2025

Research And Explain What The Umask Command Is In Linux, And How To Use It

MOSSÉ CYBERSECURITY INSTITUTE

AARON AMRAN BIN AMIRUDDIN

Student ID: nxCLnZGLgyOUMpnDw16rtDvYuTF2

Table of Contents

What is umask?

Umask (short for "user file-creation mode mask") is a setting in Unix/Linux systems that controls the default permissions of newly created files and directories.

It is a value (typically expressed in octal notation) that defines which permission bits should not be set on newly created files and directories. It acts as a filter, “masking out” specific permission bits.

How does umask work?

The umask command specifies the permissions that the user does not want to be given out to the newly created file or directory. umask works by doing a Bitwise AND with the bitwise complement(where the bits are inverted, i.e. 1 becomes 0 and 0 becomes 1) of the umask.

The bits which are set in the umask value, refer to the permissions, which are not assigned by default, as these values are subtracted from the maximum permission for files/directories.

What are the three permissions applicable to files and directories?

- Read (r): Permission to read the file or list the contents of a directory.
- Write (w): Permission to modify the file or add/remove files in a directory.
- Execute (x): Permission to run the file as a program or traverse a directory.

These permissions are applied to three categories of users:

- Owner (User)
- Group
- Others

Permissions	Octal Value	Binary Value	Description
---	0	000	No permission
--x	1	001	only permission to execute
-w-	2	010	only permission to write
-wx	3	011	permission to write and execute
r--	4	100	only permission to read
r-x	5	101	permission to read and execute
rw-	6	110	permission to read and write
rwx	7	111	permission to do all three, i.e. read, write and execute

What is the default umask value when we create a file or a folder?

The default umask value on many Unix/Linux systems is typically 022. Here's what that means:

- **Subtraction from Default Permissions:**
  Files are generally created with a base permission of 666 and directories with 777. The umask value (022) subtracts write permissions for the group and others:
  - **Files:** 666 – 022 = 644 (i.e., **rw-r--r--**)
  - **Directories:** 777 – 022 = 755 (i.e., **rwxr-xr-x**)
- **Purpose of the Default (022):**
  Using a default umask of 022 ensures that:
  - The owner has read and write (and execute for directories) access.
  - Group and others have only read (and execute for directories) access.

The default of 022 is chosen to protect files from being accidentally or maliciously altered by users who are not the owner. It helps maintain a secure environment by ensuring that newly created files are not overly permissive. This default setting is usually established in global configuration files (like /etc/profile or shell configuration files) and can be customized if a different security model is needed.

Why and when you would use umask?

Umask is used to enforce a security policy by ensuring that newly created files and directories do not have overly permissive access rights.

Depending on the environment or the specific needs of users and applications, you might adjust the umask:

- More Restrictive:
  Using a umask like 077 can ensure that files and directories are accessible only to the owner (e.g., 600 for files and 700 for directories), which is useful for sensitive data.
- Less Restrictive:
  A umask of 000 would allow full read/write permissions for everyone, which might be used in a controlled, non-public environment.

Explain the file permissions with umask values of 022 and 000? Is the umask of 000 secure?

Umask 022:

- **Files:**
  Default permission of 666 minus 022 results in 644 (-rw-r--r--).
- **Directories:**
  Default permission of 777 minus 022 results in 755 (drwxr-xr-x).

Umask 000:

- **Files:**
  Default permission of 666 minus 000 results in 666 (-rw-rw-rw-).
- **Directories:**
  Default permission of 777 minus 000 results in 777 (drwxrwxrwx).

A umask of 000 is generally considered insecure on multi-user systems because it allows all users to read from and write to the files and directories you create. This can lead to accidental or malicious modifications and potential data breaches.

References

5.3 The umask. (n.d.). Retrieved from Practical UNIX & Internet Security: https://web.deu.edu.tr/doc/oreily/networking/puis/ch05_03.htm#:~:text=2%20Common%20umask%20Values,processes%20are%20descendants%20of%20init.

Umask command in Linux with examples. (2025, March 19). Retrieved from geeksforgeeks.org: https://www.geeksforgeeks.org/umask-command-in-linux-with-examples/