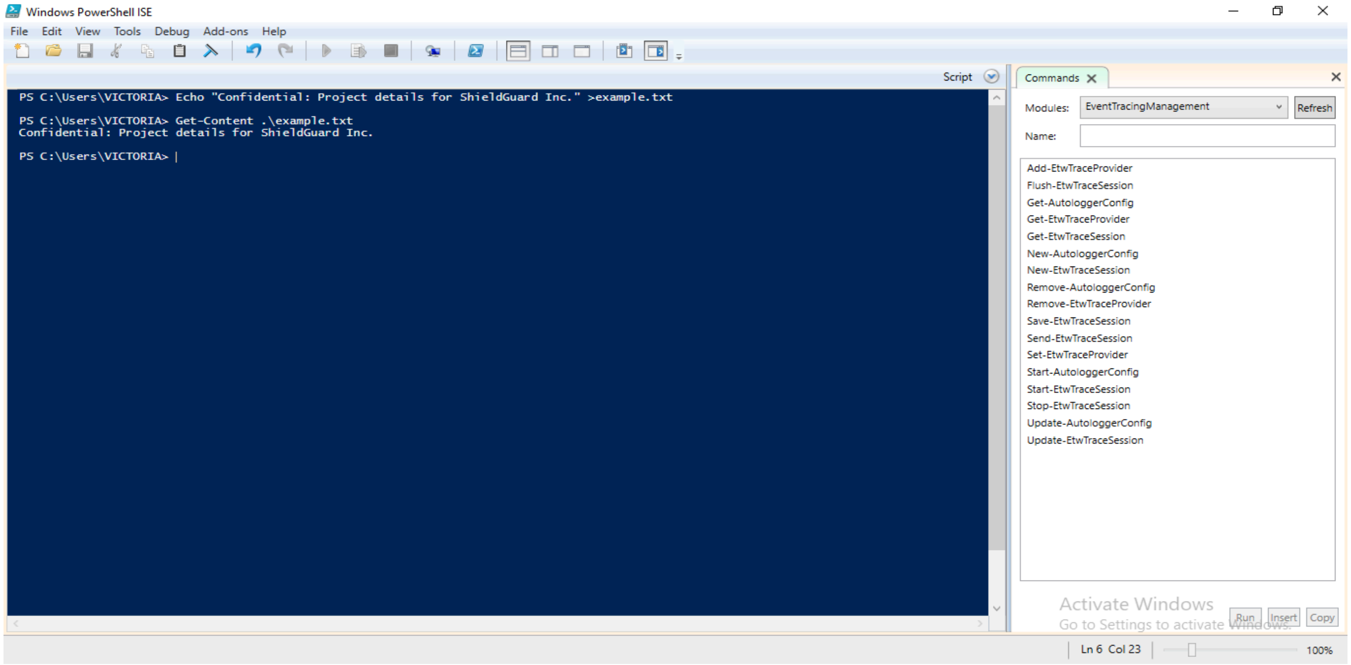
Demonstration of Data Integrity Using Cryptographic Hash Functions

Hash Function

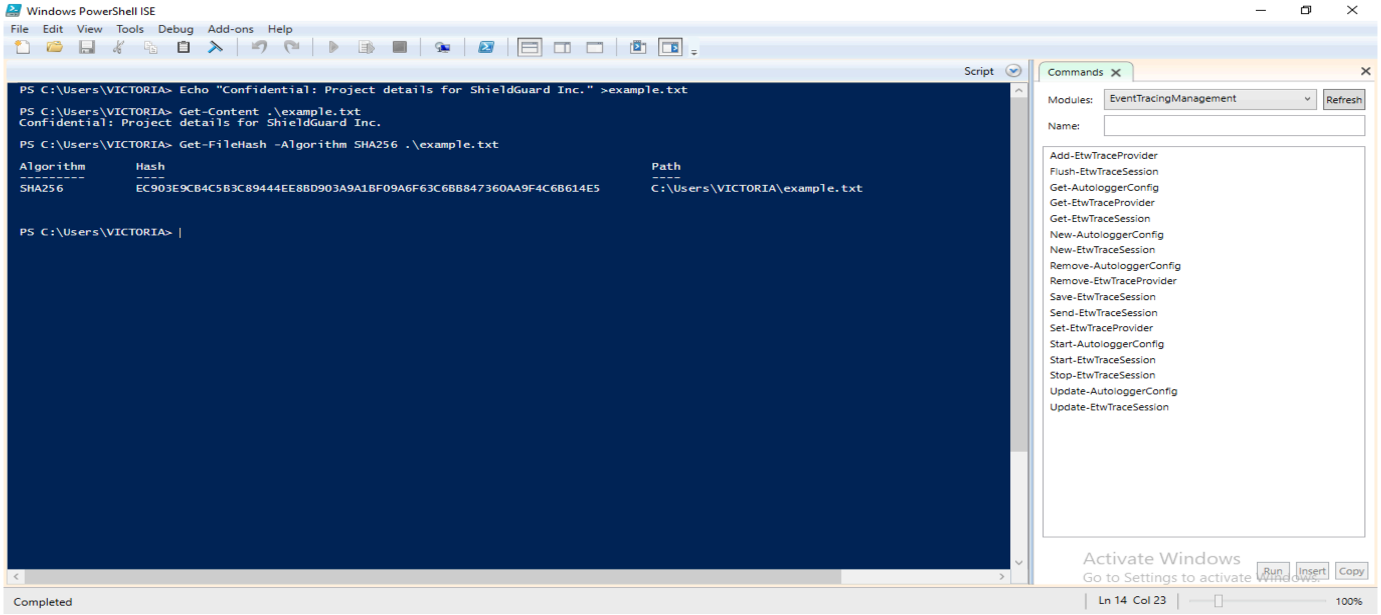
I made use of SHA256

The original file content:



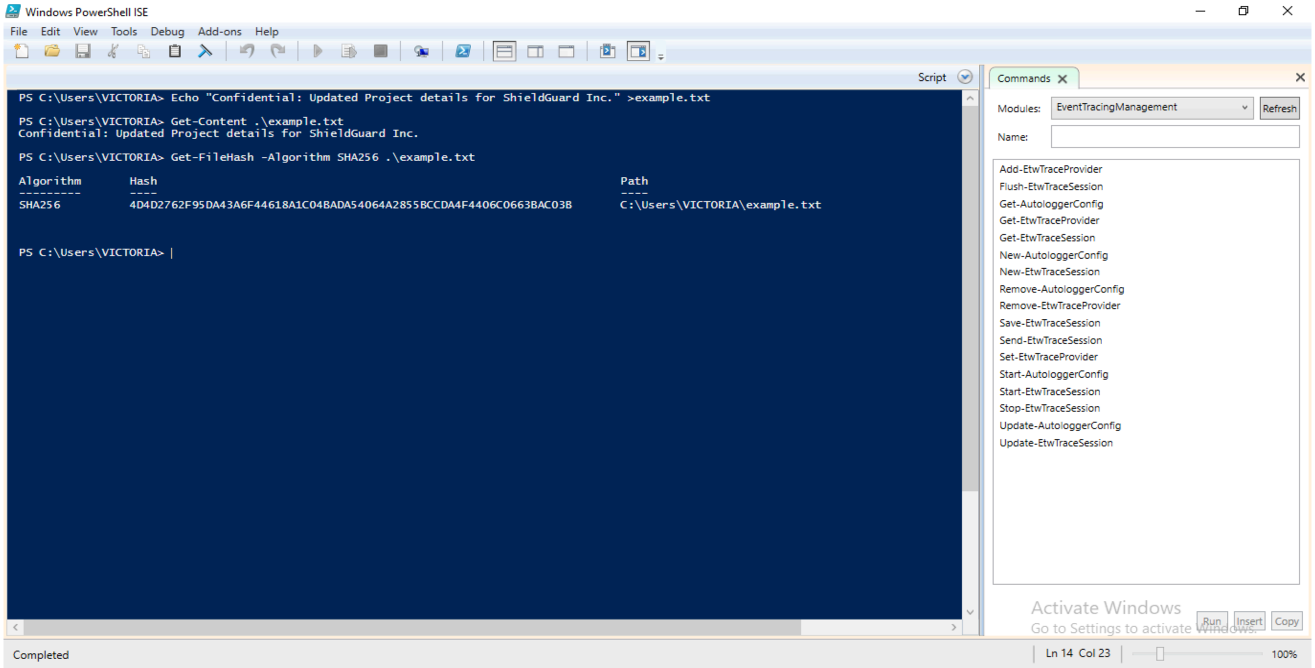
```
PS C:\Users\VICTORIA> Echo "Confidential: Project details for ShieldGuard Inc." >example.txt

PS C:\Users\VICTORIA> Get-Content .\example.txt
Confidential: Project details for ShieldGuard Inc.

PS C:\Users\VICTORIA> |
```

The Hash Value:



```
PS C:\Users\VICTORIA> Echo "Confidential: Project details for ShieldGuard Inc." >example.txt

PS C:\Users\VICTORIA> Get-Content .\example.txt
Confidential: Project details for ShieldGuard Inc.

PS C:\Users\VICTORIA> Get-FileHash -Algorithm SHA256 .\example.txt

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          EC903E9CB4C5B3C89444EE8BD903A9A1BF09A6F63C6BB847360AA9F4C6B614E5       C:\Users\VICTORIA\example.txt


PS C:\Users\VICTORIA> |
```

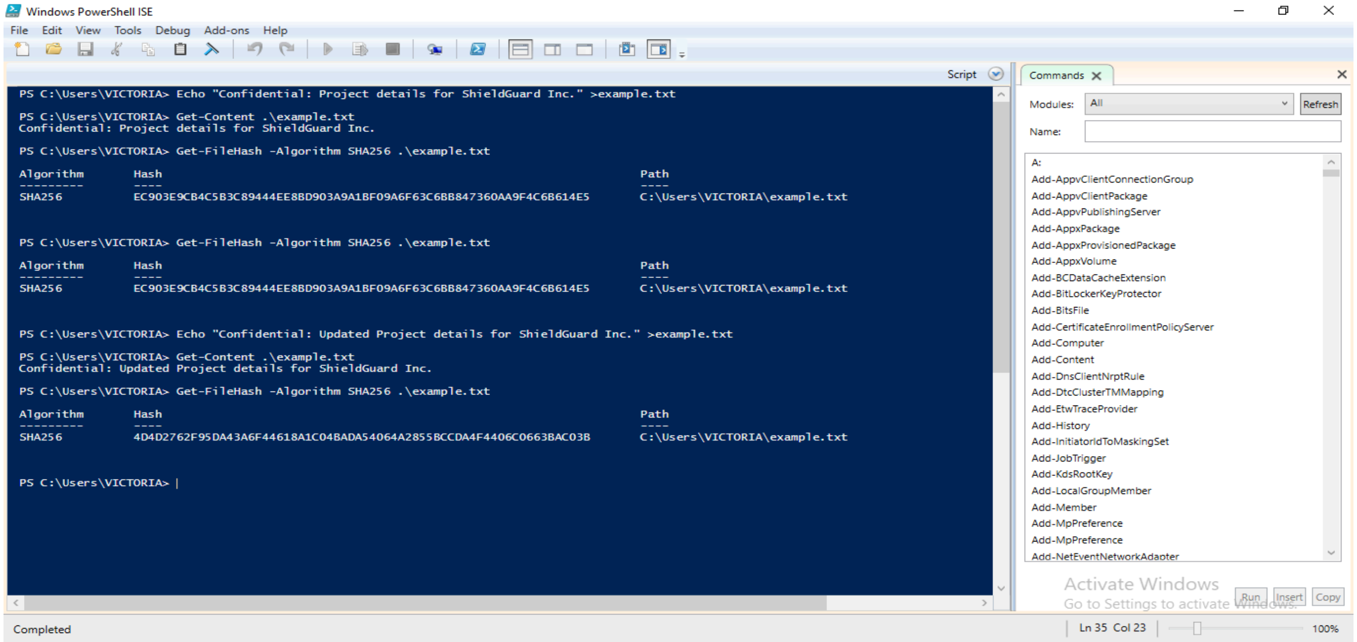
The Modified File Content and it's Hash Value:



```
PS C:\Users\VICTORIA> Echo "Confidential: Updated Project details for ShieldGuard Inc." >example.txt

PS C:\Users\VICTORIA> Get-Content .\example.txt
Confidential: Updated Project details for ShieldGuard Inc.

PS C:\Users\VICTORIA> Get-FileHash -Algorithm SHA256 .\example.txt

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          4D4D2762F95DA43A6F44618A1C04BADA54064A2855BCCDA4F4406C0663BAC03B       C:\Users\VICTORIA\example.txt



PS C:\Users\VICTORIA> |
```

Observation

I observed that even a minor alteration in the input data will produce a significantly different hash value. By comparing the hash of the original file content with the hash of the modified file content,I noticed there was a change in their hash value due to the modification of the original file.



```
PS C:\Users\VICTORIA> Echo "Confidential: Project details for ShieldGuard Inc." >example.txt

PS C:\Users\VICTORIA> Get-Content .\example.txt
Confidential: Project details for ShieldGuard Inc.

PS C:\Users\VICTORIA> Get-FileHash -Algorithm SHA256 .\example.txt

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          EC903E9CB4C5B3C89444EE8BD903A9A1BF09A6F63C6BB847360AA9F4C6B614E5       C:\Users\VICTORIA\example.txt


PS C:\Users\VICTORIA> Get-FileHash -Algorithm SHA256 .\example.txt

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          EC903E9CB4C5B3C89444EE8BD903A9A1BF09A6F63C6BB847360AA9F4C6B614E5       C:\Users\VICTORIA\example.txt


PS C:\Users\VICTORIA> Echo "Confidential: Updated Project details for ShieldGuard Inc." >example.txt

PS C:\Users\VICTORIA> Get-Content .\example.txt
Confidential: Updated Project details for ShieldGuard Inc.

PS C:\Users\VICTORIA> Get-FileHash -Algorithm SHA256 .\example.txt

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          4D4D2762F95DA43A6F44618A1C04BADA54064A2855BCCDA4F4406C0663BAC03B       C:\Users\VICTORIA\example.txt


PS C:\Users\VICTORIA> |
```

Insights on Cryptography

Cryptographic hash functions are crucial for ensuring data integrity because they create a unique, fixed-size "fingerprint" of data, allowing for easy detection of any changes or tampering. They act as a digital checksum, ensuring that data remains unchanged during transmission or storage.

Here's a more detailed look at their importance:

1. Data Integrity Verification

- Hash functions generate a unique hash value for any given input, no matter the size of the input.
- Even the slightest alteration to the original data will result in a completely different hash value.
- By comparing the hash of the original data with the hash of the received or stored data, one can quickly determine if any changes have occurred, ensuring data integrity.
- This is used in various scenarios, including file integrity checking, digital signatures, and checksums.

2. Cybersecurity Applications:

- Password Storage: Instead of storing passwords directly, systems store their hash values.
- Digital Signatures: Hash functions are used to create a hash of the data being signed, which is then signed using a private key, ensuring the document's authenticity and integrity.
- Blockchain Technology: Hash functions are used to link blocks together chronologically in blockchain networks, making tampering nearly impossible.

3. Key Properties of Hash Functions:

- Pre-image resistance: It should be computationally infeasible to find the original input data given only its hash value.
- Second pre-image resistance: It should be computationally infeasible to find a different input that produces the same hash value as a given input.
- Collision resistance: It should be computationally infeasible to find two different inputs that produce the same hash value.

In essence, cryptographic hash functions are essential for maintaining the trustworthiness and integrity of data in various digital systems, from ensuring that downloaded files are not corrupted to securing online transactions and digital signatures.