



Conversation with:

Ivy Magabo Perez
Data Security Analyst

Data Analyst Challenge

Chat Transcript

Ivy sent:

Hi @playerone! I'm Ivy, a Data Security Analyst here at TryCyber E-Buy. Welcome to the team! I'll be your mentor while you're interning here.

Ivy sent:

I'm going to be showing you how to perform the tasks of a **Data Analyst**. I usually perform my daily tasks using a workstation like the **Ubuntu 22.04 Linux** workstation you are looking at now.

Ivy sent:

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

Ivy sent:

So, are you ready to get started? 😊

Participant sent:
Terminal tutorial, please!

Ivy 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.

Ivy 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...

Ivy sent:

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

Participant sent:
I've opened the Terminal.

Ivy 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.

Ivy 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.).

Ivy sent:

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

Ivy 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.

Ivy 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...

Ivy sent:

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

Participant sent:

What's a more involved example?

Ivy sent:

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

Ivy 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.

Ivy sent:

The actual command we want to run is...

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

Ivy 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**.

Ivy 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.

Ivy sent:

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

Ivy 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?

Ivy sent:

The Terminal and shells are incredibly powerful and versatile tools, and 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.

Ivy 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.

Ivy 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.

Ivy 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]`).

Ivy 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]`).

Ivy 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.

Ivy sent:

Hopefully that was not too much! 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!

Ivy sent:

Lovely!

Ivy sent:

As you might already know, TryCyber E-buy is an online retailer, which means our most business-critical technology is our website. Our website being taken down by a *denial-of-service* attack or hacked is one of the worst things that can happen to us!

Ivy sent:

One of the tools we use to defend the web servers running our website is **Fail2ban**. Our Fail2ban configuration monitors server logs for malicious looking entries and automatically temporarily bans IP addresses associated with those malicious looking entries.

Ivy sent:

Today, I want your help **analyzing two Fail2ban logs** from last month. For each log, we need to **identify the IP address that exceeds our threshold of acceptable temporary bans** and then **add a report entry** that includes the log file name and IP address exceeding the ban threshold. The report entries will go into a report I am making, which I am later going to send to the security team so they can add the reported IP addresses to the permanent ban list.

Ivy sent:

Collecting and analyzing data for trends like this is one of the core duties of a Data Analyst!

Participant sent:

Good to know! Where do we start?

Ivy sent:

Well, we *could* complete our analysis by manually opening and reviewing the log files, but that is more error-prone and would take a while. Instead, we will use a **Python script** I wrote to analyze these ban logs and review the script's output. 😊

Ivy sent:

I've placed a copy of my Python script, `log_analyzer.py` in your `home` directory and the `Materials` folder on your desktop (at the path `/home/playerone/Desktop/Materials`). The two log files we will be working with today, `first_week.log` and `second_week.log`, are also in the `Materials` folder on your desktop.

Ivy sent:

My `log_analyzer.py` Python script essentially goes through log files formatted like `first_week.log` and `second_week.log` and counts how often it sees a line with a ban entry for each IP address in the log. Once it is done, it will output the results to the Terminal in ascending order.

Ivy sent:

Feel free to open the `Materials` folder and review my script and the logs. Just be sure not to alter anything; otherwise, you might adversely affect our analysis.

Ivy sent:

Let me know when you're ready to analyze the first log file and add our first report entry.

Participant sent:
I'm ready!

Ivy sent:

Wonderful!

Ivy sent:

To start, let me show you how to interact with a Python script in the Terminal.

Ivy sent:

While there is more than one way to run a Python script in the Terminal, I'll suggest you use the `python3` command here. Use the following `python3` command to run the `log_analyzer.py` script in your home directory with the `-h` option...

```
python3 log_analyzer.py -h
```

Ivy sent:

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

Ivy sent:

```
playerone@workstation:~$ python3 log_analyzer.py -h
usage: log_analyzer.py [-h] logfile

Analyze IP ban logs to identify the most frequently banned IP addresses in a
log.

positional arguments:
  logfile      The path to the ban log file

options:
  -h, --help  show this help message and exit
playerone@workstation:~$
```

Ivy sent:

That `python3` command will run the script with its `-h` option, which will display the script's help text. Help text is kind of like a short version of a command's manual page. It explains the formatting, options, and arguments associated with the script.

Ivy sent:

Once you've run the command, take a second to review the `log_analyzer.py` script's help text.

Participant sent:

I've reviewed it.

Ivy sent:

Now that you've seen the script's help text, why don't you go ahead and use the `python3` command and `log_analyzer.py` script to analyze the first of our two log files, `first_week.log`.

Ivy sent:

When you've run the script on the log file correctly, you will get a bunch of output formatted like this...

```
[IP Address] was banned [# of bans in log] times
```

IP addresses are formatted like this `192.168.1.12`.

Ivy sent:

We are looking for the IP Address that exceeds 100 ban entries, which is the threshold an IP address has before it gets added to the permanent ban list. Once you've identified the IP address that exceeds the threshold in the `first_week.log` log, you will need to add an entry to our report to the security team.

Ivy sent:

To **add an entry** to our report, you must use the **Report Form**. (Note: You can find the Report Form on the Report Tab).

Ivy sent:

Let me know once you've successfully run the script on the first log, identified the IP address exceeding the threshold, and added the entry to the report with the Report Form. Or, let me know if you need some extra help!

Participant sent:

Help, please!

Ivy sent:

No problem!

Ivy sent:

The command you need to run in the Terminal to analyze the `first_week.log` log with the `log_analyzer.py` script is...

```
python3 log_analyzer.py /home/playerone/Desktop/Materials/first_week.log
```

Ivy sent:

The last line of output you should get from running that command is the following...

```
10.167.132.6 was banned 398 times
```

Ivy sent:

This means the IP address `10.167.132.6` has clearly exceeded our ban threshold of 100 and needs to be reported so it can be added to the permanent ban list.

Ivy sent:

So, for the `first_week.log` log's entry in our report, you need to input the following via the Report Form...

Which log file is this report entry for? `first_week.log`

What IP address in the log file exceeded the ban threshold (i.e., 100 bans)? `10.167.132.6`

Ivy sent:

Let me know once you've successfully added the entry for the `first_week.log` log to the report.

Participant sent:
I'm done.

Ivy sent:

Excellent! ★

Ivy sent:

Now it looks like I, unfortunately, won't be available to complete the analysis and report entry for the second and last of our log files with you. I've been called into an urgent meeting. 😞

Ivy sent:

I'm sure you can handle this last one alone, though! You just need to analyze the `second_week.log` log with the `log_analyzer.py` script and add a report entry for the `second_week.log` log via the **Report Form**.

Ivy sent:

To do that, you basically just need to redo the steps you just did for `first_week.log` for `second_week.log`.

Ivy sent:

Okay, I need to get to that meeting! Thanks for all your help today! Good luck!

TRY
CYBER

© 2023 Try Cyber - Sponsored by CISA

TRY
CYBER