



Conversation with:
Colt Anderson
Network Administrator

Network Operations Adv. Challenge Chat Transcript

Colt sent:

Hey @playerone, great to meet you!

Welcome to the TryCyber Foundation! I'm Colt Anderson, one of the Network Administrators that work here, and I'll be your mentor for today.

Colt sent:

I'm going to be showing you how to perform some core **Network Operations** tasks. I usually perform my tasks on switches, routers, and servers like the **Ubuntu 22.04 Linux** server you are looking at now.

Colt sent:

All the tasks you'll be performing today will require you to use the **Terminal** and a text editor. If you're unfamiliar with using a terminal, that's fine! I can bring you up to speed with a brief tutorial on it.

Colt sent:

Shall we get started? 😊

Participant sent:
Terminal tutorial, please!

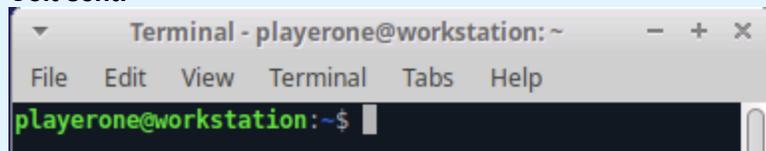
Colt sent:

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.

Colt 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...

Colt sent:



Participant sent:
I've opened the Terminal.

Colt 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.

Colt 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.).

Colt sent:

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

Colt 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.

Colt 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...

Colt sent:

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

Participant sent:

What's a more involved example?

Colt sent:

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

Colt 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.

Colt sent:

The actual command we want to run is...

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

Colt 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.

Colt 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.

Colt sent:

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

Colt 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?

Colt 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.

Colt 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.

Colt 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.

Colt 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]`).

Colt 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]`).

Colt 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.

Colt 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!

Colt sent:

Awesome!

Colt sent:

Today, we will be looking at a recently provisioned server that was apparently not integrated into the network correctly. The server in question is the Ubuntu 22.04 Linux server you are looking at now.

Colt sent:

This server is supposed to be **available on the network at three different IP addresses** but reportedly **only responds on one** of the three IP addresses it should respond on.

Colt sent:

Diagnosing and resolving network connectivity issues are core Network Operations duties!

Colt sent:

I suspect the **network interface**, sometimes called a NIC, is just misconfigured, but whatever it is, we will fix it! 🙌

Participant sent:

Good to know! Where do we start?

Colt sent:

We are going to start by checking to see **what IP addresses are configured on the server's main network interface** `ens3`, which should be configured to have all three of this server's IP addresses on it.

Colt sent:

To see what IP addresses are currently configured on `ens3`, run the following command in the Terminal...

```
ip -4 -br address show
```

Colt sent:

That `ip` command uses the `-4` option to limit the output to only include IPv4 IP addresses and the `-br` option to further limit the output to just basic details. Then the `address` object followed by the `show` option indicates we want to see the addressing information for the server's network interfaces.

Colt sent:

The output of that command should look like this...

Colt sent:

```
playerone@workstation:~$ ip -4 -br address show
lo                UNKNOWN        127.0.0.1/8
ens3              UP            192.168.100.200/24 192.168.10.230/24 192.168.100
.245/24
playerone@workstation:~$
```

Participant sent:
Got it.

Colt sent:

Alright, let's see. The **first two IP addresses this server is supposed be configured with are** `192.168.100.200/24` and `192.168.100.230/24`.

Colt sent:

The first IP address, `192.168.100.200/24`, looks right, but there appears to be a **typo in the third octet of the second IP address**. The **second IP address should be** `192.168.100.230/24`, but according to the output of our `ip` command, we see the second IP address is set to `192.168.10.230/24`.

Colt sent:

Before we move on to the third IP address, let's fix the second one. To ensure we **permanently fix this configuration**, we need to **update** the **netplan configuration file**, `01-network-manager-all.yaml` (found at the path `/etc/netplan/01-network-manager-all.yaml`) and then we need to **apply** our updated netplan to the server.

Colt sent:

To edit the `01-network-manager-all.yaml` file, we need to open it in a text editor with elevated permissions. To do that, run the following command in the Terminal...

```
sudoedit /etc/netplan/01-network-manager-all.yaml
```

That command will open the `01-network-manager-all.yaml` file at the provided path, with the default text editor (`mousepad` in this case) running as the superuser.

Colt sent:

When you use the `sudoedit` 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)

Colt sent:

You should now be looking at a graphical text editor, with a file open that looks like this...

Colt sent:

```
File Edit Search View Document Help
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens3:
      dhcp4: no
      addresses:
        - 192.168.100.200/24
        - 192.168.10.230/24
        - 192.168.100.245/24
      routes:
        - to: default
          via: 192.168.100.1
```

Participant sent:
I'm editing that file now.

Colt sent:

Awesome! You're now looking at the netplan configuration file, `01-network-manager-all.yaml`. This netplan configuration file is responsible for a whole bunch of networking configuration settings, so take great care when editing it. Additionally, since this is a YAML file, white space, indentation, and order determine how this file is read, so be careful not to alter the formatting.

Colt sent:

The section of this file we are interested in are the lines under the line `addresses:` and above the line `routes:`. These **three lines (i.e., lines 8-10) are the IP addresses that are assigned to the network interface `ens3`.**

Colt sent:

We need to **update line 9**, which is the line with the typo in it that set the second IP address to the wrong IP address.

Colt sent:

Once you find the right line in the configuration file, you need to update the line from reading...

```
- 192.168.10.230/24
to read...
- 192.168.100.230/24
```

Colt sent:

Which would look like this...

Colt sent:

```
- 192.168.100.230/24
```

Participant sent:
I've updated the line.

Colt sent:

Nice!

Colt sent:

Now, if you have not already, you will need to save the file and close the text editor; if you didn't save it before trying to close it, it will ask you if you want to save the changes. As a note, when using `sudoedit`, you must save and close the text editor before the changes are written to the open file.

Colt sent:

With the `01-network-manager-all.yaml` file updated, we now need to **apply the new netplan** to the server. To apply the updated netplan to the server, run the following command in the Terminal...

```
sudo netplan apply
```

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)

Colt sent:

If you've done everything correctly, that command should not have any output. If you see any errors, you'll likely need to re-open the `01-network-manager-all.yaml` file and review the changes you've made to make sure they are exactly as specified.

Colt sent:

To confirm your change to the network interfaces configuration, run the following command in the Terminal again...

```
ip -4 -br address show
```

Colt sent:

This time, the output should look like this...

Colt sent:

```
playerone@workstation:~$ ip -4 -br address show
lo                UNKNOWN          127.0.0.1/8
ens3              UP               192.168.100.200/24 192.168.100.230/24 192.168.10
0.245/24
playerone@workstation:~$
```

Colt sent:

Let me know once you've finished so we can review the third and last IP address configured on the `ens3` network interface.

Participant sent:
I'm done.

Colt sent:
Nice!

Colt sent:

Okay... so I guess I should have looked at what the third IP address was supposed to be earlier. We could have easily fixed the issue with the third IP address while we were fixing the second one.

Colt sent:

The issue looks to be just another typo. 😊

Colt sent:

The **third IP address is supposed to be set to 192.168.100.240/24** but, as we previously saw, it is configured as **192.168.100.245/24**.

Colt sent:

Dang! It also looks like I forgot about one of my meetings today. I've got to get going. Could you take care of this last IP address configuration fix for me?

Colt sent:

You just need to follow the same process we used to update the second IP address, but this time, you will need to edit a different line in the netplan configuration file.

Colt sent:

Basically, edit the **01-network-manager-all.yaml** netplan configuration file, fix the typo in line 10, and then apply the updated netplan to the server again using the **sudo netplan apply** command.

Colt sent:

Once you've correctly updated the netplan configuration file and re-applied the netplan on the server, you can verify your change by running the command...

```
ip -4 -br address show
```

Colt sent:

The output should look like this...

Colt sent:

```
playerone@workstation:~$ ip -4 -br address show
lo                UNKNOWN        127.0.0.1/8
ens3              UP             192.168.100.200/24 192.168.100.230/24 192.168.100.240/24
playerone@workstation:~$
```

Colt sent:

Alright, I need to get to that meeting! Thanks again for all your help! 🙏

TRY
CYBER

© 2023 Try Cyber - Sponsored by CISA