



Conversation with:  
**Tomás Delgado**  
Service Desk Operator

# Technical Support Specialist Adv. Challenge Chat Transcript

**Tomás sent:**

Hey, @playerone! I'm glad to have you as my new intern! 😊  
I'm Tomás Delgado, I'm a Service Desk Operator here at TryCyber Manufacturing Company.

**Tomás sent:**

I'm going to be showing you how to perform some of the core tasks of a **Technical Support Specialist**. You'll be helping me complete some of the simpler tasks I take care of using the **Ubuntu 22.04 Linux** workstation you are looking at now.

**Tomás sent:**

The tasks we will be doing today will be completed using the **Terminal**. It can be intimidating if you have not used it before. So let me know if you want a brief tutorial on it.

**Tomás sent:**

Are you ready to get started? 🤖

**Participant sent:**  
Terminal tutorial, please!

**Tomás sent:**

The **Terminal** is primarily used to access an application called a shell. We interact with shells via a **command-line interface (CLI)** where we input and execute text-based commands on a system.

**Tomás sent:**

Let's open the Terminal on this system now and run a few basic commands so you get the hang of it. You can open the **Terminal** by double-clicking the icon on the desktop named **Terminal Emulator**. It should look like this once you have it open...

**Tomás sent:**

```
Terminal - playerone@workstation: ~
File Edit View Terminal Tabs Help
playerone@workstation:~$
```

**Participant sent:**  
I've opened the Terminal.

**Tomás sent:**

Great. Basically, you type commands into the shell within the Terminal and then hit **Enter** or **Return** on the keyboard to run the command.

**Tomás sent:**

To properly utilize any CLI command, you must understand its format, options, and other arguments. The **format** defines the structure and order of options and other arguments, the **options** define or modify behavior, and the **arguments** are command defined input types (e.g., files, directories, users, software packages, etc.).

**Tomás sent:**

Additionally, all CLI commands on this system are case-sensitive, so pay close attention to the capitalization of letters.

**Tomás sent:**

The basic command format is...

```
command [OPTIONS] [ARGUMENTS]
```

However, it varies a lot from command to command, and options and other arguments are not always required.

**Tomás sent:**

An example of a command that does not require any options or other arguments is `whoami`. If you type `whoami` into the shell and then hit `Enter` or `Return` on the keyboard, the `whoami` command will output the name of the user running the command into the Terminal. It should look like this if you run that command...

**Tomás sent:**

```
playerone@workstation:~$ whoami  
playerone
```

**Participant sent:**  
What's a more involved example?

**Tomás sent:**

I'll give you a more complex example using the `ls` command.

**Tomás sent:**

In this example, we will use the `ls` command to list detailed information about the entire contents of the `Templates` directory (i.e., folder) in your (playerone's) `home` directory. Our example `ls` command will use the following format...

```
ls [OPTIONS] [DIRECTORY]
```

Note that in this case, the command's argument requires the input to be a directory.

**Tomás sent:**

The actual command we want to run is...

```
ls -a -l /home/playerone/Templates
```

**Tomás sent:**

The `-a` and the `-l` are both options that modify the `ls` command's behavior, and `/home/playerone/Templates` is our argument which is a directory provided in the form of a path.

**Tomás sent:**

For this command's argument, we must provide the **path** to the directory from the root of the file system; otherwise, the system won't know which directory named `Templates` we are referring to.

**Tomás sent:**

And finally, when you run that command, the output should look like this...

**Tomás sent:**

```
playerone@workstation:~$ ls -a -l /home/playerone/Templates/
total 24
drwxr-xr-x  2 playerone playerone 4096 Jul 12 21:36 .
drwxr-x--- 14 playerone playerone 4096 Jul 12 21:36 ..
-rw-r--r--  1 playerone playerone 6859 Apr  9 2022 'OpenDocument Spreadsheet.ods'
-rw-r--r--  1 playerone playerone 7388 Apr  9 2022 'OpenDocument Text.odt'
-rw-r--r--  1 playerone playerone   0 Apr  9 2022 'Plain Text.txt'
```

**Participant sent:**  
Got it. Any extra notes?

**Tomás sent:**

The Terminal and shells are incredibly powerful and versatile tools. Not all commands and programs follow the general structure I've provided you with here. Unfortunately, we only have time to cover the basics, but I do have a few last things I'd like to mention.

**Tomás sent:**

Some commands will not print visual output to the Terminal in normal operation, such as the `cp` command, which is used to copy files and directories.

**Tomás sent:**

Many commands can use multiple arguments, handle multiple input types for arguments (e.g., file and/or directory paths), or have options that will have their own arguments.

**Tomás sent:**

Some options can, or must be, written in a long-form format (e.g., `ls --all [DIRECTORY]` is the same as `ls -a [DIRECTORY]`).

**Tomás sent:**

Options without arguments can often be provided together and in any order (e.g., `ls -la [DIRECTORY]` is the same as `ls -a -l [DIRECTORY]`).

**Tomás sent:**

And last, but certainly not least, you can almost always reference a command's format, options, and other arguments using the command `man [COMMAND]` (e.g., `man cp`) to view the provided command's manual page in the Terminal.

**Tomás sent:**

Hopefully that was not too much information! I know it seems like a lot, but it gets easier the more you use it. For today's tasks, I'll be sure to provide you with more details for any commands and programs you'll need.

**Participant sent:**  
Sounds good! I'm ready to get started!

**Tomás sent:**

Awesome!

**Tomás sent:**

For today's tasks, we're going to be resolving two very similar support tickets, each of which is from employees who recently returned from an extended leave and are requesting their system user accounts be re-activated on this Ubuntu Linux workstation.

**Tomás sent:**

Their system accounts were de-activated since they were on an extended leave. This is good security hygiene and company policy here at TryCyber Manufacturing Company. 😊

**Tomás sent:**

Resolving support tickets, assisting users with basic technical issues, and administering user accounts are all common duties for a Technical Support Specialist! That is why I'm having you help me with these two tasks today!

**Tomás sent:**

However, since you are only an intern right now, you don't have access to the support ticket system. So, I will just relay the ticket details to you here.

**Participant sent:**  
Okay! What's the first ticket?

**Tomás sent:**

The first support ticket, and your first task, is from James Jones, whose user account is `jjones`. Their ticket states they have just returned from a long vacation and are **requesting that their user account be re-activated**.

**Tomás sent:**

I already went ahead and confirmed this with their manager, so we are clear to re-activate the `jjones` user account. 👍

**Tomás sent:**

But, before we do that, let me quickly explain how we de-activate/re-activate accounts here, as there are multiple ways to do so. Basically, we **expire and un-expire accounts**, as that is the more complete way to de-activate an account on this Ubuntu Linux system.

**Tomás sent:**

To de-activate an account, we use a `usermod` command to set the date the accounts expires to a date far in the past. Then, and more importantly, since this is what you will be doing today, to **re-activate an account**, we use a `usermod` command to set the date the accounts expire to `never`.

**Tomás sent:**

To show you what I'm talking about and so we can verify that the `jjones` account is still expired and de-activated, run the following `chage` command in the Terminal to display account aging information for the `jjones` account...

```
sudo chage -l jjones
```

The above command utilizes the `sudo` command, which will sometimes prompt you for your password. Just enter your password into the prompt and hit `Enter` or `Return` on the keyboard when you're done. (Note: Your password can be found on the Info Tab)

**Tomás sent:**

When you run the command, the end of the command output should look like this...

**Tomás sent:**

```
Password expires           : never
Password inactive         : never
Account expires           : Jan 02, 1970
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

**Participant sent:**  
Got it.

**Tomás sent:**

Based on that output, we can see that the `jjones` account has its `Account expires` setting set to `January 2, 1970`, that date far in the past I mentioned earlier. This verifies that the `jjones` account is indeed still expired and de-activated.

**Tomás sent:**

With that out of the way, why don't you try and un-expire/re-activate the `jjones` user account for me? Ideally, I'd like you to set the `Account expires` setting to `never`; however, since you are new and just learning, you can set it to any day in the future. I will just fix it up before I close the support ticket if needed.

**Tomás sent:**

While there are multiple commands you could use to un-expire a user account on the system, I'll suggest you use the `usermod` command here. The `usermod` command format for the command you will need is...

```
usermod [OPTIONS] [USERNAME]
```

**Tomás sent:**

If you are not familiar with how to use the `usermod` command to do this, you could search something like 'usermod unexpire user' using a search engine on the web (e.g., Google, Bing, DuckDuckGo) or pull up the manual page for the `usermod` utility with the command `man usermod` to help you look up what to do. I have to look up information like this almost every day!

**Tomás sent:**

If you do need to look it up on the web, be sure to do it from your computer and not the workstation. This workstation is not connected to the internet. 🚫

**Tomás sent:**

Additionally, since the command involves administrative changes to the system, you must run the command as a superuser. To do that, prepend the command you want to run with the `sudo` command (i.e., `sudo usermod [OPTIONS] [USERNAME]`).

**Tomás sent:**

As a quick reminder, when you use the `sudo` command, you will sometimes be prompted for your password. Just enter your password into the prompt and hit `Enter` or `Return` on the keyboard when you're done. (Note: Your password can be found on the Info Tab)

**Tomás sent:**

You've got this! Let me know when you are done or if you need any help.

**Participant sent:**  
Help, please!

**Tomás sent:**

No worries!

**Tomás sent:**

The command you need to run in the Terminal to un-expire and re-activate the `jjones` user account is...

```
sudo usermod -e "" jjones
```

Note that the two characters between the end of the `-e` option and the beginning of the `jjones` argument are two double-quote characters (i.e., `"`) and not four single-quote characters.

**Tomás sent:**

You must start with the `sudo` command, as normal user accounts cannot make these changes to the system at their typical permission level.

**Tomás sent:**

The `-e` option of the `usermod` command indicates we want to set the expire date for a user account, which is what sets the `Account expires` settings for a user account. Then, we need to provide the `-e` option with a date argument of `"`, which leaves the option argument empty and indicates we want the account to expire `never`.

**Tomás sent:**

Finally, we need to provide the user account we want to perform the modifications to as the last argument, which in this case is `jjones`.

**Tomás sent:**

Once you've un-expired and re-activated that account, let me know, and we can move on!

**Participant sent:**

I'm done.

**Tomás sent:**

Great!

**Tomás sent:**

The second support ticket, and your last task for the day, is from Debera Smith, whose user account is `dsmith`. Their ticket states they have returned from an extended sick leave and are **requesting that their user account be re-activated**.

**Tomás sent:**

As I did with the prior ticket, I confirmed this with their manager.

**Tomás sent:**

You might be thinking this ticket and task will require you to **un-expire** the `dsmith` user account **to re-activate it**, just like you did for the `jjones` user account just a minute ago.

**Tomás sent:**

And you'd be right! Which is why I'm going to let you handle this last task on your own. I've got a few high-priority tickets I must take care of right away. 😊

**Tomás sent:**

Remember, you can use that `chage` command I showed you earlier to see if you've un-expired a user account and re-activated it. In this case, the `chage` command would be...

```
sudo chage -l dsmith
```

**Tomás sent:**

When you've un-expired and re-activated the `dsmith` account correctly, the end of the `chage` commands output should look like this...

**Tomás sent:**

```
Password expires : never
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

**Tomás sent:**  
You've got this!

**Tomás sent:**  
I will check your work and close out the support tickets at the end of the day. Thanks for all your help! 😊

TRY  
CYBER

© 2023 Try Cyber - Sponsored by CISA