Conversation with:

Tomás Delgado
Service Desk Operator

Technical Support 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 core **Technical Support** tasks. 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.

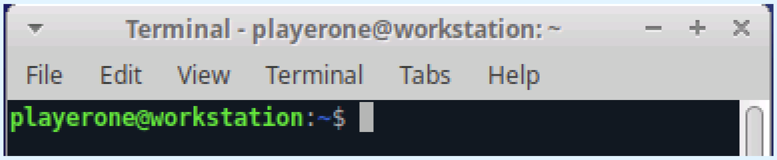
Tomás sent:
Are you ready to get started? 😁

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:



```
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.

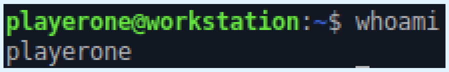
Tomás sent:
The basic command format is...

```
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...

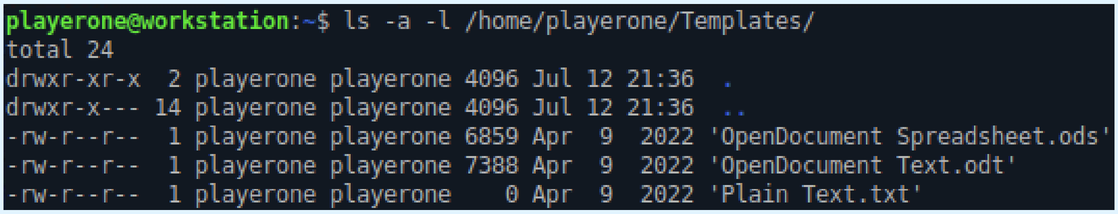
```
ls -a -l /home/playerone/Templates
```

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:
As it turns out, you're not the only new employee to join the company recently! As is typical when that happens, I have received a few new account requests via our internal ticketing system.

Tomás sent:
Provisioning basic new accounts on systems is a rather common Technical Support task, so I'll be having you help me resolve these new account requests today.

Tomás sent:
The first task you'll be helping me with today is **adding an account** for a new employee, `apatel`, on this Ubuntu Linux workstation. You will be adding the new account to the system; I will work with the new employee later to fully set up the account.

Tomás sent:
Adding user accounts is often done when multiple people need to access the same system. Generally, this allows users to keep their files and settings separate.

Tomás sent:
As an added benefit, with individual user accounts, we can control access to resources on a system based on what each user needs. 😎

Participant sent:
Sounds simple enough! How do I get started?

Tomás sent:
To get started, we are going to look at some of the accounts that are currently on this system. Most accounts you'll find on a Linux system come in two types, **normal user accounts** and system accounts.

Tomás sent:
Normal user accounts, also known as regular or standard accounts, are the types of accounts that we give to people who will be working on this system, which is exactly what we will be creating today!

Tomás sent:
System accounts, sometimes called service accounts, are generally used by the system and installed services to perform specific tasks, such as running a webserver or database.

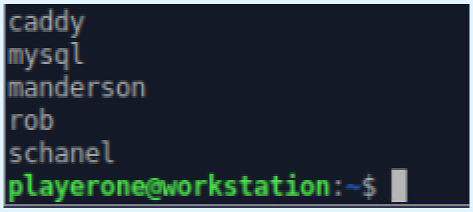
Tomás sent:
Normally, we would look at the **passwd** file to review the registered users on the system. However, that file contains a lot more information than just user account names, and I'd rather not overwhelm you on your first day trying to explain it all. 😅

Tomás sent:
So, we are going to use a `compgen` command that will give us just a list of account names without all that extra information! To see what accounts we have on this workstation, run the following command in the Terminal...

```
compgen -u
```

Tomás sent:
When you run the command, the output will probably fill up your whole terminal window! However, the last five lines of the output should look like this...

Tomás sent:



```
caddy
mysql
manderson
rob
schanel
playerone@workstation:~$ 
```

Participant sent:
Got it.

Tomás sent:
Based on that output, we've got examples of both normal and system accounts. You can see some of our co-worker's accounts like `manderson` and `rob`. Meanwhile, the `caddy` (webserver) and `mysql` (database) are system accounts named after the service/application they are associated with.

Tomás sent:
Also, based on that output, it looks like the `apatel` **user account** doesn't exist on this workstation yet. Why don't you try and add that user account for me?

Tomás sent:
While there are multiple commands you could use to add a user account to the system, I'll suggest you use the `useradd` command here. The `useradd` command format for the command you will need is...

```
useradd [USERNAME]
```

Tomás sent:
If you are not familiar with how to use the `useradd` command to do this, you could search something like 'useradd add new user' using a search engine on the web (e.g., Google, Bing, DuckDuckGo) or pull up the manual page for the `useradd` utility with the command `man useradd` 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 server. This server 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 useradd [USERNAME]`).

Tomás sent:
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 problem!

Tomás sent:
The command you need to run in the Terminal to add the `apatel` user account to the system is...

```
sudo useradd apatel
```

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:
Then have the command `useradd` followed by the username we are trying to add as the argument, which is `apatel` this time.

Tomás sent:
Once you've added that account, let me know and we can move on!

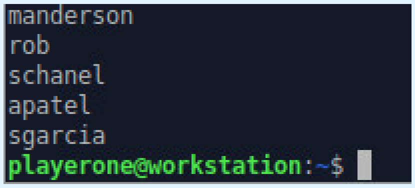
Participant sent:
I'm done.

Tomás sent:
Awesome, now let's move on to our last account request for the day. You'll basically be doing the same thing you just did again, so this should be easy! 😜

Tomás sent:
I need you to **add another user account**, `sgarcia`, to this workstation. You should be able to handle this with the `sudo` and `useradd` commands.

Tomás sent:
Oh! And remember, you can use the command `compgen -u` to see that the `apatel` and `sgarcia` user accounts have been added to the system. When you have done it correctly, the `compgen -u` command output should look like this...

Tomás sent:



```
manderson
rob
schanel
apatel
sgarcia
playerone@workstation:~$ 
```

Tomás sent:
Uh oh! I just got some high-priority tickets I need to take care of right away 😱! Looks like you're on your own for this one. You got this! 👍

© 2023 Try Cyber - Sponsored by CISA