Working with Files and Directories

The main difference between working with files in Linux and Windows is the way we can access the files. For example, we usually have to open Explorer to find and edit files in Windows. In Linux, on the other hand, we have a terminal where we can access and edit files using commands. Moreover, we can even edit the files interactively without using an editor, such as `vim` or `nano`.

The terminal in Linux is a more efficient and faster tool because you can access the files directly with a few commands and edit and modify them selectively with regular expressions (`regex`). You can also run several commands simultaneously and redirect the output to a file. This saves time and is very handy when we want to edit many files at once.

Create, Move, and Copy

Next, let us work with files and directories and learn how to create, rename, move, copy, and delete. First, let us create an empty file and a directory. We can use `touch` to create an empty file and `mkdir` to create a directory.

The syntax for this is the following:

Syntax - touch

Syntax - touch

```
amit8986@htb[/htb]$ touch <name>
```

Syntax - mkdir

Syntax - mkdir

```
amit8986@htb[/htb]$ mkdir <name>
```

In this example, we name the file `info.txt` and the directory `Storage`. To create these, we follow the commands and their syntax shown above.

Create an Empty File

Create an Empty File

```
amit8986@htb[/htb]$ touch info.txt
```

Create a Directory

Create a Directory

```
amit8986@htb[/htb]$ mkdir Storage
```

We may want to have specific directories in the directory, and it would be very time-consuming to create this command for every single directory. The command `mkdir` has an option marked `-p` to add parent directories.

Create a Directory

```
amit8986@htb[/htb]$ mkdir -p Storage/local/user/documents
```

We can look at the whole structure after creating the parent directories with the tool `tree`.

Create a Directory

```
amit8986@htb[/htb]$ tree .

.
├── info.txt
└── Storage
    └── local
        └── user
            └── documents

4 directories, 1 file
```

We can also create files directly in the directories by specifying the path where the file should be stored. The trick is to use the single dot (`.`) to tell the system that we want to start from the current directory. So the command for creating another empty file looks like this:

Create userinfo.txt

Create userinfo.txt

```
amit8986@htb[/htb]$ touch ./Storage/local/user/userinfo.txt
```

Create userinfo.txt

```
amit8986@htb[/htb]$ tree .

.
├── info.txt
└── Storage
    └── local
        └── user
            ├── documents
            └── userinfo.txt

4 directories, 2 files
```

With the command `mv`, we can move and also rename files and directories. The syntax for this looks like this:

Syntax - mv

Syntax - mv

```
amit8986@htb[/htb]$ mv <file/directory> <renamed file/directory>
```

First, let us rename the file `info.txt` to `information.txt` and then move it to the directory `Storage`.

Rename File

Rename File

```
amit8986@htb[/htb]$ mv info.txt information.txt
```

Now let us create a file named `readme.txt` in the current directory and then copy the files `information.txt` and `readme.txt` into the `Storage/` directory.

Create readme.txt

Create readme.txt

```
amit8986@htb[/htb]$ touch readme.txt
```

Move Files to Specific Directory

Move Files to Specific Directory

```
amit8986@htb[/htb]$ mv information.txt readme.txt Storage/
```

Move Files to Specific Directory

```
amit8986@htb[/htb]$ tree .

.
└── Storage
    ├── information.txt
    ├── local
    │   └── user
    │       ├── documents
    │       └── userinfo.txt
    └── readme.txt

4 directories, 3 files
```

Let us assume we want to have the `readme.txt` in the `local/` directory. Then we can copy them there with the paths specified.

Copy readme.txt

Copy readme.txt

```
amit8986@htb[/htb]$ cp Storage/readme.txt Storage/local/
```

Now we can check if the file is thereby using the tool `tree` again.

Copy readme.txt

```
amit8986@htb[/htb]$ tree .

.
└── Storage
    ├── information.txt
    ├── local
    │   ├── readme.txt
    │   └── user
    │       ├── documents
    │       └── userinfo.txt
    └── readme.txt

4 directories, 4 files
```

There are also many other ways to work with files using redirects or text editors, which we will see and discuss later in other sections.

? Optional Exercise:

Use the tools we already know to find out how to delete files and directories.

VPN Servers

⚠ Warning: Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.

All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.

Existing PwnBox instances will automatically switch to the new VPN server.

eu-academy-1

PROTOCOL

UDP 1337 TCP 443

DOWNLOAD VPN CONNECTION FILE

Start Instance

0 / 1 spawns left

Waiting to start...

Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

Download VPN Connection File

Target: 10.129.55.110

Life Left: 97 minutes

SSH to 10.129.55.110 with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 0 What is the name of the last modified file in the "/var/backups" directory?

Submit your answer here...

Submit

+ 1 What is the inode number of the "shadow.bak" file in the "/var/backups" directory?

Submit your answer here...

Submit

Previous Next

Cheat Sheet

Go to Questions

Table of Contents