps
- Why should not trust search results

Goal of search

- Retrieve all evidence that is related to crime activities
  - evidence are saved in file systems
- Answer questions
  - who, when, what, how
- Many search commands in Linux
  - ls (list directory contents, simply file name search)
  - *find* (search for file names based on file attributes)
  - grep (search for file names based on file content)
  - locate, which

find command

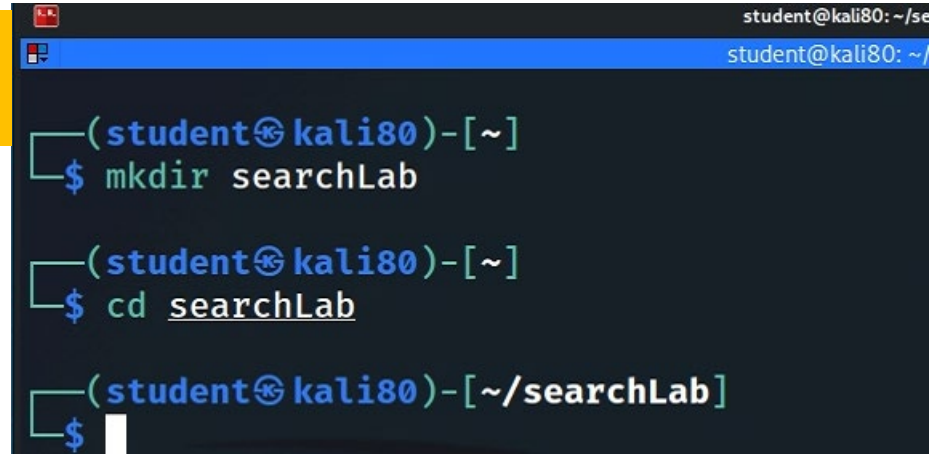
- Search for files by
  - permissions,
  - users, groups,
  - file types,
  - file date,
  - file size, and
  - other possible criteria
- One of the most important commands
  - in computer forensics

Basic *find* command usages

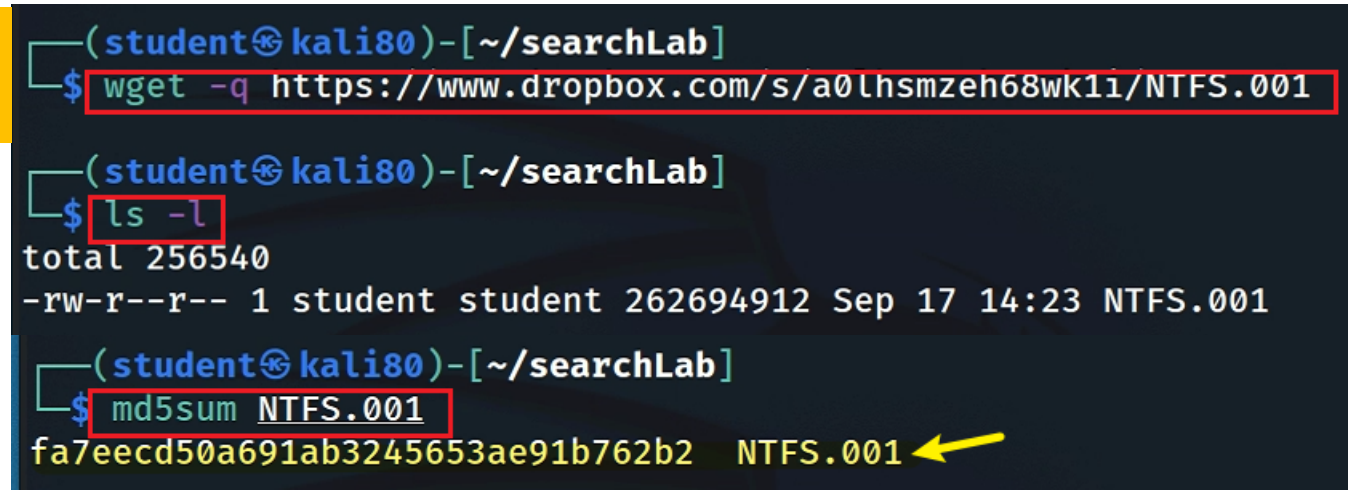
name

The lab uses a USB image

Create a working folder



Download a USB image

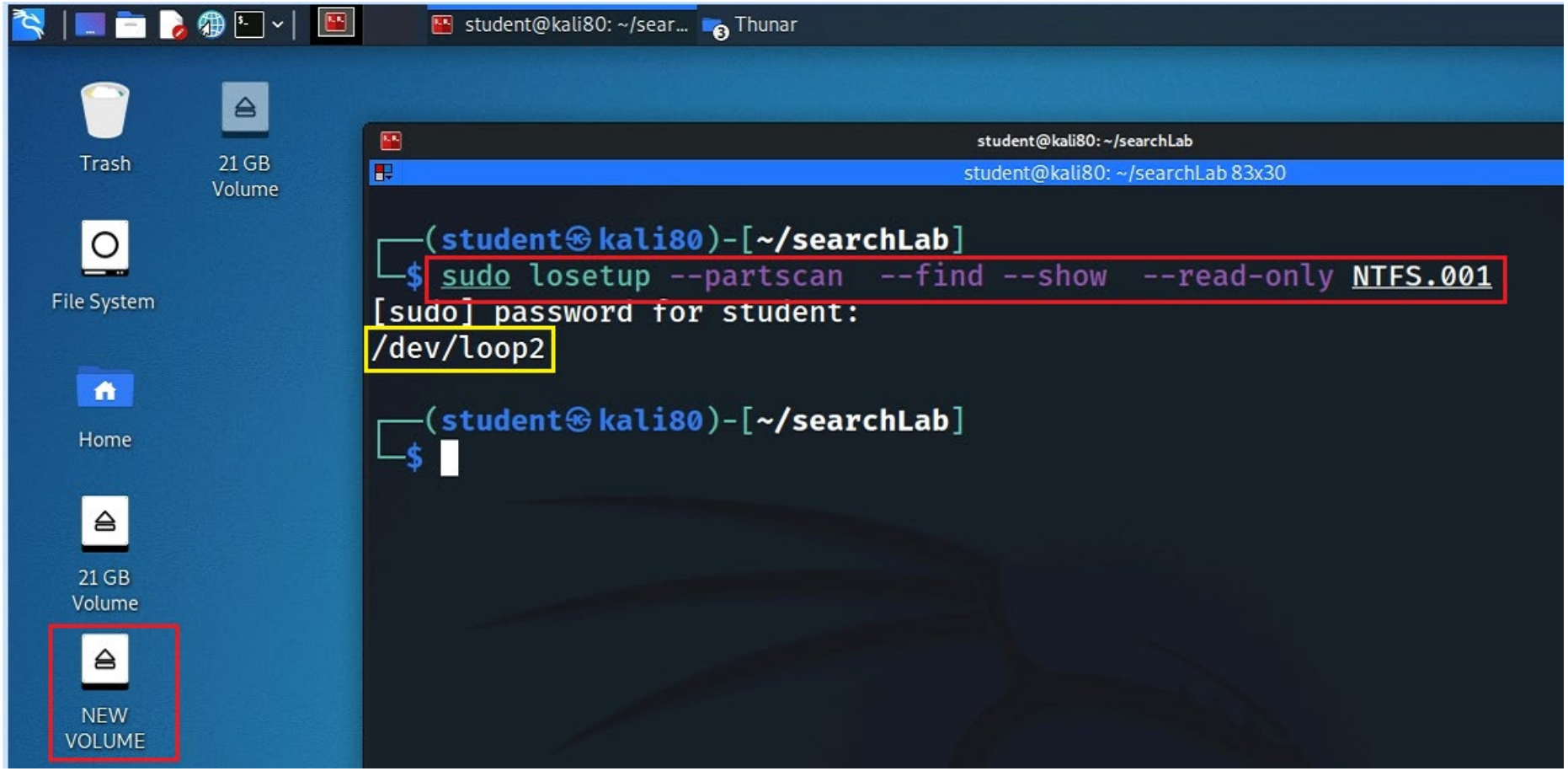


https://www.dropbox.com/s/a0lhsmzeh68wk1i/NTFS.001

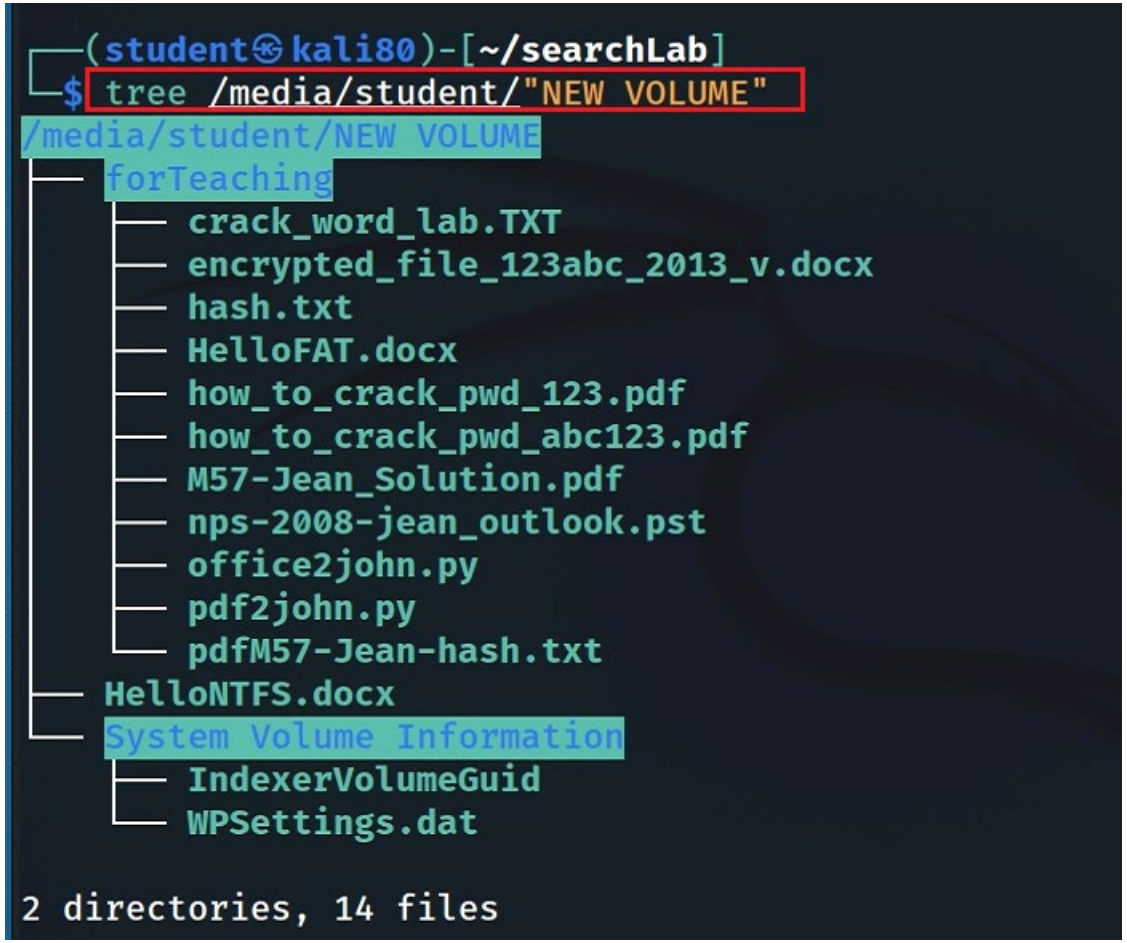
Show image information

```
┌──(student㉿kali80)-[~/searchLab]
└─$ file NTFS.001
NTFS.001: DOS/MBR boot sector MS-MBR Windows 7 english at offset 0x163
"Invalid partition table" at offset 0x17b "Error loading operating syst
em" at offset 0x19a "Missing operating system", disk signature 0x994067
3b; partition 1 : ID=0x7, start-CHS (0x0,2,3), end-CHS (0x1e,254,63), s
tartsector 128, 509952 sectors
```

Mount the USB to Linux as a read-only loop device

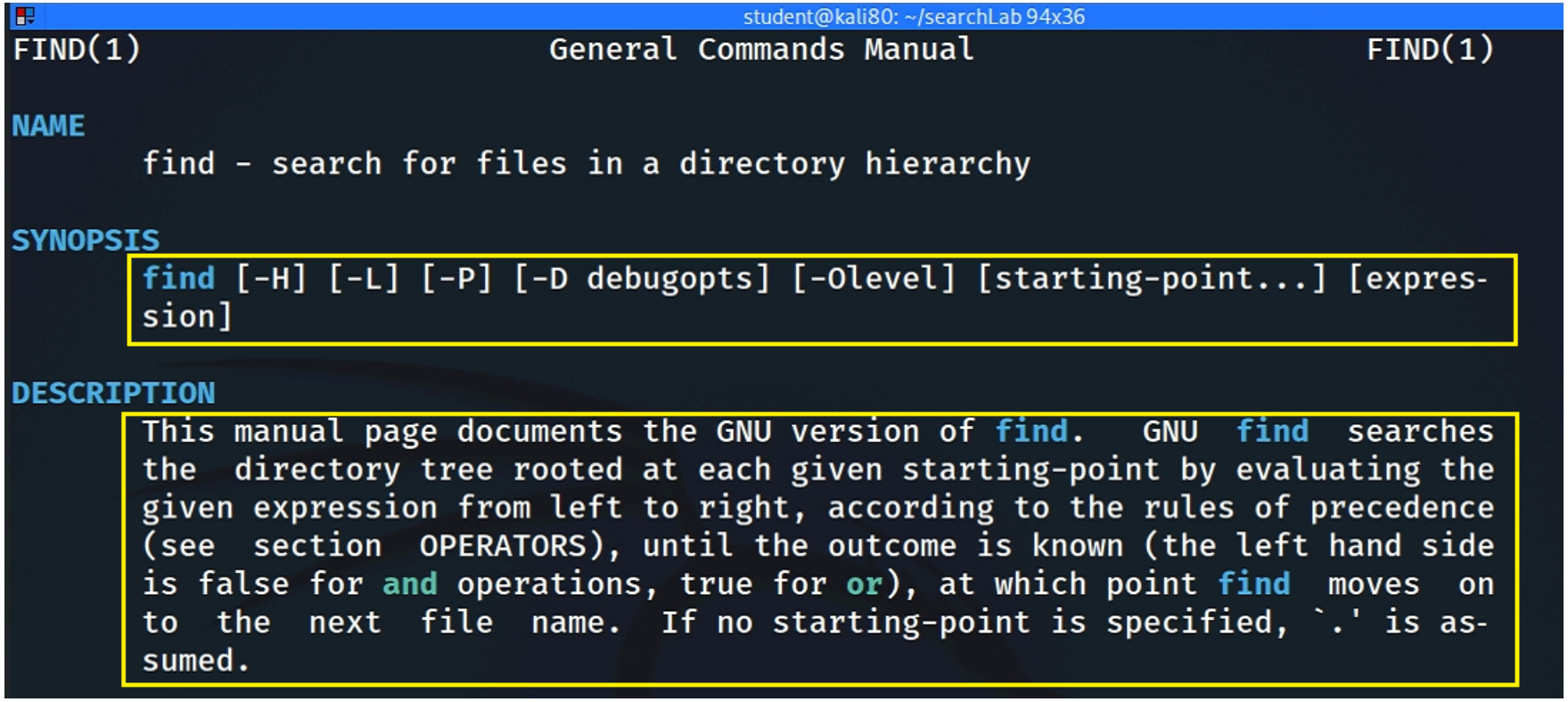


View file structure



Syntax

$ find [where to start searching from] [expression determines what to find]



Find Files/directories starts from Current Directory (recursively)

```
┌──(student㉿kali80)-[~/searchLab]
└─$ cd /media/student/"NEW VOLUME/"

┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find .
.
./forTeaching
./forTeaching/crack_word_lab.TXT
./forTeaching/encrypted_file_123abc_2013_v.docx
./forTeaching/hash.txt
./forTeaching/HelloFAT.docx
./forTeaching/how_to_crack_pwd_123.pdf
./forTeaching/how_to_crack_pwd_abc123.pdf
./forTeaching/M57-Jean_Solution.pdf
./forTeaching/nps-2008-jean_outlook.pst
./forTeaching/office2john.py
./forTeaching/pdf2john.py
./forTeaching/pdfM57-Jean-hash.txt
./HelloNTFS.docx
./System Volume Information
./System Volume Information/IndexerVolumeGuid
./System Volume Information/WPSettings.dat
```

make sure you are in the directory

include all subfolders, recursively

Default is current directory. The command is the same as

```
└─$ find
```

-name and *-iname*: find file/directory names

find a file

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find ./forTeaching  -name hash.txt
./forTeaching/hash.txt

┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find .  -name hash.txt
./forTeaching/hash.txt
```

note: only name without path

find a file and ignore case

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find .  -name hash.TXT
                              <- no return because case-sensitive

┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find .  -iname hash.TXT
./forTeaching/hash.txt        <- ignore case
```

find a directory

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find .  -name forTeaching
./forTeaching <-
```

-type: specify string types

Specify the search type *d: directory*

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find . -type d -name forTeaching
./forTeaching
```

Specify the search type *f: file*

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find . -type f -name hash.txt
./forTeaching/hash.txt
```

Search all files with extension *.txt*

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find . -type f -name "*.txt"
./forTeaching/hash.txt
./forTeaching/pdfM57-Jean-hash.txt
```

Match all file names starts with the letter "*h*"

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find . -type f -name "h*"
./forTeaching/hash.txt
./forTeaching/how_to_crack_pwd_123.pdf
./forTeaching/how_to_crack_pwd_abc123.pdf
```

Match all files which first and four letters are "*h*"

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find . -type f -name "h??h*"
./forTeaching/hash.txt
```

Match all files contains "*123*" (in the middle of the file name)

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find . -type f -name "*123*"
./forTeaching/encrypted_file_123abc_2013_v.docx
./forTeaching/how_to_crack_pwd_123.pdf
./forTeaching/how_to_crack_pwd_abc123.pdf
```

Match on the whole path for *-regex*

Match all files which name ends with any length of digits

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find . -type f  -regextype egrep -regex  ".*[0-9]+\..*"
./forTeaching/how_to_crack_pwd_123.pdf
./forTeaching/how_to_crack_pwd_abc123.pdf
```

Escape Sequences

\	Escape following character
\Q	Begin literal sequence
\E	End literal sequence

"Escaping" is a way of treating characters which have a special meaning in regular expressions literally, rather than as special characters.

Quantifiers

*	0 or more	{3}	Exactly 3
+	1 or more	{3,}	3 or more
?	0 or 1	{3,5}	3, 4 or 5

Add a ? to a quantifier to make it ungreedy.

Groups and Ranges

.	Any character except new line (\n)
(a|b)	a or b
(...)	Group
(?:...)	Passive (non-capturing) group
[abc]	Range (a or b or c)
[^abc]	Not (a or b or c)
[a-q]	Lower case letter from a to q
[A-Q]	Upper case letter from A to Q
[0-7]	Digit from 0 to 7
\x	Group/subpattern number "x"

Ranges are inclusive.

Match exact three up case in the path, excluding file extension.

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find . -type f   -regextype egrep -regex ".*[A-Z]{3}.*\..*"
./forTeaching/HelloFAT.docx
./HelloNTFS.docx
./System Volume Information/WPSettings.dat
```

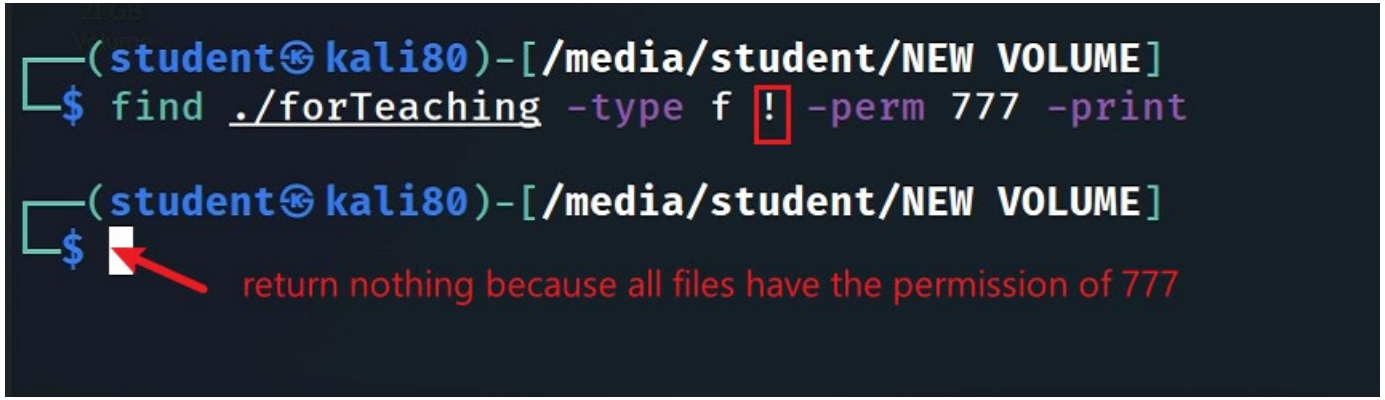
metadata and find usage

timestamps/permissions/users

Find Files With 777 Permissions

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ find ./forTeaching -type f -perm 777 -print
./forTeaching/crack_word_lab.TXT
./forTeaching/encrypted_file_123abc_2013_v.docx
./forTeaching/hash.txt
./forTeaching/HelloFAT.docx
./forTeaching/how_to_crack_pwd_123.pdf
./forTeaching/how_to_crack_pwd_abc123.pdf
./forTeaching/M57-Jean_Solution.pdf
./forTeaching/nps-2008-jean_outlook.pst
./forTeaching/office2john.py
./forTeaching/pdf2john.py
./forTeaching/pdfM57-Jean-hash.txt

┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ ls -l ./forTeaching
total 4082
-rwxrwxrwx 1 student student      57 Oct  2  2018 crack_word_lab.TXT
-rwxrwxrwx 1 student student   17920 Oct  2  2018 encrypted_file_123abc_2013_v.docx
-rwxrwxrwx 1 student student     194 Oct  2  2018 hash.txt
-rwxrwxrwx 1 student student   11514 Oct  8  2018 HelloFAT.docx
-rwxrwxrwx 1 student student 1185208 Oct  1  2018 how_to_crack_pwd_123.pdf
-rwxrwxrwx 1 student student  380207 Oct  1  2018 how_to_crack_pwd_abc123.pdf
-rwxrwxrwx 1 student student   96290 Oct  5  2018 M57-Jean_Solution.pdf
-rwxrwxrwx 1 student student 2326528 Jul 20  2008 nps-2008-jean_outlook.pst
-rwxrwxrwx 1 student student  134772 Oct  2  2018 office2john.py
-rwxrwxrwx 1 student student   13904 Oct  5  2018 pdf2john.py
-rwxrwxrwx 1 student student     191 Oct  5  2018 pdfM57-Jean-hash.txt
```

Find Files Without 777 Permissions



Note: we can't change the permission of files here because we mounted the image as read only

“atime”, “mtime” and “ctime”

Change working directory

Create a new file

Review the permission and timestamps

Modified a line

Review the permission and timestamps

Access the file content

```
┌──(student㉿kali80)-[/media/student/NEW VOLUME]
└─$ cd ~/searchLab

┌──(student㉿kali80)-[~/searchLab]
└─$ echo hello world > myFile.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ ls -l
total 256544
-rw-r--r-- 1 student student        12 Sep 17 21:15 myFile.txt
-rw-r--r-- 1 student student 262694912 Sep 17 14:23 NTFS.001

┌──(student㉿kali80)-[~/searchLab]
└─$ echo hello world again! >> myFile.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ ls -l
total 256544
-rw-r--r-- 1 student student        31 Sep 17 21:17 myFile.txt
-rw-r--r-- 1 student student 262694912 Sep 17 14:23 NTFS.001

┌──(student㉿kali80)-[~/searchLab]
└─$ cat myFile.txt
hello world
hello world again!
```

Difference Between “atime”, “mtime” and “ctime”

```
┌──(student㉿kali80)-[~/searchLab]
└─$ stat myFile.txt
  File: myFile.txt
  Size: 31              Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 4594300     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ student)   Gid: ( 1000/ student)
Access: 2021-09-17 21:18:21.465102327 -0400
Modify: 2021-09-17 21:17:54.153103104 -0400
Change: 2021-09-17 21:17:54.153103104 -0400
 Birth: 2021-09-17 21:15:57.617106417 -0400
```

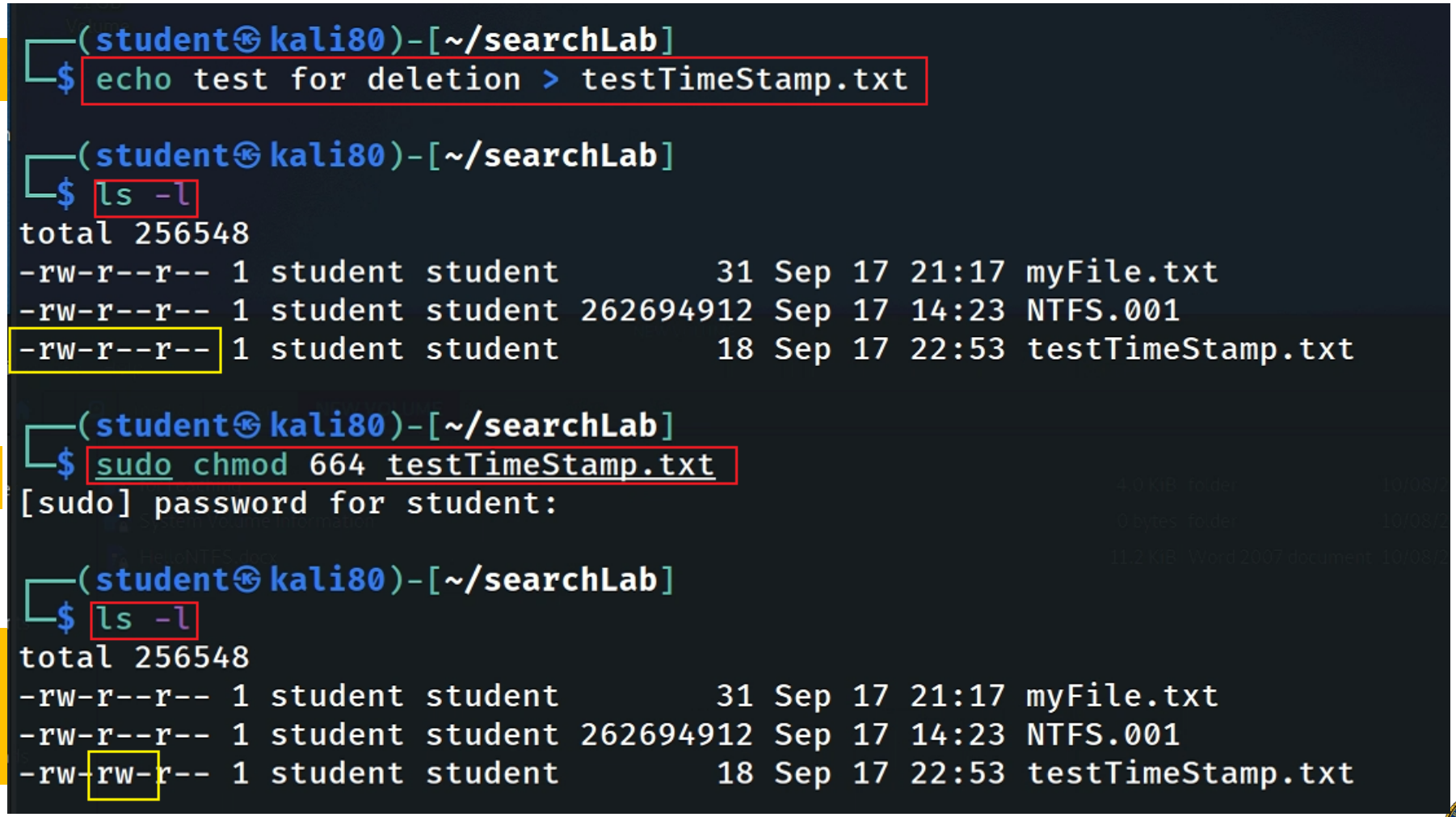
Timestamp	When it gets updated?
atime	**Access time** gets updated when you open a file or when a file is used for other operations like grep, cat, head and so on (not for stat command!).
mtime	**Modify time** gets updated when you whenever update content of a file or save a file.
ctime	**Change time** gets updated when the file attributes are changed, like changing the owner, changing the permission or moving it to another filesystem, but will also be updated when you modify a file.

Timestamp changes when changing permission

Create a new file

Change permission

Change permission doesn't change the timestamp (content)



```
┌──(student㉿kali80)-[~/searchLab]
└─$ stat testTimeStamp.txt
  File: testTimeStamp.txt
  Size: 18              Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 4594301     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ student)   Gid: ( 1000/ student)
Access: 2021-09-17 22:53:24.132940182 -0400
Modify: 2021-09-17 22:53:24.132940182 -0400
Change: 2021-09-17 22:54:13.552938777 -0400
 Birth: 2021-09-17 22:53:24.132940182 -0400
```

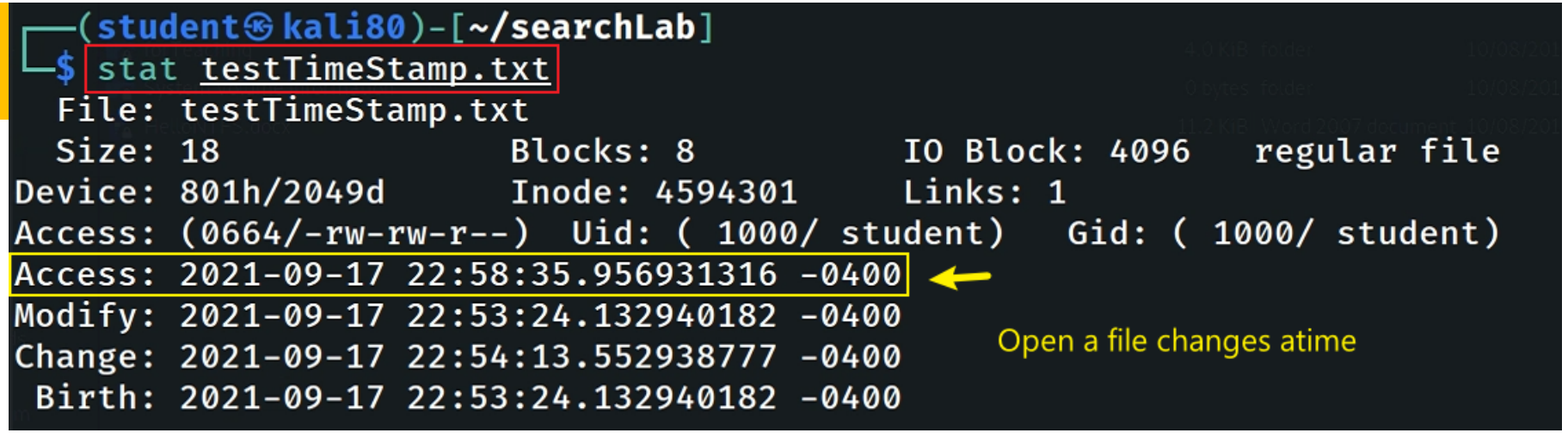
chnage permission

note: we did not modify the content of the file, so birth time=Modify time

View (Access) the file with leafpad

Note: we did not change the content

Check timestamps



Note: mtime doesn't change, but atime changed

Find Files With *–perm=xxx* Permissions

```
┌──(student㉿kali80)-[~/searchLab]
└─$ ls -l
total 256548
-rw-r--r-- 1 student student        31 Sep 17 21:17 myFile.txt
-rw-r--r-- 1 student student 262694912 Sep 17 14:23 NTFS.001
-rw-rw-r-- 1 student student        18 Sep 17 22:53 testTimeStamp.txt
```

```
┌──(student㉿kali80)-[~/searchLab]
└─$ find . -type f -perm 664
./testTimeStamp.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ find . -type f -perm 644
./NTFS.001
./myFile.txt
```

Find Files With */u=r /u=w /u=x* Permissions

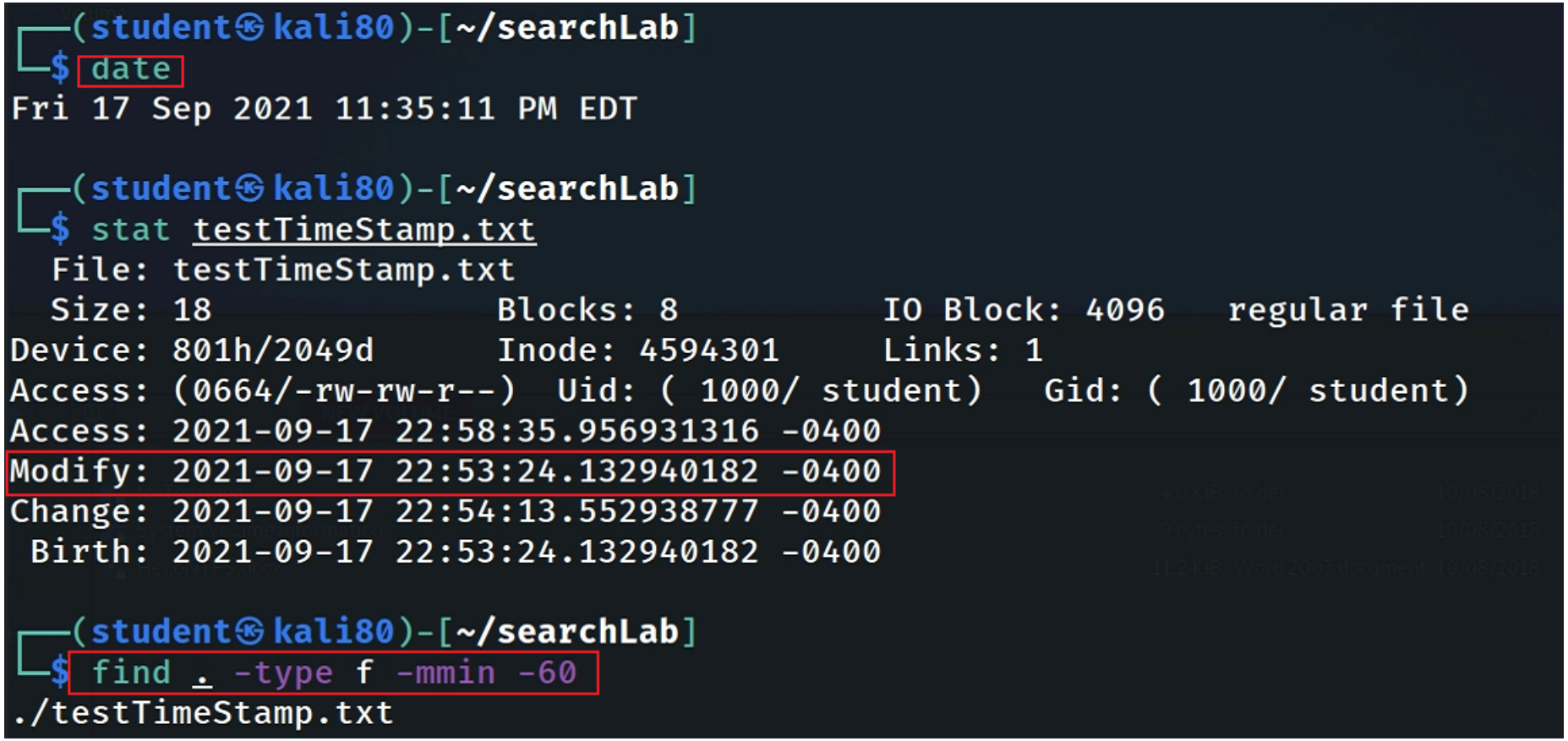
```
┌──(student㉿kali80)-[~/searchLab]
└─$ ls -l
total 256548
-rw-r--r-- 1 student student        31 Sep 17 21:17 myFile.txt
-rw-r--r-- 1 student student 262694912 Sep 17 14:23 NTFS.001
-rw-rw-r-- 1 student student        18 Sep 17 22:53 testTimeStamp.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ find . -type f -perm /u=r      read
./testTimeStamp.txt
./NTFS.001
ious visited folder.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ find . -type f -perm /u=w      write
./testTimeStamp.txt
./NTFS.001
./myFile.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ find . -type f -perm /u=x      execute

┌──(student㉿kali80)-[~/searchLab]
└─$
```

Find Changed Files in Last 1 Hour

+n	for greater than n,
-n	for less than n,
n	for exactly n.

```
┌──(student㉿kali80)-[~/searchLab]
└─$ date
Fri 17 Sep 2021 11:31:53 PM EDT

┌──(student㉿kali80)-[~/searchLab]
└─$ find . -type f -cmin -60
./testTimeStamp.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ stat testTimeStamp.txt
  File: testTimeStamp.txt
  Size: 18              Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 4594301     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ student)   Gid: ( 1000/ student)
Access: 2021-09-17 22:58:35.956931316 -0400
Modify: 2021-09-17 22:53:24.132940182 -0400
Change: 2021-09-17 22:54:13.552938777 -0400
 Birth: 2021-09-17 22:53:24.132940182 -0400
```

Find Modified Files in Last 1 Hour



Find Accessed Files in Last 1 Hour

```
┌──(student㉿kali80)-[~/searchLab]
└─$ date
Fri 17 Sep 2021 11:38:16 PM EDT

┌──(student㉿kali80)-[~/searchLab]
└─$ find . -type f -amin -60
./testTimeStamp.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ stat testTimeStamp.txt
  File: testTimeStamp.txt
  Size: 18              Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 4594301     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ student)   Gid: ( 1000/ student)
Access: 2021-09-17 22:58:35.956931316 -0400
Modify: 2021-09-17 22:53:24.132940182 -0400
Change: 2021-09-17 22:54:13.552938777 -0400
 Birth: 2021-09-17 22:53:24.132940182 -0400
```

File was last accessed less than n*24 hours ago

```
┌──(student㉿kali80)-[~/searchLab]
└─$ sudo find / -type f -atime -1 | grep "test"
find: ‘/home/student/thinclient_drives’: Permission denied
/home/student/searchLab/testTimeStamp.txt
find: ‘/run/user/1000/gvfs’: Permission denied
find: ‘/run/user/133/gvfs’: Permission denied
find: ‘/proc/1475338/task/1475338/fdinfo/5’: No such file or directory
find: ‘/proc/1475338/fdinfo/6’: No such file or directory
/var/lib/dpkg/info/ruby-minitest.list
/var/lib/dpkg/info/ruby-test-unit.list
/var/lib/dpkg/info/davtest.list
```

Verify file was last accessed less than n*24 hours ago

```
┌──(student㉿kali80)-[~/searchLab]
└─$ stat /var/lib/dpkg/info/ruby-minitest.list
  File: /var/lib/dpkg/info/ruby-minitest.list
  Size: 1219            Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 2756424     Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2021-09-18 00:06:19.272815783 -0400
Modify: 2021-08-26 09:09:29.199343555 -0400
Change: 2021-08-26 09:09:29.199343555 -0400
 Birth: 2021-08-26 09:09:29.199343555 -0400

┌──(student㉿kali80)-[~/searchLab]
└─$ date
Sat 18 Sep 2021 12:23:54 AM EDT
```

File was last modified/changed less than n*24 hours ago

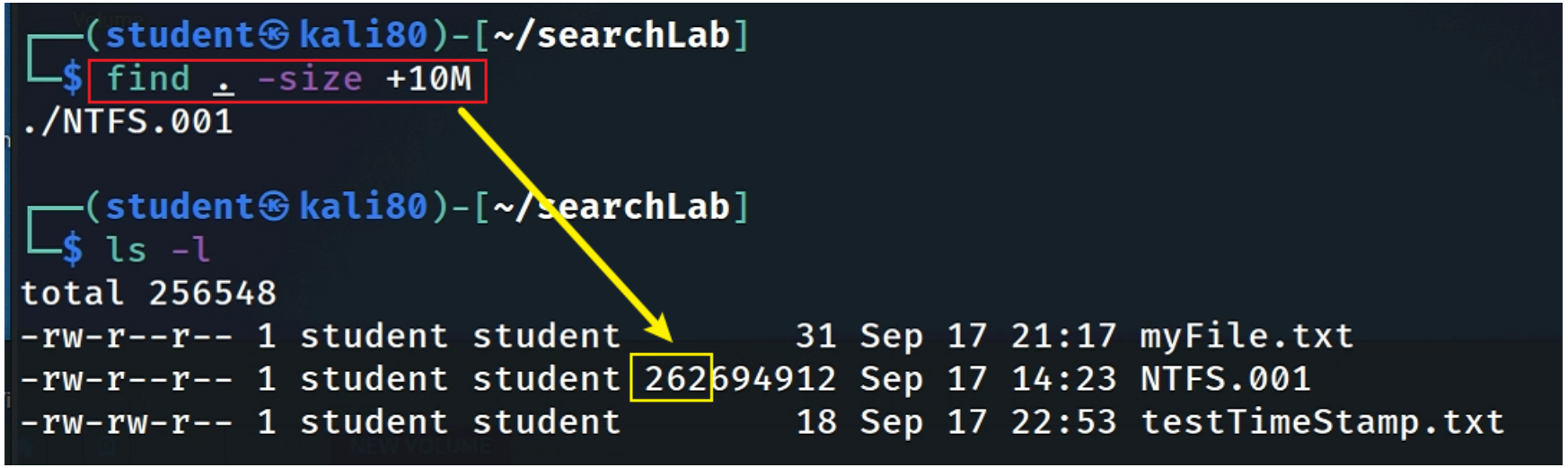
```
$ sudo find / -type f -mtime -1
```

```
$ sudo find / -type f -ctime -1
```

try by yourself and verify your command!

Find Size great than 10MB

+n	for greater than n,
-n	for less than n,
n	for exactly n.

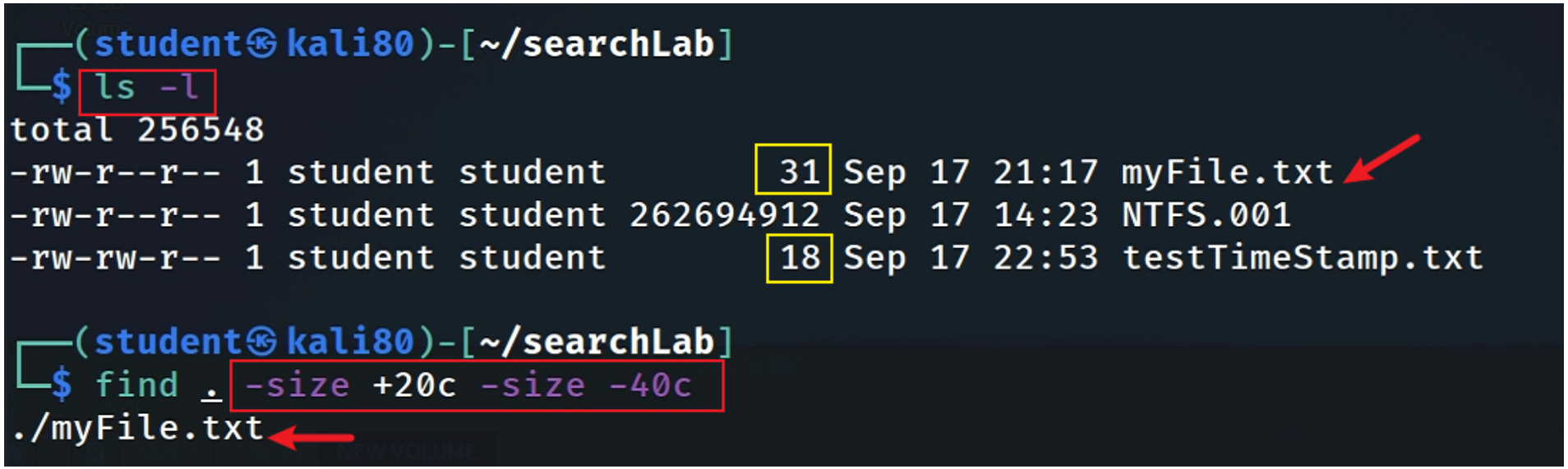


Find Size between 10MB and 300MB

```
┌──(student㉿kali80)-[~/searchLab]
└─$ find . -size +10M -size -300M
./NTFS.001

┌──(student㉿kali80)-[~/searchLab]
└─$ ls -l
total 256548
-rw-r--r-- 1 student student        31 Sep 17 21:17 myFile.txt
-rw-r--r-- 1 student student 262694912 Sep 17 14:23 NTFS.001
-rw-rw-r-- 1 student student        18 Sep 17 22:53 testTimeStamp.txt
```

```
-size n[cwbkMG]
       File uses less than, more than or exactly n units of space, rounding
       up.  The following suffixes can be used:

       `b'    for  512-byte  blocks  (this  is  the default if no suffix is
              used)

       `c'    for bytes

       `w'    for two-byte words

       `k'    for kibibytes (KiB, units of 1024 bytes)

       `M'    for mebibytes (MiB, units of 1024 * 1024 = 1048576 bytes)

       `G'    for gibibytes (GiB, units of 1024 * 1024 * 1024 =  1073741824
              bytes)
```

Find Size between 20 – 40 bytes



Find Particular Files of User

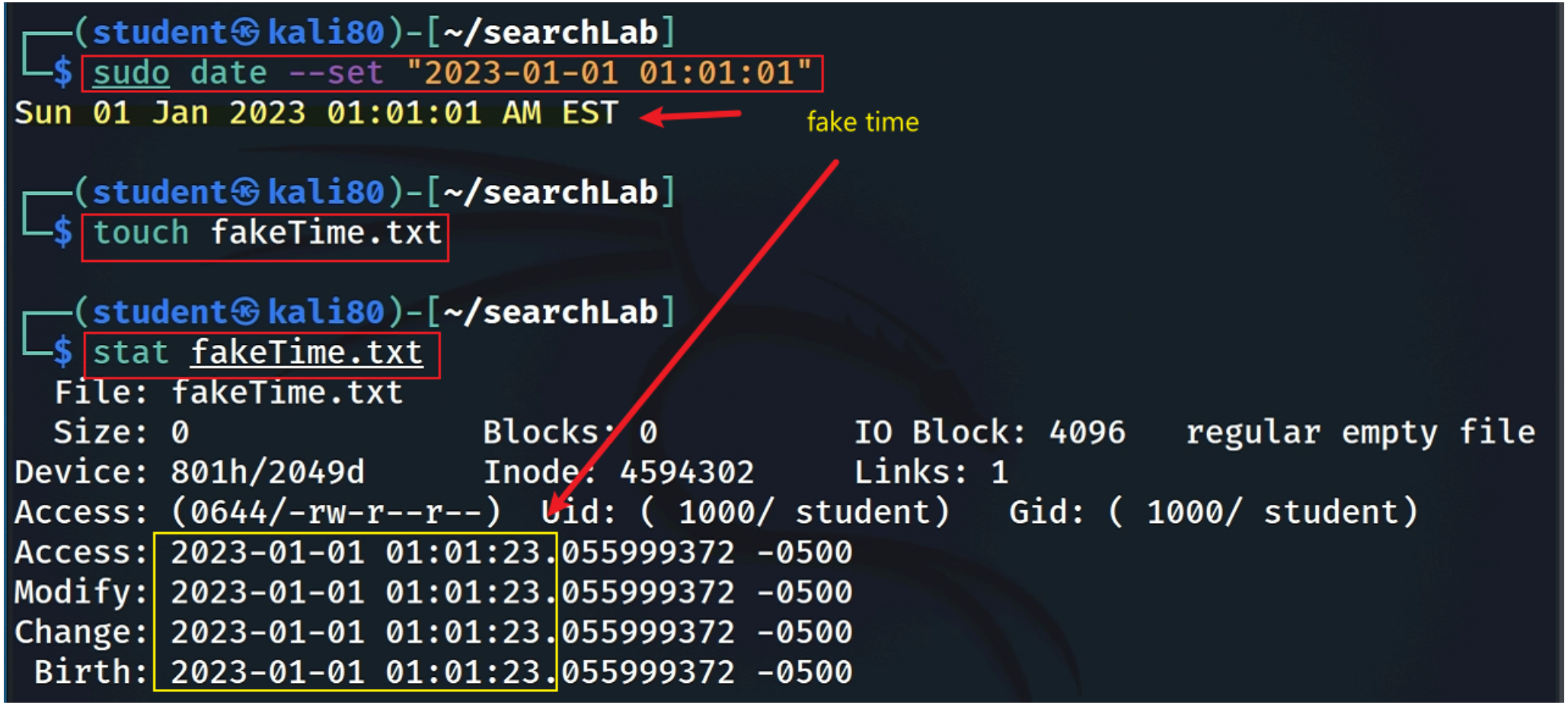
To find all .txt files of user *student* under /home directory.

```
┌──(student㉿kali80)-[~/searchLab]
└─$ find /home -user student -iname "*.txt"
/home/student/.pki/nssdb/pkcs11.txt
/home/student/received.txt
/home/student/.local/share/Trash/files/myFile.txt
/home/student/.local/lib/python3.9/site-packages/time_de
l.txt
/home/student/myFolder/received.txt
/home/student/myFolder/myTest2.txt
/home/student/myFolder/myTest.txt
/home/student/myFolder/error.txt
/home/student/myFolder/display.txt
/home/student/myFolder/myFile.txt
/home/student/searchLab/testTimeStamp.txt
/home/student/searchLab/myFile.txt
```

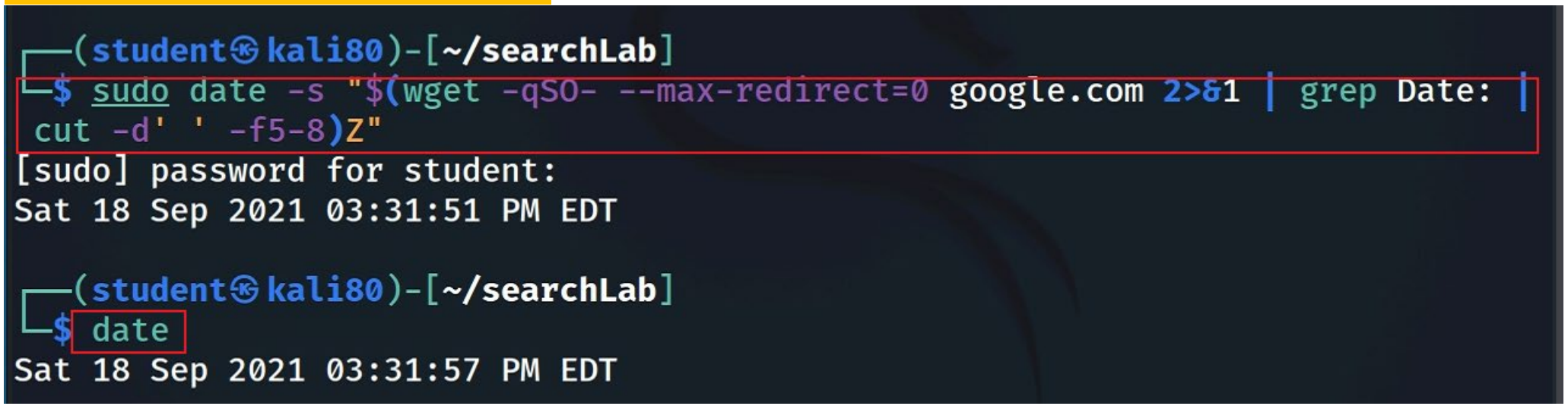
Why should not trust search results?

anti-forensic

Set fake *[amc]time* by changing system time



Reset time back to correct time



sudo date -s "$(wget -qSO- --max-redirect=0 google.com 2>&1 | grep Date: | cut -d' ' -f5-8)Z"

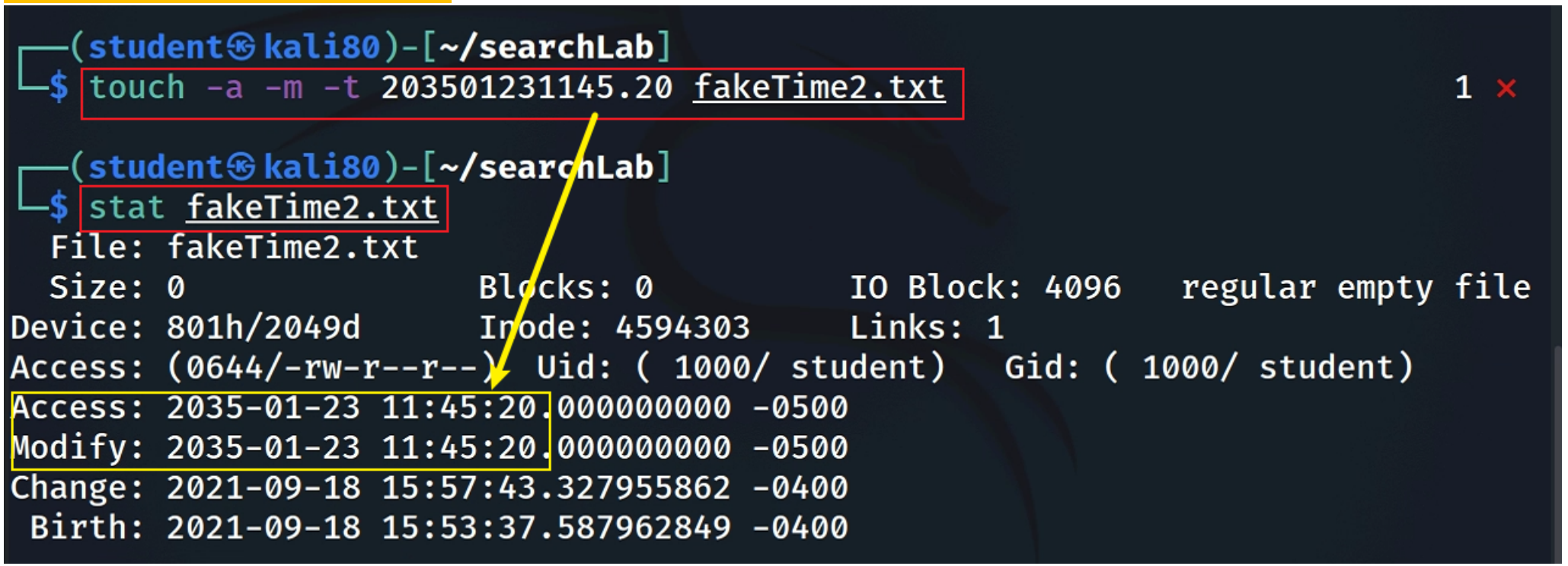
Set fake time using *touch -t*

Create a normal file

```
┌──(student㉿kali80)-[~/searchLab]
└─$ touch fakeTime2.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ stat fakeTime2.txt
  File: fakeTime2.txt
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 801h/2049d      Inode: 4594303     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ student)   Gid: ( 1000/ student)
Access: 2021-09-18 15:53:37.587962849 -0400
Modify: 2021-09-18 15:53:37.587962849 -0400
Change: 2021-09-18 15:53:37.587962849 -0400
 Birth: 2021-09-18 15:53:37.587962849 -0400
```

Create a normal file

touch -t [[CC]YY]MMDDhhmm[.SS]



touch -a -m -t 203501231145.20 fakeTime2.txt

Copy timestamp from Another File *-r*

Create a normal file

Copy timestamp for fakeTimes.txt

```
┌──(student㉿kali80)-[~/searchLab]
└─$ touch fakeTime3.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ stat fakeTime3.txt
  File: fakeTime3.txt
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 801h/2049d      Inode: 4594304     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ student)   Gid: ( 1000/ student)
Access: 2021-09-18 18:22:44.567708458 -0400
Modify: 2021-09-18 18:22:44.567708458 -0400
Change: 2021-09-18 18:22:44.567708458 -0400
 Birth: 2021-09-18 18:22:44.567708458 -0400

┌──(student㉿kali80)-[~/searchLab]
└─$ touch fakeTime3.txt -r fakeTime2.txt

┌──(student㉿kali80)-[~/searchLab]
└─$ stat fakeTime3.txt
  File: fakeTime3.txt
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 801h/2049d      Inode: 4594304     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ student)   Gid: ( 1000/ student)
Access: 2035-01-23 11:45:20.000000000 -0500
Modify: 2035-01-23 11:45:20.000000000 -0500
Change: 2021-09-18 18:23:32.051707108 -0400
 Birth: 2021-09-18 18:22:44.567708458 -0400
```

Anti-anti-forensics

```
┌──(student㉿kali80)-[~/searchLab]
└─$ history | tail
 1068  stat fakeTime.txt
 1069  sudo date -s "$(wget -qSO- --max-redirect=0 google.com 2>&1 | grep Date
: | cut -d' ' -f5-8)Z"
 1070  date
 1071  touch fakeTime2.txt
 1072  stat fakeTime2.txt
 1073  touch -a -m -t 2035012311.20 fakeTime2.txt
 1074  touch -a -m -t 203501231145.20 fakeTime2.txt
 1075  stat fakeTime2.txt
 1076  clear
 1077  history
```