Unpacking Linux Commands Using Al Tools

Assignment

1h



Reflect

In this activity, we will be working with the Linux System.

Consider your experience working with Linux in your past Al activities. You created Python scripts and performed workflow monitoring tools project to extract log file information.



Can you think of how these experiences will be relevant in the next lecture?

Did you have to do any troubleshooting when writing your script, and how did you approach that?

Unpacking Linux Commands

You will be given a block of Linux code below that includes a command and potential error messages.



Task #1: Your first task is to analyze this code on your own.

- Try to understand what each section of the command is doing.
- Identify any errors or issues in the code.
- Write down what you think the error messages mean and how you might resolve them.

sudo openssl enc -aes-256-cbc -salt -in file.txt -out encrypted.txt -k Encrypt123*

The command we've provided is a Linux command that uses the OpenSSL tool to encrypt the given file using our specified encryption algorithm.



Pay attention to how the AI interprets the commands and error messages.



Task #3: Compare the Al's analysis with your own. Note any differences in understanding the commands and resolving the errors.



Time to check your understanding:

There are two potential error messages when executed - what are they?

Remember you can use AI Tools to help you understand what the error messages mean, too!

- 1. Can't open "file.txt" for reading, No such file or directory 806B920ED77F0000:error:80000002:system library:BIO_new_file:No such file or directory:../crypto/bio/bss_file.c:67:calling fopen(file.txt, rb) 806B920ED77F0000:error:10000080:BIO routines:BIO_new_file:no such file:../crypto/bio/bss_file.c:75:
- 2. *** WARNING: deprecated key derivation used. Using -iter or -pbkdf2 would be better.

Toggle Answer

What do those error messages mean? Why are they happening?

Let's breakdown each command in this line of code:

\$ sudo: This part of the command runs the following command with superuser privileges. The sudo command allows you to execute commands with elevated permissions, typically requiring the user's password.

openss1: This is the command for the OpenSSL tool, which is a widely used open-source software library for Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. In this context, you are using OpenSSL to encrypt an individual file.

Enc: This is used to specify that we are using openssl for encryption.

aes-256-cbc: This specifies the encryption algorithm and version that we are using for this process.

- -Salt: This adds a salt to the encryption, which is a randomely generated piece of information which helps to make the encrypted data (ciphertext) more difficult to break via crypto attacks.
- -In file.txt: This specifies the file that we want to encrypt.
- -out encrypted.txt: This specifies the name of the file that we want to outure our encrypted information, if no file exists with this name in your directory, one will be created.
- -k Encrypt123*: This is a key/password that will be required for decrypting this file, in this case our key is Encrypt123*



How would you summarize, in one sentence, what this line of code does knowing what you now know about each command?

Hint: You can use ChatGPT to help you refine your sentence and make it more accurate!

Your Answer

you type. Once you click Toggle Answer below, your answer cannot be changed.

Toggle Answer

Post-Task Reflection

Reflect on your experience of unpacking the code yourself versus using Al.



Did the AI provide additional insights or corrections that you missed?

Consider how you interacted with the AI tool and think about the prompts you used.



Were they effective? How could you refine your prompts or questions in the future to get more precise or useful responses from the AI?

Reflect on where you might find this approach of using AI for code analysis useful.

Consider both the remainder of this program and your future professional/educational journeys.

Bash Scripting

In this section, we move from debugging code to writing code. So, it's time we discussed something called bash scripting.

In the Linux command line, all commands are written in a language called bash. Like Python, bash is a scripting language, but it is used natively in the Linux command line.



We can package Linux commands in a bash script so that they can be re-run just like a Python script can be reused.

To create a bash script, you only need three things:

- 1. Create a file with the .sh file extension.
- 2. Add the shebang at the top of the file #!/bin/bash.
- 3. Give the file permissions to execute with a command like chmod 777 filename.sh.

In the next exercise, you will create a Bash script containing Linux commands to accomplish your specific goals.

Writing Linux Commands

Now, let's dive into a hands-on exercise where you'll create and analyze Linux commands. This exercise is designed to enhance your practical understanding of Linux and to explore the role of AI in coding and command review.



Create a bash script that contains a set of Linux commands that perform any of these tasks.

These tasks can range from:

- file management
- system monitoring
- network configuration



Think about real-world scenarios where these commands could be useful. This will help make your learning more practical and relevant.

Once you have written your script...



Use an AI tool to review your script.

Input your individual commands into the AI system and analyze its feedback.

Post-Task Reflection

Compare the AI's feedback with your initial understanding and decisions.



Did you find the AI review helpful or necessary?

Where might you find using AI for command review helpful in the future?

Think about how you interacted with the AI tool.



How could you refine your prompts or questions to the AI for better, more relevant responses?





Previous **Differences Between SSL and TLS**

How well did this activity help you to understand the content?

Let us know how we're doing













Tue Aug 6

> Outline & Notes (1)

- Lectures (1)
 Work (6)
 5 hrs
 Case Studies: Encyption & Data Breach
 Cryptography Recap
 - Cryptographics Algorithms
 - ? Common Encryption Methods Quiz
 - Differences Between SSL and TLS
- Unpacking Linux Commands Using AI Tools

W07D2 Schedule »

Powered by <u>Lighthouse Labs</u>.