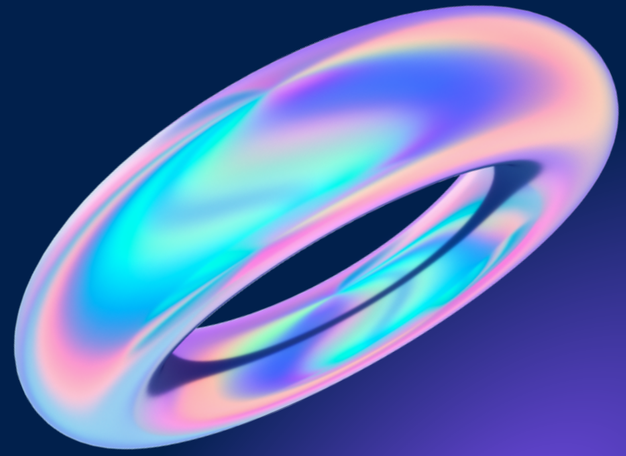
Let's get started



What are we going to learn

Linux Commands

- Learn command line
- Commands
- Syntax

Bandit

- Application of Command line
- Use of different commands

CTF

- More into Cybersecurity
- Applications of different tools

ls

```
LS(1)                          General Commands Manual                          LS(1)

NAME
     ls - list directory contents

SYNOPSIS
     ls [-@ABCFGHILOPRSTUWabcdefghiklmnopqrstuvwxy1%,] [--color=when] [-D format] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls displays its name as well as any requested, associated information.
     For each operand that names a file of type directory, ls displays the names of files contained within that directory, as well as any
     requested, associated information.

     If no operands are given, the contents of the current directory are displayed.  If more than one operand is given, non-directory operands
     are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.

     The following options are available:

     -@      Display extended attribute keys and sizes in long (-l) output.

     -A      Include directory entries whose names begin with a dot (‘.’) except for . and ..  Automatically set for the super-user unless -I is
             specified.

     -B      Force printing of non-printable characters (as defined by ctype(3) and current locale settings) in file names as \xxx, where xxx is
             the numeric value of the character in octal.  This option is not defined in IEEE Std 1003.1-2008 (“POSIX.1”).

     -C      Force multi-column output; this is the default when output is to a terminal.

     -D format
             When printing in the long (-l) format, use format to format the date and time output.  The argument format is a string used by
             strftime(3).  Depending on the choice of format string, this may result in a different number of columns in the output.  This option
             overrides the -T option.  This option is not defined in IEEE Std 1003.1-2008 (“POSIX.1”).

     -F      Display a slash (‘/’) immediately after each pathname that is a directory, an asterisk (‘*’) after each that is executable, an at
             sign (‘@’) after each symbolic link, an equals sign (‘=’) after each socket, a percent sign (‘%’) after each whiteout, and a
             vertical bar (‘|’) after each that is a FIFO.

     -G      Enable colorized output.  This option is equivalent to defining CLICOLOR or COLORTERM in the environment and setting --color=auto.
             (See below.)  This functionality can be compiled out by removing the definition of COLORLS.  This option is not defined in IEEE Std
             1003.1-2008 (“POSIX.1”).

     -H      Symbolic links on the command line are followed.  This option is assumed if none of the -F, -d, or -l options are specified.

     -I      Prevent -A from being automatically set for the super-user.  This option is not defined in IEEE Std 1003.1-2008 (“POSIX.1”).

     -L      Follow all symbolic links to final target and list the file or directory the link references rather than the link itself.  This
             option cancels the -P option.

     -O      Include the file flags in a long (-l) output.  This option is incompatible with IEEE Std 1003.1-2008 (“POSIX.1”).  See chflags(1)
             for a list of file flags and their meanings.
```

ls <options> <file/directory>`

cat

CAT(1) General Commands Manual CAT(1)

NAME

cat - concatenate and print files

SYNOPSIS

cat [**-belnstuv**] [file ...]

DESCRIPTION

The **cat** utility reads files sequentially, writing them to the standard output. The file operands are processed in command-line order. If file is a single dash (‘**-**’) or absent, **cat** reads from the standard input. If file is a UNIX domain socket, **cat** connects to it and then reads it until EOF. This complements the UNIX domain binding capability available in inetd(8).

The options are as follows:

-b Number the non-blank output lines, starting at 1.

-e Display non-printing characters (see the **-v** option), and display a dollar sign (‘$’) at the end of each line.

-l Set an exclusive advisory lock on the standard output file descriptor. This lock is set using fcntl(2) with the F_SETLKW command. If the output file is already locked, **cat** will block until the lock is acquired.

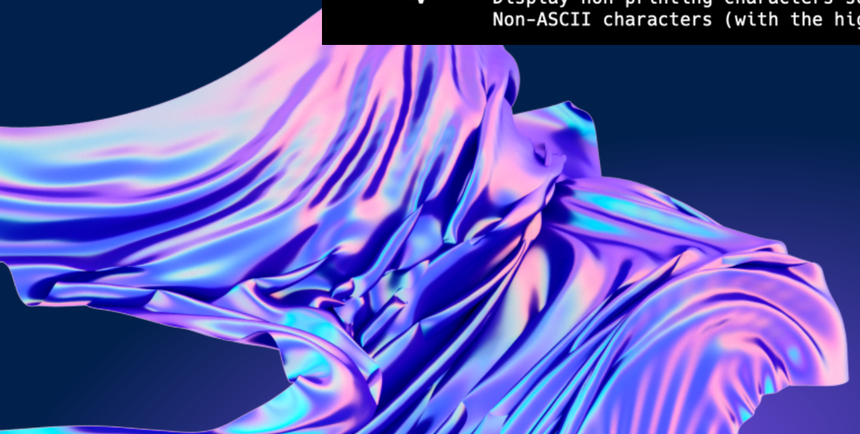
-n Number the output lines, starting at 1.

-s Squeeze multiple adjacent empty lines, causing the output to be single spaced.

-t Display non-printing characters (see the **-v** option), and display tab characters as ‘^I’.

-u Disable output buffering.

-v Display non-printing characters so they are visible. Control characters print as ‘^X’ for control-X; the delete character (octal 0177) prints as ‘^?’. Non-ASCII characters (with the high bit set) are printed as ‘M-’ (for meta) followed by the character for the low 7 bits.



cat <options> file

mkdir

```
MKDIR(1)                        General Commands Manual                        MKDIR(

NAME
     mkdir - make directories

SYNOPSIS
     mkdir [-pv] [-m mode] directory_name ...

DESCRIPTION
     The mkdir utility creates the directories named as operands, in the order specified, using mode “rwxrwxrwx” (0777) as modified by the current umask(2).

     The options are as follows:

     -m mode        Set the file permission bits of the final created directory to the specified mode.  The mode argument can be in any of the formats specified to the
                    chmod(1) command.  If a symbolic mode is specified, the operation characters ‘+’ and ‘-’ are interpreted relative to an initial mode of “a=rwx”.

     -p             Create intermediate directories as required.  If this option is not specified, the full path prefix of each operand must already exist.  On the other
                    hand, with this option specified, no error will be reported if a directory given as an operand already exists.  Intermediate directories are created
                    with permission bits of “rwxrwxrwx” (0777) as modified by the current umask, plus write and search permission for the owner.

     -v             Be verbose when creating directories, listing them as they are created.

     The user must have write permission in the parent directory.

EXIT STATUS
     The mkdir utility exits 0 on success, and >0 if an error occurs.

EXAMPLES
     Create a directory named foobar:

           $ mkdir foobar

     Create a directory named foobar and set its file mode to 700:

           $ mkdir -m 700 foobar

     Create a directory named cow/horse/monkey, creating any non-existent intermediate directories as necessary:

           $ mkdir -p cow/horse/monkey
```

mkdir <options>directory

touch

TOUCH(1) General Commands Manual TOUCH(1)

NAME

touch - change file access and modification times

SYNOPSIS

touch [**-A** [-][[hh]mm]SS] [**-achm**] [**-r** file] [**-t** [[CC]YY]MMDDhhmm[.SS]] [**-d** YYYY-MM-DDThh:mm:SS[.frac][tz]] file ...

DESCRIPTION

The **touch** utility sets the modification and access times of files. If any file does not exist, it is created with default permissions.

By default, **touch** changes both modification and access times. The **-a** and **-m** flags may be used to select the access time or the modification time individually. Selecting both is equivalent to the default. By default, the timestamps are set to the current time. The **-d** and **-t** flags explicitly specify a different time, and the **-r** flag specifies to set the times those of the specified file. The **-A** flag adjusts the values by a specified amount.

The following options are available:

-A Adjust the access and modification time stamps for the file by the specified value. This flag is intended for use in modifying files with incorrectly set time stamps.

The argument is of the form "[-][[hh]mm]SS" where each pair of letters represents the following:

- `-` Make the adjustment negative: the new time stamp is set to be before the old one.
- `hh` The number of hours, from 00 to 99.
- `mm` The number of minutes, from 00 to 59.
- `SS` The number of seconds, from 00 to 59.

The **-A** flag implies the **-c** flag: if any file specified does not exist, it will be silently ignored.

-a Change the access time of the file. The modification time of the file is not changed unless the **-m** flag is also specified.

-c Do not create the file if it does not exist. The **touch** utility does not treat this as an error. No error messages are displayed and the exit value is not affected.

-d Change the access and modification times to the specified date time instead of the current time of day. The argument is of the form "YYYY-MM-DDThh:mm:SS[.frac][tz]" where the letters represent the following:

- `YYYY` At least four decimal digits representing the year.
- `MM`, `DD`, `hh`, `mm`, `SS` As with **-t** time.
- `T` The letter T or a space is the time designator.
- `.frac` An optional fraction, consisting of a period or a comma followed by one or more digits. The number of significant digits depends on the kernel configuration and the filesystem, and may be zero.
- `tz` An optional letter Z indicating the time is in UTC. Otherwise, the time is assumed to be in local time. Local time is affected by the value of the TZ environment variable.

touch <directory/file name>

curl

curl(1) curl Manual curl(1)

NAME

curl - transfer a URL

SYNOPSIS

curl [options / URLs]

DESCRIPTION

curl is a tool for transferring data from or to a server. It supports these protocols: DICT, FILE, FTP, FTPS, GOPHER, GOPHERS, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, MQTT, POP3, POP3S, RTMP, RTMPS, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET, TFTP, WS and WSS. The command is designed to work without user interaction.

curl offers a busload of useful tricks like proxy support, user authentication, FTP upload, HTTP post, SSL connections, cookies, file transfer resume and more. As you will see below, the number of features will make your head spin.

curl is powered by libcurl for all transfer-related features. See libcurl(3) for details.

URL

The URL syntax is protocol-dependent. You find a detailed description in RFC 3986.

You can specify multiple URLs or parts of URLs by writing part sets within braces and quoting the URL as in:

"http://site.{one,two,three}.com"

or you can get sequences of alphanumeric series by using [] as in:

"ftp://ftp.example.com/file[1-100].txt"

"ftp://ftp.example.com/file[001-100].txt" (with leading zeros)

"ftp://ftp.example.com/file[a-z].txt"

Nested sequences are not supported, but you can use several ones next to each other:

"http://example.com/archive[1996-1999]/vol[1-4]/part{a,b,c}.html"

curl<options/URL>

mv

```
MV(1)                          General Commands Manual                          MV(1)

NAME
     mv - move files

SYNOPSIS
     mv [-f | -i | -n] [-hv] source target
     mv [-f | -i | -n] [-v] source ... directory

DESCRIPTION
     In its first form, the mv utility renames the file named by the source operand to the destination path named by the target operand.  This form is assumed when the last
     operand does not name an already existing directory.

     In its second form, mv moves each file named by a source operand to a destination file in the existing directory named by the directory operand.  The destination path
     for each operand is the pathname produced by the concatenation of the last operand, a slash, and the final pathname component of the named file.

     The following options are available:

     -f      Do not prompt for confirmation before overwriting the destination path.  (The -f option overrides any previous -i or -n options.)

     -h      If the target operand is a symbolic link to a directory, do not follow it.  This causes the mv utility to rename the file source to the destination path target
             rather than moving source into the directory referenced by target.

     -i      Cause mv to write a prompt to standard error before moving a file that would overwrite an existing file.  If the response from the standard input begins with
             the character 'y' or 'Y', the move is attempted.  (The -i option overrides any previous -f or -n options.)

     -n      Do not overwrite an existing file.  (The -n option overrides any previous -f or -i options.)

     -v      Cause mv to be verbose, showing files after they are moved.

     It is an error for the source operand to specify a directory if the target exists and is not a directory.

     If the destination path does not have a mode which permits writing, mv prompts the user for confirmation as specified for the -i option.

     As the rename(2) call does not work across file systems, mv uses cp(1) and rm(1) to accomplish the move.  The effect is equivalent to:

           rm -f destination_path && \
           cp -pRP source_file destination && \
           rm -rf source_file
```

mv <options> <source><destination>`

cp

```
CP(1)                          General Commands Manual                          CP(1)

NAME
     cp - copy files

SYNOPSIS
     cp [-R [-H | -L | -P]] [-fi | -n] [-alpSsvXx] source_file target_file
     cp [-R [-H | -L | -P]] [-fi | -n] [-alpSsvXx] source_file ... target_directory
     cp [-f | -i | -n] [-alPpSsvx] source_file target_file
     cp [-f | -i | -n] [-alPpSsvx] source_file ... target_directory

DESCRIPTION
     In the first synopsis form, the cp utility copies the contents of the source_file to the target_file.  In the second synopsis form, the contents of each named
     source_file is copied to the destination target_directory.  The names of the files themselves are not changed.  If cp detects an attempt to copy a file to itself, the
     copy will fail.

     The following options are available:

     -H    If the -R option is specified, symbolic links on the command line are followed.  (Symbolic links encountered in the tree traversal are not followed.)

     -L    If the -R option is specified, all symbolic links are followed.

     -P    No symbolic links are followed.  This is the default if the -R option is specified.

     -R    If source_file designates a directory, cp copies the directory and the entire subtree connected at that point.  If the source_file ends in a /, the contents of
           the directory are copied rather than the directory itself.  This option also causes symbolic links to be copied, rather than indirected through, and for cp to
           create special files rather than copying them as normal files.  Created directories have the same mode as the corresponding source directory, unmodified by the
           process' umask.

           In -R mode, cp will continue copying even if errors are detected.

           Note that cp copies hard linked files as separate files.  If you need to preserve hard links, consider using tar(1), cpio(1), or pax(1) instead.

     -a    Archive mode.  Same as -RpP options. Preserves structure and attributes of files but not directory structure.

     -c    copy files using clonefile(2).  Note that if clonefile(2) is not supported for the target filesystem, then cp will fallback to using copyfile(2) instead to
           ensure the copy still succeeds.
```

cp <option> <source><destination>

man

```
MAN(1)                                  General Commands Manual                                  MAN(1)

NAME
     man, apropos, whatis - display online manual documentation pages

SYNOPSIS
     man [-adho] [-t | -w] [-M manpath] [-P pager] [-S mansect] [-m arch[:machine]] [-p [eprtv]] [mansect] page ...

     man -f [-d] [-M manpath] [-P pager] [-S mansect] keyword ...
     whatis [-d] [-s mansect] keyword ...

     man -k [-d] [-M manpath] [-P pager] [-S mansect] keyword ...
     apropos [-d] [-s mansect] keyword ...

DESCRIPTION
     The man utility finds and displays online manual documentation pages.  If mansect is provided, man restricts the search to the specific section of the manual.

     The sections of the manual are:
          1.   General Commands Manual
          2.   System Calls Manual
          3.   Library Functions Manual
          4.   Kernel Interfaces Manual
          5.   File Formats Manual
          6.   Games Manual
          7.   Miscellaneous Information Manual
          8.   System Manager's Manual
          9.   Kernel Developer's Manual
```

tr

```
TR(1)                          General Commands Manual                          TR(1)

NAME
     tr - translate characters

SYNOPSIS
     tr [-Ccsu] string1 string2
     tr [-Ccu] -d string1
     tr [-Ccu] -s string1
     tr [-Ccu] -ds string1 string2

DESCRIPTION
     The tr utility copies the standard input to the standard output with substitution or deletion of selected characters.

     The following options are available:

     -C      Complement the set of characters in string1, that is "-C ab" includes every character except for 'a' and 'b'.

     -c      Same as -C but complement the set of values in string1.

     -d      Delete characters in string1 from the input.

     -s      Squeeze multiple occurrences of the characters listed in the last operand (either string1 or string2) in the input into a single instance of the character.
             This occurs after all deletion and translation is completed.

     -u      Guarantee that any output is unbuffered.
```

tr <-options> set1 [set2]

rm

```
RM(1)                        General Commands Manual                        RM(1)

NAME
     rm, unlink - remove directory entries

SYNOPSIS
     rm [-f | -i] [-dIRrvWx] file ...
     unlink [--] file

DESCRIPTION
     The rm utility attempts to remove the non-directory type files specified on the command line.  If the permissions of the file do not permit writing, and the standard
     input device is a terminal, the user is prompted (on the standard error output) for confirmation.

     The options are as follows:

     -d      Attempt to remove directories as well as other types of files.

     -f      Attempt to remove the files without prompting for confirmation, regardless of the file's permissions.  If the file does not exist, do not display a diagnostic
             message or modify the exit status to reflect an error.  The -f option overrides any previous -i options.

     -i      Request confirmation before attempting to remove each file, regardless of the file's permissions, or whether or not the standard input device is a terminal.
             The -i option overrides any previous -f options.

     -I      Request confirmation once if more than three files are being removed or if a directory is being recursively removed.  This is a far less intrusive option than
             -i yet provides almost the same level of protection against mistakes.

     -P      This flag has no effect.  It is kept only for backwards compatibility with 4.4BSD-Lite2.

     -R      Attempt to remove the file hierarchy rooted in each file argument.  The -R option implies the -d option.  If the -i option is specified, the user is prompted
             for confirmation before each directory's contents are processed (as well as before the attempt is made to remove the directory).  If the user does not respond
             affirmatively, the file hierarchy rooted in that directory is skipped.
```

rm: rm <–options> <file/directory>

Redirection

Redirection in Linux involves changing the flow of input and output streams between commands and files or devices. Here are some common examples to help explain the concept:

1. Standard Output (stdout) Redirection (`>`):
   - Example 1: Redirect Output to a File

```
ls -l > file_list.txt
```

This command lists the files in the current directory in long format and redirects the output to a file named `file_list.txt`. If the file exists, it will be overwritten; otherwise, a new file will be created.

- Example 2: Append Output to a File

```
echo "New content" >> existing_file.txt
```

This appends the text "New content" to an existing file named `existing_file.txt`. If the file does not exist, it will be created.

2. Standard Input (stdin) Redirection (`<`):
- Example: Read Input from a File

```
sort < unsorted_file.txt
```

This sorts the contents of the file `unsorted_file.txt` and displays the sorted output. The `<` symbol is used to redirect the input from the specified file.

3. Combining Input and Output Redirection:
- Example: Count Lines in a File and Save the Result to Another File

```
wc -l < input_file.txt > line_count.txt
```

This counts the number of lines in `input_file.txt` and redirects the output (which includes the line count) to a file named `line_count.txt`.

4. Piping Output to Another Command (`|`):
- Example: Combine Commands with a Pipe

```
cat file.txt | grep "pattern" | sort
```

This concatenates the contents of `file.txt`, filters lines containing "pattern" using `grep`, and then sorts the output. The `|` symbol is used to redirect the output of one command as input to another.

Bandit

https://overthewire.org/wargames/bandit/

What we learnt

- How useful windows is
- How to use command line
- Different important commands
- What are CTFs
- Basics on how to play CTFs
- How cool MIST is

Thank You

@sudo.mist

Sudo.mist@gmail.com

wearemist.in