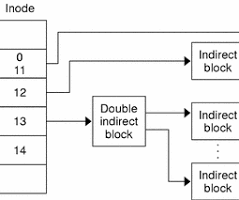
OVER THE WIRE  
30 sept   
  
whats inode ? [Grab your reader’s attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]



find . -inum 123456

(inum = inode num)

An inode, or **index node**, is a data structure in a Unix-style file system that stores information about a file system object, such as a file or directory. It contains all the metadata for a file, except for the file's actual name and its content.

**What is stored in an inode?**

Each inode is identified by a unique inode number within its file system. This inode stores the following crucial metadata:

* **File type**: For example, is it a regular file, directory, or symbolic link?
* **Permissions**: Defines who can read, write, or execute the file.
* **Ownership**: The User ID (UID) and Group ID (GID) of the file's owner.
* **File size**: The size of the file in bytes.
* **Timestamps**: The last time the file was accessed (atime), modified (mtime), and the inode itself was changed (ctime).
* **Link count**: The number of hard links pointing to the inode.
* **Data block pointers**: Pointers to the physical data blocks on the disk that contain the file's actual content.

**Inode vs. file name**

A key concept in file systems is the separation of the file name and the inode.

* **The directory** is what links a human-readable **file name** to its corresponding **inode number**.
* **The inode** itself does not contain the file name.

“-“ Filename?

and "-" filename?

In the context of the command line, a single dash

- is not a filename but a special character that often refers to standard input or standard output, depending on the command. This provides a flexible way to pipe data between commands without creating temporary files.

**Standard input**

When a command expects input from a file, using - as the filename tells the command to read its input from the standard input stream instead.

**Example:**  
To pass text directly from the keyboard to the cat command, you can use cat -.

bash

$ cat -

Hello, world!

This is standard input.

(Press Ctrl+D to send an end-of-file signal)

Level 4:

USE THE FILE ./\* COMMAND TO CHECK THE DETALIS OF FILES OF ALL FILES IN SAME DIR

Level 5   
I used ls -ilR , to find all the files and their sub file’s name , size , inode nums

I used ls -iRl to find the size , inode num and files under files but didn’t get it   
  
solution: find . -type f -size 1033c -not -executable -exec file {} + | grep ASCII HWasnPhtq9AVKe0dmk45nxy20cvUa6EG

* find .
  + **find**: The command used to search for files in a directory hierarchy.
  + **.**: The starting point for the search. The dot . represents the current directory.
* -type f
  + **-type**: The flag to specify the file type.
  + **f**: Restricts the search to regular files, excluding directories, links, and other types.
* -size 1033c
  + **-size**: The flag to specify the size of the file.
  + **1033c**: Searches for files that are exactly 1033 bytes in size. The c suffix ensures the size is counted in bytes.
* -not -executable
  + **-not**: A logical operator that negates the next condition.
  + **-executable**: The test that checks if the file has its executable permission bit set.
  + **-not -executable**: Together, this combination finds files that are not executable.
* -exec file {} +
  + **-exec**: The action to execute a command on the files that match the previous criteria.
  + **file**: The command to run on each found file. The file utility attempts to classify the type of content in a file.
  + **{}**: A placeholder that find replaces with the name of each file it finds.
  + **+**: An important terminator that tells find to pass all the found files to a *single* invocation of the file command, instead of running file once for each result. This is more efficient for a large number of files.

Level 6   
I used **find / -type f -user bandit7 -group bandit6 -size 33c 2> /dev/null**

* **find /**: This begins the search from the root directory (/), which includes all files and subdirectories on the system.
* **-type f**: This option narrows the search to only include regular files (f), excluding directories, symbolic links, and other file types.
* **-user bandit7**: This specifies that the file must be owned by the user bandit7.
* **-group bandit6**: This specifies that the file must belong to the group bandit6.
* **-size 33c**: This filters for files that are exactly 33 bytes in size. The c suffix indicates that the size is specified in bytes.
* **2> /dev/null**: This redirects standard error (represented by the file descriptor 2) to /dev/null, which discards any error messages. This is particularly useful when searching the entire system (/) to prevent "Permission denied" errors from cluttering the output.

**Here, cat will take its input from input.txt instead of waiting for you to type.**

**Cat < hi.txt**

**Here , hi.txt will ask to be inputted**

**Cat>hi.txt**

**This > will overwrite**

**Cat >>hi.txt , >> will allow new input from new line**

So this < is basically “0>” , and < is basically “1>”   
and sending output to somewhere else ? use “2>”

Lvl8 :

to **find a line in a file that appears only once** (a “lonely” line).

sort file.txt | uniq -u

**uniq -u**  prints only lines that are unique

**sort**

ensures duplicate lines are grouped together

level9

first we decrypct the file to human redable texts using: then we grep “==”  
cat filename | strings -e s | grep ==

The -e flag is used to specify the character encoding. We are assuming the human readable line is ASCII text so we use “s” for the encoding type *(Refer attached resources for more information)*

**-e option formats**

According to GNU strings (from GNU binutils), the common encodings are:

* s → 7-bit **ASCII** (default)
* S → 8-bit **ISO-8859-1** (Latin-1)
* b → 16-bit **big-endian** (UTF-16BE style)
* l → 16-bit **little-endian** (UTF-16LE style)
* B → 32-bit **big-endian** (UTF-32BE style)
* L → 32-bit **little-endian** (UTF-32LE style)

Level10 : base64 encoded means **Base64 encoding** is a way to represent **binary data** (like images, files, or any arbitrary bytes) as **plain text** using only 64 safe characters.

**cat filename | base64**  to encode file to base64

**cat filename | base64 -d** here d means decode , so its for decoding