Activity: Examine input and output in the shell

**Activity overview**

Previously, you discussed how the Bash shell helps you communicate with a computer's operating system.

When you communicate with the shell, the commands in the shell can take input and return output or error messages.

In this lab activity, you'll use the echo command to examine how input is received and how output is returned in the shell. Next, you'll use the expr command to further explore input and output while performing some basic calculations in the shell.

This activity will build foundations in understanding how you communicate with the Linux operating system through the shell. As a security analyst, you'll need to input commands into the shell and recognize when the shell returns either output or an error message.

Next, you'll explore the scenario!

**Scenario**

As a security professional, it's important to understand the concept of communicating with your computer via the shell.

In this scenario, you have to input a specified string of text that you want the shell to return as output. You'll also need to input a few mathematical calculations so the OS (operating system) can return the result.

Here's how you'll do this: **First**, you'll use the echo command to generate some output in the shell. **Second**, you'll use the expr command to perform basic mathematical calculations. **Next**, you'll use the clear command to clear the Bash shell window. **Finally**, you'll have an opportunity to explore the echo and expr commands further.

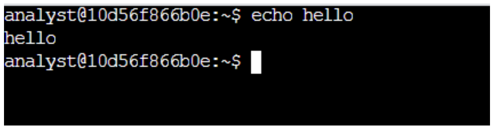
Get ready to examine input and output in the Bash shell!

**Task 1. Generate output with the echo command**

The echo command in the Bash shell outputs a specified string of text. In this task, you'll use the echo command to generate output in the Bash shell.

1. Type echo hello into the shell and press **ENTER**.

The hello string should be returned:

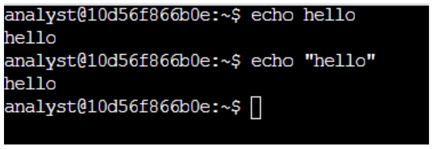


```
analyst@10d56f866b0e:~$ echo hello
hello
analyst@10d56f866b0e:~$ 
```

The command echo hello is the **input** to the shell, and hello is the **output** from the shell.

2. Rerun the command, but include quotation marks around the string data. Type echo "hello" into the shell and press **ENTER**.

The hello string should be returned again:

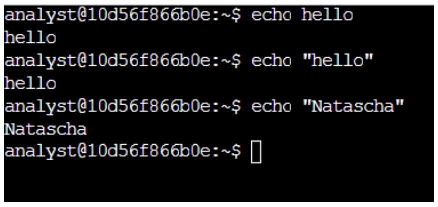


```
analyst@10d56f866b0e:~$ echo hello
hello
analyst@10d56f866b0e:~$ echo "hello"
hello
analyst@10d56f866b0e:~$ 
```

3. Use the echo command to output your name to the shell.

Type echo "name" into the shell, replacing "name" with your own name, and press **ENTER**.

The name you've entered as the string should return as the output.



```
analyst@10d56f866b0e:~$ echo hello
hello
analyst@10d56f866b0e:~$ echo "hello"
hello
analyst@10d56f866b0e:~$ echo "Natascha"
Natascha
analyst@10d56f866b0e:~$ 
```

**Task 2. Generate output with the expr command**

In this task, you'll use the expr command to generate some additional output in the Bash shell. The expr command performs basic mathematical calculations and can be useful when you need to quickly perform a calculation.

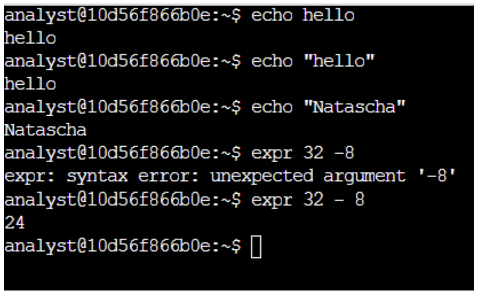
Imagine that the system has shown you that you have 32 alerts, but only 8 required action. You want to calculate how many alerts are false positives so that you can provide feedback to the team that configures the alerts.

To do this, you need to subtract the number of alerts that required action from the total number of alerts.

1. Calculate the number of false positives using the expr command.

Type expr 32 - 8 into the shell and press **ENTER**.

The following result should be returned:



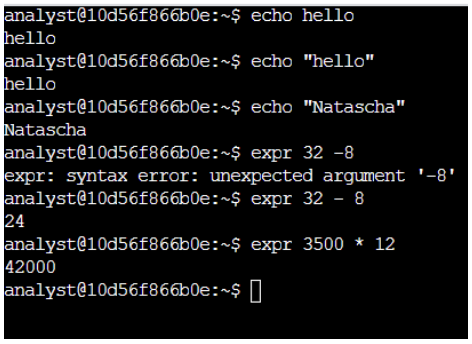
```
analyst@10d56f866b0e:~$ echo hello
hello
analyst@10d56f866b0e:~$ echo "hello"
hello
analyst@10d56f866b0e:~$ echo "Natascha"
Natascha
analyst@10d56f866b0e:~$ expr 32 -8
expr: syntax error: unexpected argument '-8'
analyst@10d56f866b0e:~$ expr 32 - 8
24
analyst@10d56f866b0e:~$ 
```

**Now**, you need to calculate the average number of login attempts that are expected over the course of a year. From the information you have, you know that an average of 3500 login attempts have been made each month so far this year.

So, you should be able to calculate the total number of logins expected in a year by multiplying **3500** by **12**.

2. Type expr 3500 * 12 into the shell and press **ENTER**.

The correct result should now be returned:



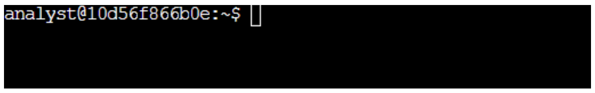
```
analyst@10d56f866b0e:~$ echo hello
hello
analyst@10d56f866b0e:~$ echo "hello"
hello
analyst@10d56f866b0e:~$ echo "Natascha"
Natascha
analyst@10d56f866b0e:~$ expr 32 -8
expr: syntax error: unexpected argument '-8'
analyst@10d56f866b0e:~$ expr 32 - 8
24
analyst@10d56f866b0e:~$ expr 3500 * 12
42000
analyst@10d56f866b0e:~$ 
```

**Task 3. Clear the Bash shell**

In this task, you'll use the clear command to clear the Bash shell of all existing output. This allows you to start with the cursor at the top of the Bash shell window.

When you work in a shell environment, the screen can fill with previous input and output data. This can make it difficult to process what you're working on. Clearing the screen allows you to create a clutter-free text environment to allow you to focus on what is important at that point in time.

- Type clear into the shell and press **ENTER**.



```
analyst@10d56f866b0e:~$ 
```

**Optional task: Perform more calculations with the expr command**

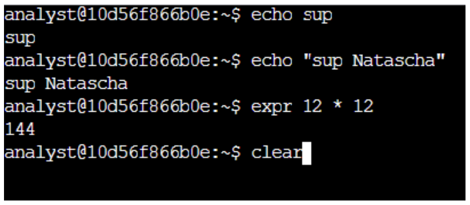
You have the opportunity to explore input and output further using
the echo and expr commands.

1. Generate at least one new output using the echo command.

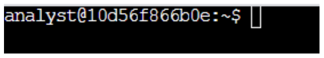
(Remember the echo "hello" output you generated).

2. Perform at least one new calculation using the expr command.

The mathematical operators you can use with the expr command for **adding, subtracting, dividing,** and **multiplying** are +, -, / and *.



```
analyst@10d56f866b0e:~$ echo sup
sup
analyst@10d56f866b0e:~$ echo "sup Natascha"
sup Natascha
analyst@10d56f866b0e:~$ expr 12 * 12
144
analyst@10d56f866b0e:~$ clear
```



```
analyst@10d56f866b0e:~$
```

**Lab Summary: Examine Input and Output in the Shell**

**Course:** Tools of the Trade: Linux and SQL
**Lab Environment:** Debian-based Linux VM (Bash shell)
**Tools Used:** echo, expr, clear

**Objective**

This lab focused on understanding how input is entered and output is returned in the Linux Bash shell. It introduced the use of basic shell commands that display text, perform calculations, and manage the shell environment. These are foundational skills for communicating with the operating system as a security analyst.

**Tasks Completed**

**Task 1: Generate Output with the echo Command**

- Ran echo hello and echo "hello" to display plain text in the terminal.
- Used echo "name" to display a string with a personal name as output.
- Demonstrated how input entered at the command prompt is returned as output.

**Task 2: Perform Calculations with the expr Command**

- Used expr 32 - 8 to calculate the number of false positives from alert data.
- Used expr 3500 * 12 to estimate yearly login attempts based on monthly averages. (Note: The asterisk must be escaped with a backslash in many shells.)

- Confirmed that the shell can process basic arithmetic using expr.

**Task 3: Clear the Bash Shell**

- Used the clear command to wipe the terminal screen and reset the view for continued input.

**Optional Exploration**

- Ran additional echo and expr commands to test string display and arithmetic.
- Practiced using other mathematical operators: +, -, /, and *.

**Summary**

This lab demonstrated how to interact with the Linux Bash shell by inputting text and basic arithmetic operations. Understanding how commands produce output, and recognizing errors when commands are misused, is an essential part of managing systems and troubleshooting as a cybersecurity professional. These skills form the basis for more advanced command-line operations.