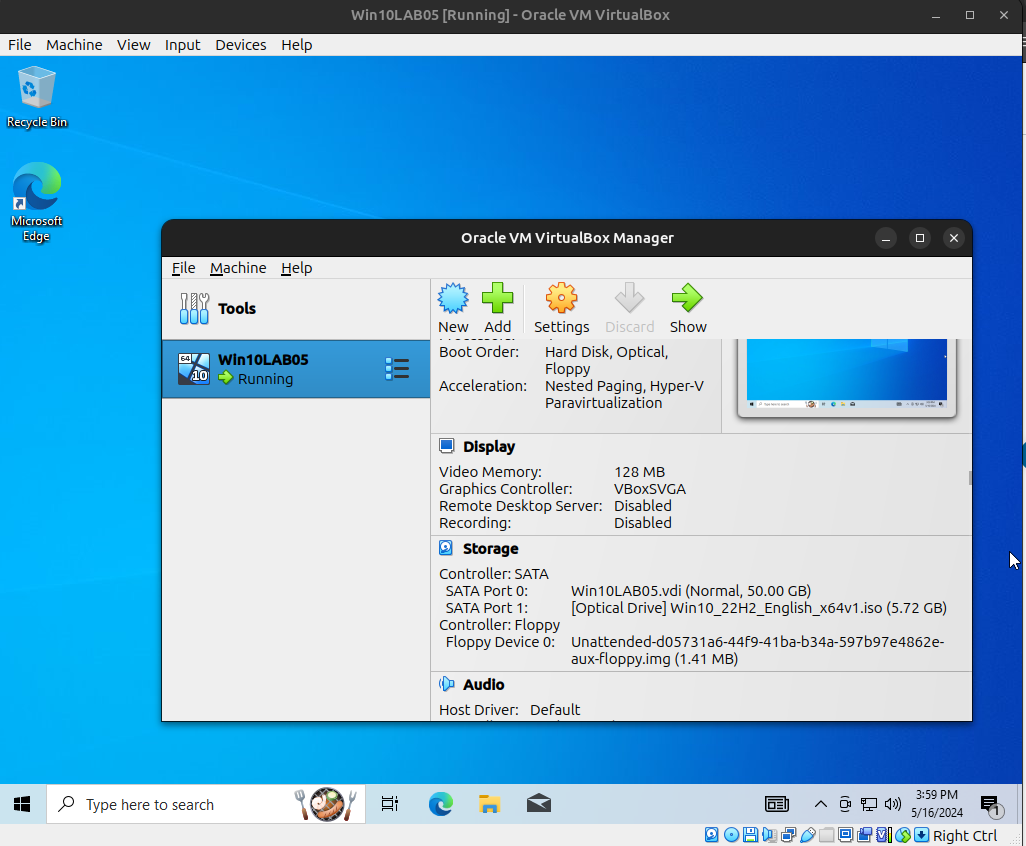
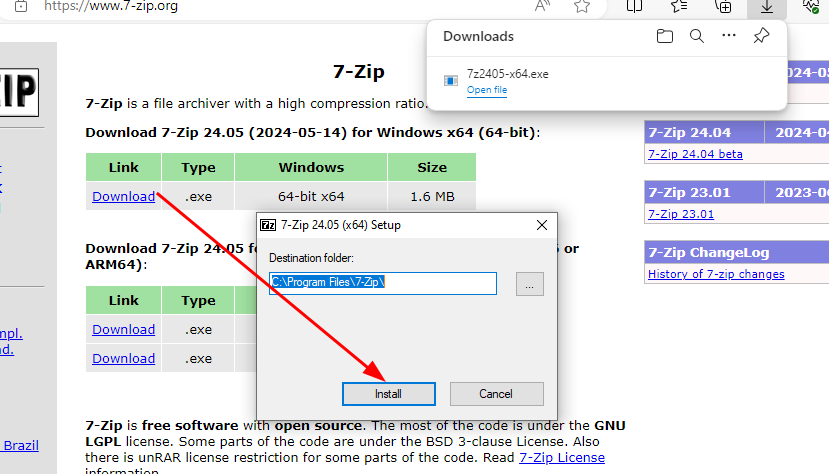
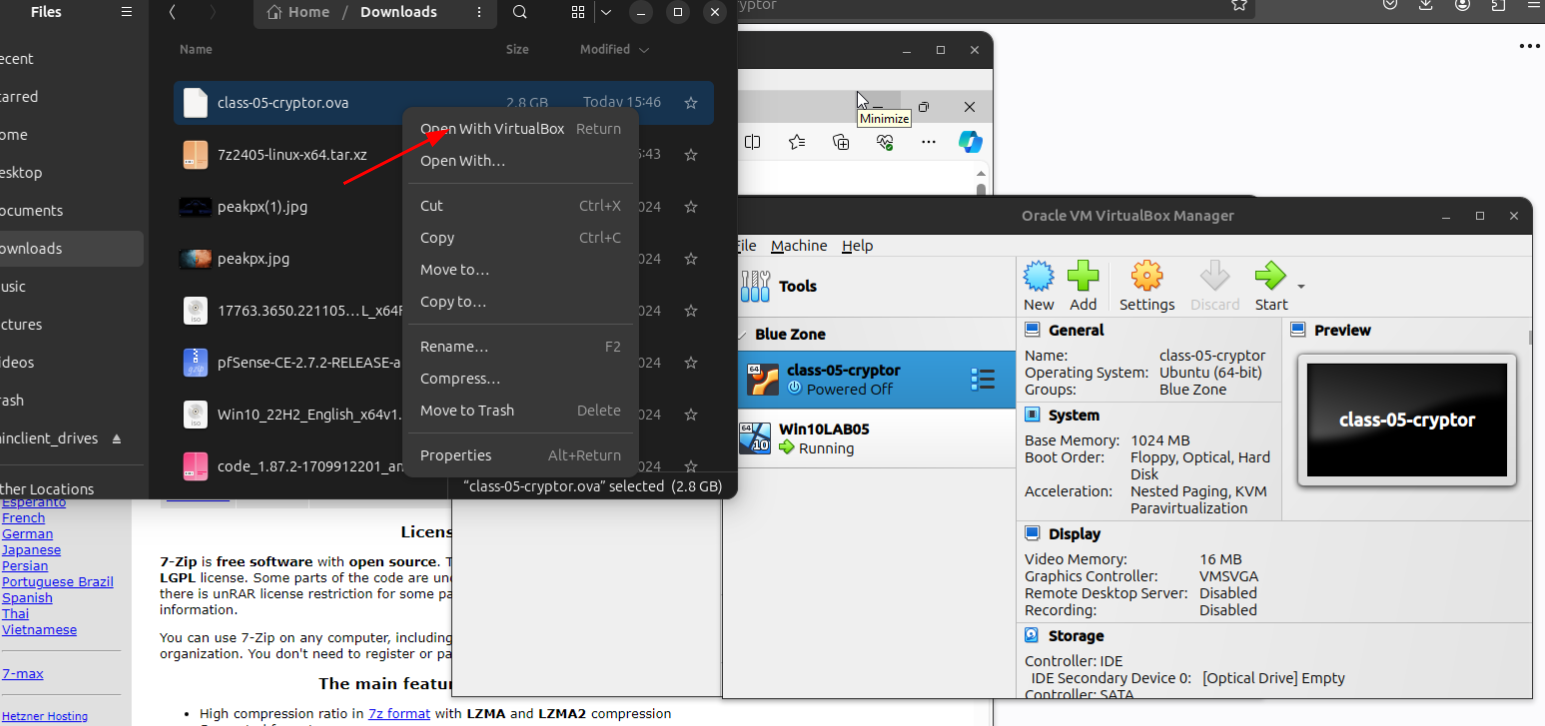
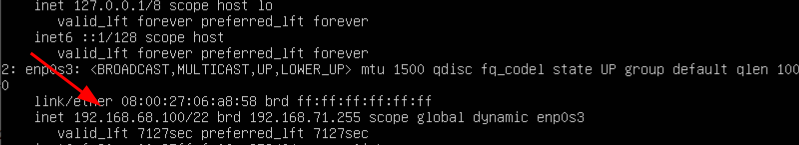
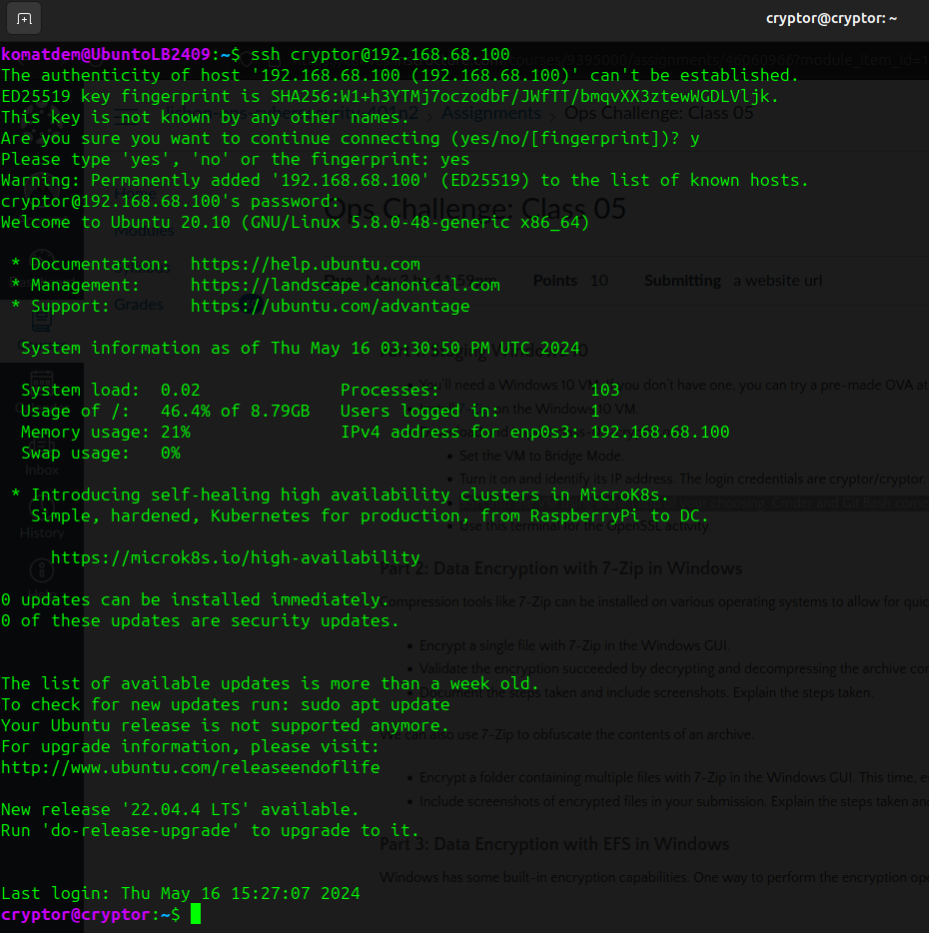
Hélio Ferreira 13/05/2024

# **File Encryption**

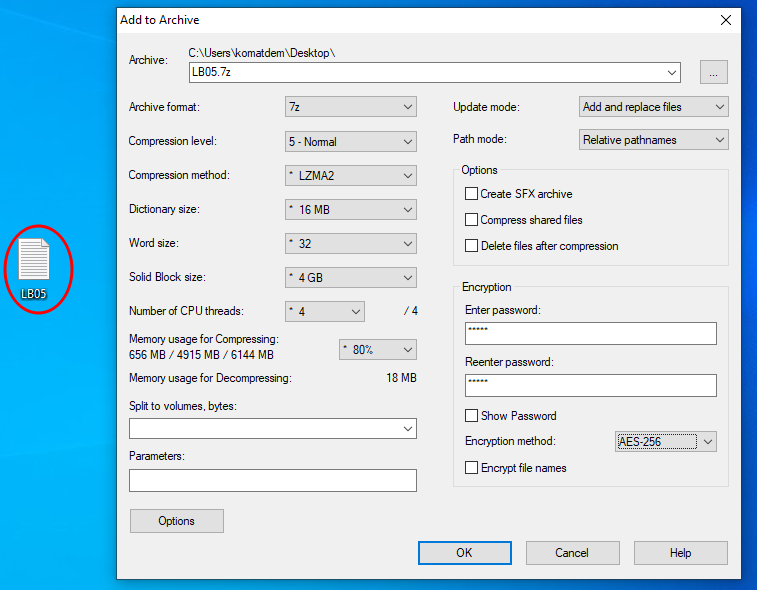
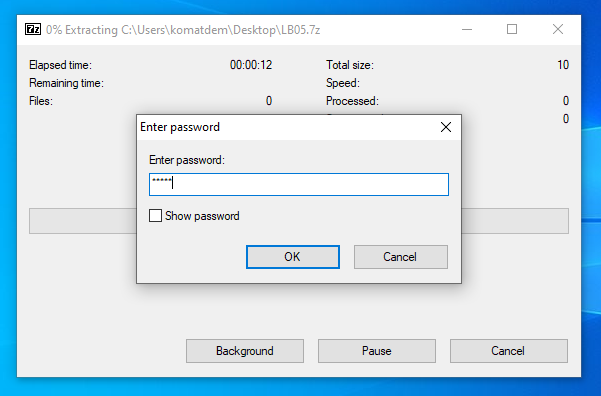
## **Tasks**

### **Part 1: Staging Windows 10**

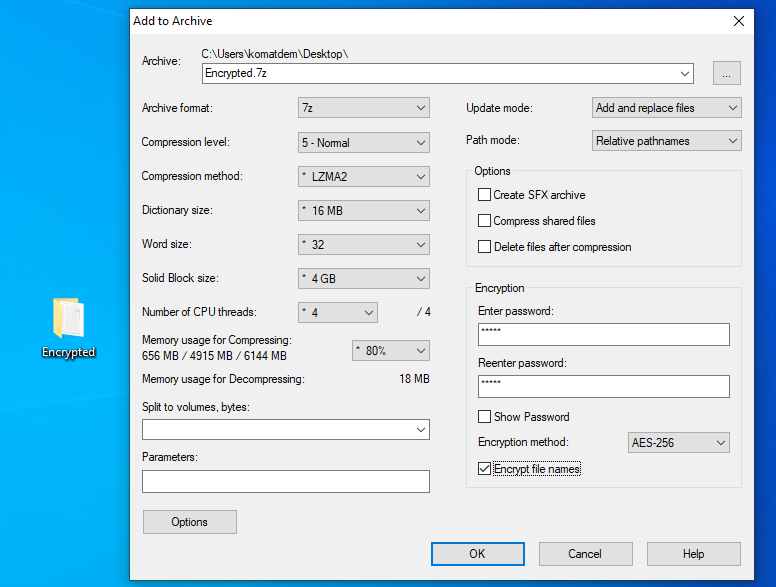
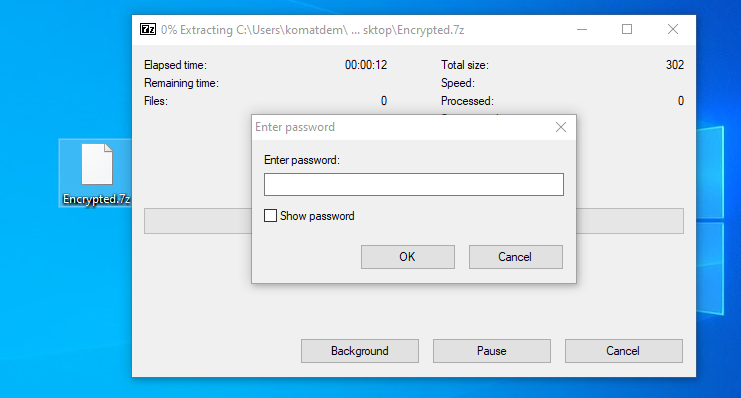
* You’ll need a Windows 10 VM. If you don’t have one, you can try a pre-made OVA at [Windows 10 VM](https://codefellows.github.io/ops-401-cybersecurity-guide/curriculum/#downloads-table) or load a fresh one from ISO.
* Install [7-Zip](https://www.7-zip.org/) on the Windows 10 VM.
* Download and import class-05-cryptor.ova.
  + Set the VM to Bridge Mode.  
    By default of the .OVA file, it has already “Bridged”
  + Turn it on and identify its IP address. The login credentials are cryptor/cryptor.
  + SSH into the VM using a shell app of your choosing. Cmder and Git Bash   
    come to mind.
  + Use this terminal for the OpenSSL activity.

### **Part 2: Data Encryption with 7-Zip in Windows**

Compression tools like 7-Zip can be installed on various operating systems to allow for quick password encryption of a file or set of files.

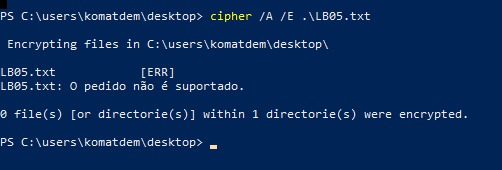
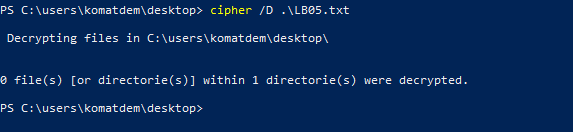
* Encrypt a single file with 7-Zip in the Windows GUI.  
    
    
  Right Mouse Button ; 7-ZIP - Add to Archive; Wrote Password and; OK
* Validate the encryption succeeded by decrypting and decompressing the archive contents to your desktop.  
    
  Right Mouse Button; 7-ZIP - Extract Here; Prompt Password

We can also use 7-Zip to obfuscate the contents of an archive.

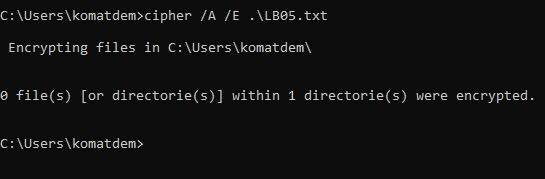
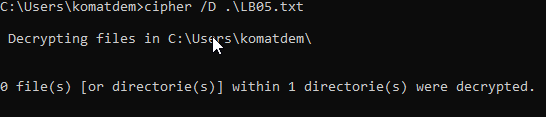
* Encrypt a folder containing multiple files with 7-Zip in the Windows GUI. This time, encrypt the file names.  
    
  
* Include screenshots of encrypted files in your submission. Explain the steps taken and what changed when you encrypted file names.  
    
  Same procedure as encrypted files.

### **Part 3: Data Encryption with EFS in Windows**

Windows has some built-in encryption capabilities. One way to perform the encryption operation is via Powershell.

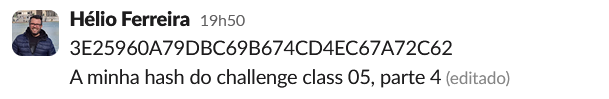
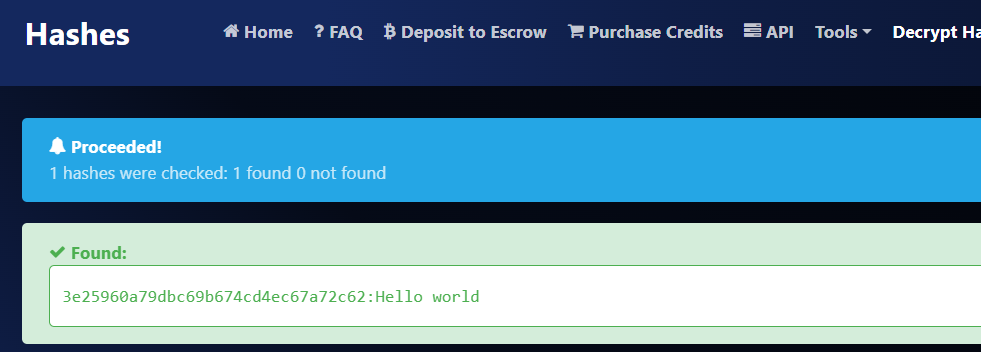
* Encrypt a single file using PowerShell in Windows 10. Include a screenshot and the line of code you used.
* Decrypt a the file using PowerShell in Windows 10. Include a screenshot and the line of code you used.

In Windows you can also perform encryption directly from the command line interface.

* Encrypt a single file in Windows 10 CLI.  
  
* Decrypt that same file in Windows 10 CLI.  
  

### **Part 4: Secret Messages with PowerShell Hash Generation**

Determining the source data from a hash is called “cracking” and is a different operation than decryption. Here’s [why](https://www.techsolvency.com/passwords/dehashing-reversing-decrypting/).

* Use PowerShell to generate the hash value of a single word or simple sentence in MD5 algorithm. Keep your original string simple and code-of-conduct friendly please! An example would be “hello” as your string.
* Post your MD5 hash value to Slack class channel for others to crack using rainbow table techniques.  
  I’ve shared it on our class Slack.
* Once a classmate has posted a MD5 hash value, crack it using a rainbow table and add it to your submission alongside the original hash value.  
    
  
* Is MD5 considered secure? Why/why not?  
    
  No, MD5 is not considered secure for cryptographic purposes. It has known vulnerabilities that make it unreliable for tasks like:

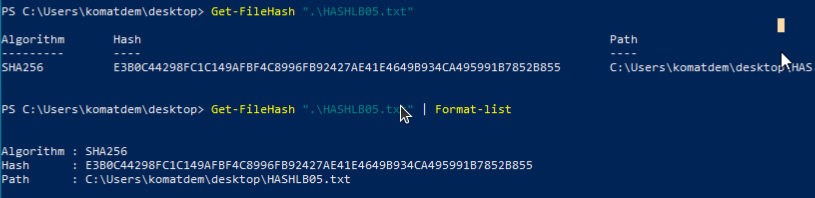
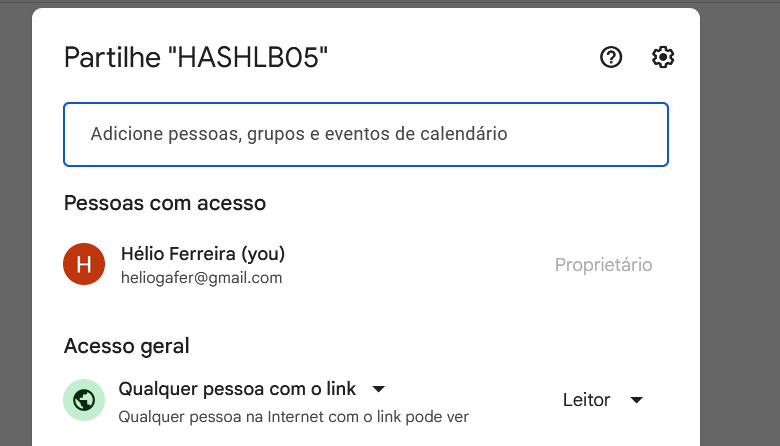
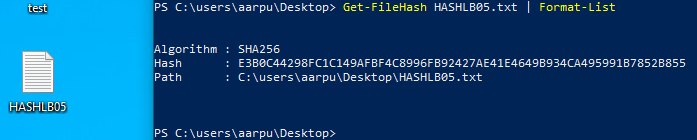
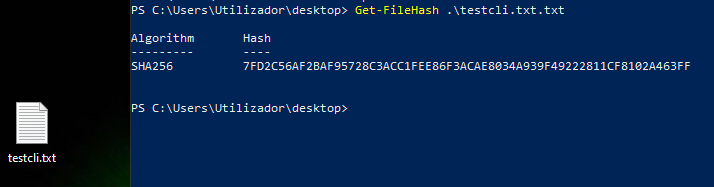
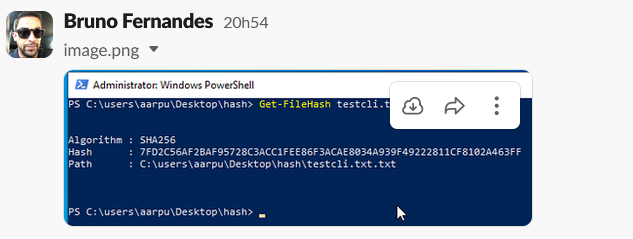
**- Storing passwords:** An attacker could potentially create a malicious file with the same MD5 hash as a legitimate one.

**-Verifying file integrity:** Collisions (different files with the same hash) are relatively easy to generate for MD5.

There are more secure hashing algorithms available, such as SHA-256 or SHA-3. These use longer hash lengths and are more resistant to attacks.

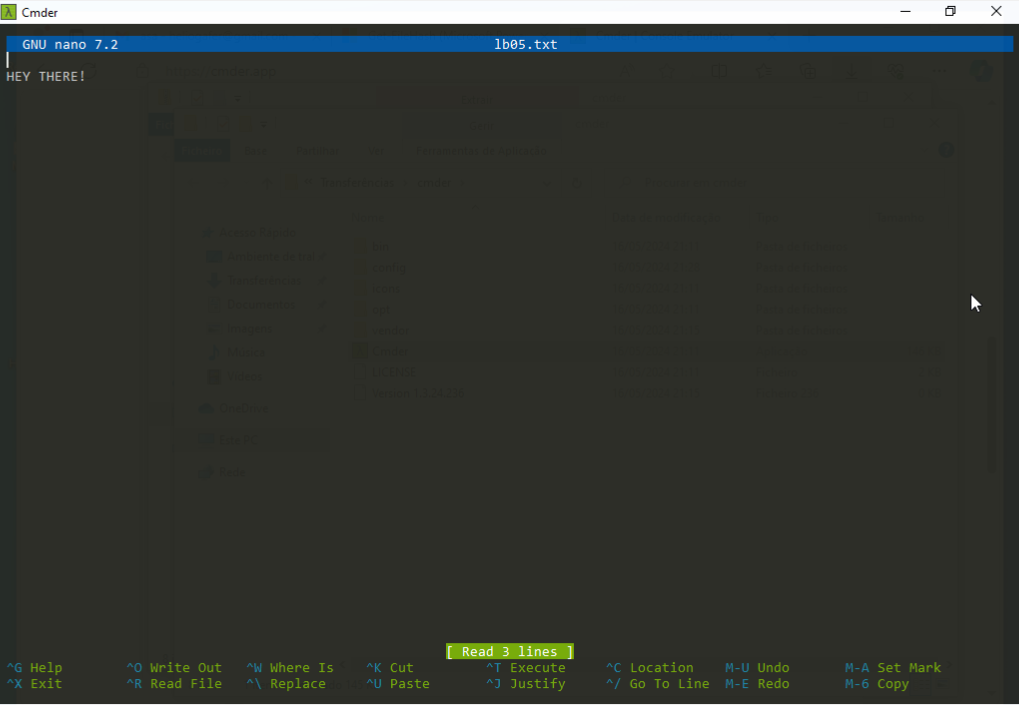
### **Part 5: Hash Validation**

Hash validation is a process by which we compare a file’s original hash to the hash of the file in front of us. Sometimes file servers and GitHub repos will publish the file’s hash in order for you to verify that the file you downloaded is legitimate and safe.

* Use PowerShell to generate the hash value of a small file, such as “example.txt”.
* Upload the file to Google Drive and generate a shareable link.  
    
  
* Share the file with a classmate, along with the hash value.
* Have your classmate generate the hash value of the downloaded file and compare it to the value you provided.  
  This Screenshot was sent by my friend Bruno Frenandes  
    
  
* Once you have validated the hash of a file sent to you by a classmate, post screenshot evidence to your submission for the day.

### **Part 6: Data File Encryption and Secret Messages with OpenSSL in Linux**

A versatile cryptography library, OpenSSL can also help us encrypt files.

* Compose a .txt file in Linux and write a short sentence in it, such as “Hello world!” or the like.
* Encrypt the .txt file with OpenSSL.
* Contact a classmate and ask them if it’s OK to send an OpenSSL-encrypted file over.   
  Transmit once they’ve responded.
* When you have received the OpenSSL-encrypted file from your classmate, decrypt it.
* Include screenshots and contents of the file you decrypted in your submission. Explain the steps taken.

OpenSSL can also encrypt a string of text.

* Use OpenSSL to generate base64 ciphertext from a cleartext string sentence.
* Post your ciphertext to Slack class channel for others to decrypt. Don’t indicate this is Part 7; keep your classmates guessing!
* Once a classmate has posted a base64 ciphertext string, decrypt it and add it to your submission alongside the original hash value.

### **Part 7: Reporting**

Include in your Google Doc the below:

* Why might you encrypt data in the workplace?  
    
  There are two main reasons for encrypting data in the workplace:  
    
  **Confidentiality**: This is the big one. Encryption scrambles data into an unreadable format, making it gibberish to anyone who doesn't have the decryption key. This is crucial for protecting sensitive information in your company.  
    
  **Compliance**: Many regulations across different industries mandate organizations to secure sensitive data. Encrypting data demonstrates your commitment to data security and helps you meet these compliance requirements. This can avoid hefty fines and reputational damage from data breaches.
* Why might you hash a file in the workplace?  
    
  Hashing a file in the workplace serves a different purpose than encryption. Here's why it's valuable:

**Verifying Data Integrity:** Imagine a critical document being transferred or downloaded. Hashing creates a unique fingerprint, like a digital checksum, for the file. This hash acts as a reference point. Once the file reaches its destination, you can recalculate the hash and compare it to the original one. If they match, you can be confident the file hasn't been tampered with during transfer.

**Ensuring Consistency:** Hashing is handy for version control. When you save different versions of a file, hashing allows you to confirm you have the correct iteration. This is important for collaborative projects or ensuring legal documents haven't been accidentally modified.

**Identifying Duplicates:** Hashes can help identify identical files stored across different systems. This can be useful for optimizing storage space by eliminating redundant copies.

* What are some limitations to this type of encryption? Describe a few scenarios where these techniques would not be as useful.  
    
  Here are some limitations to consider:

**Key Management:** Encryption relies on keys for decryption. These keys need to be secure, but managing them can be tricky. Losing a key can render encrypted data permanently inaccessible. Having weak keys or poor key management practices can make it easier for attackers to crack the encryption.

**Performance Overhead:** The encryption and decryption processes take time and resources. This can slow down systems, especially for large datasets or real-time processing. Finding the right balance between security and performance is crucial.

**Recovery Challenges:** Imagine a system crash or accidental deletion of encrypted data. Without proper backups, even with the decryption key, recovering the data becomes impossible. Encryption strategies need to be paired with robust backup procedures.  
  
Possible Scenarios that encryption might not be the best solution:  
  
**Legacy Systems:** Some older systems might not be technically equipped to handle encryption. Forcing encryption on these systems could lead to compatibility issues and disrupt workflows.

**Real-time Collaboration:** Constantly encrypting and decrypting files during collaborative editing can be cumbersome and slow down the process. Alternative access controls or data compartmentalization might be more suitable in such situations.

**Sharing with Third Parties:** Securely sharing encrypted data with external partners requires establishing methods for safe key exchange, adding an extra layer of complexity to the process. Depending on the data and collaboration needs, alternative security measures might be more efficient.

* How could data file encryption be used offensively by a cyber criminal or a pentester?  
    
  Data file encryption can be a double-edged sword, and while it protects data confidentiality, cybercriminals and pen testers can exploit it for malicious purposes:

**Ransomware:** This is a classic example. Hackers infiltrate a system, encrypt critical files, and then demand a ransom from the victim to provide the decryption key. This can cripple a company's operations, forcing them to pay to regain access to their own data.

**Data Exfiltration:** Even if attackers can't access the content of encrypted data immediately, they might steal it and hold it for ransom later. They can also sell the encrypted data on the black market, where other criminals might try to crack the encryption or use it for other malicious purposes.

**Penetration Testing (Pen Testing):** Pen testers are ethical hackers who simulate real-world attacks to identify vulnerabilities in a system's security. During a pen test, they might use encryption to mask their activities and make it harder for defenders to detect their movements within the system. This helps evaluate the effectiveness of an organization's intrusion detection and response measures.