**🧪 Cybersecurity Portfolio**

**Hide Files with OpenStego**

**🔍 Objective:**

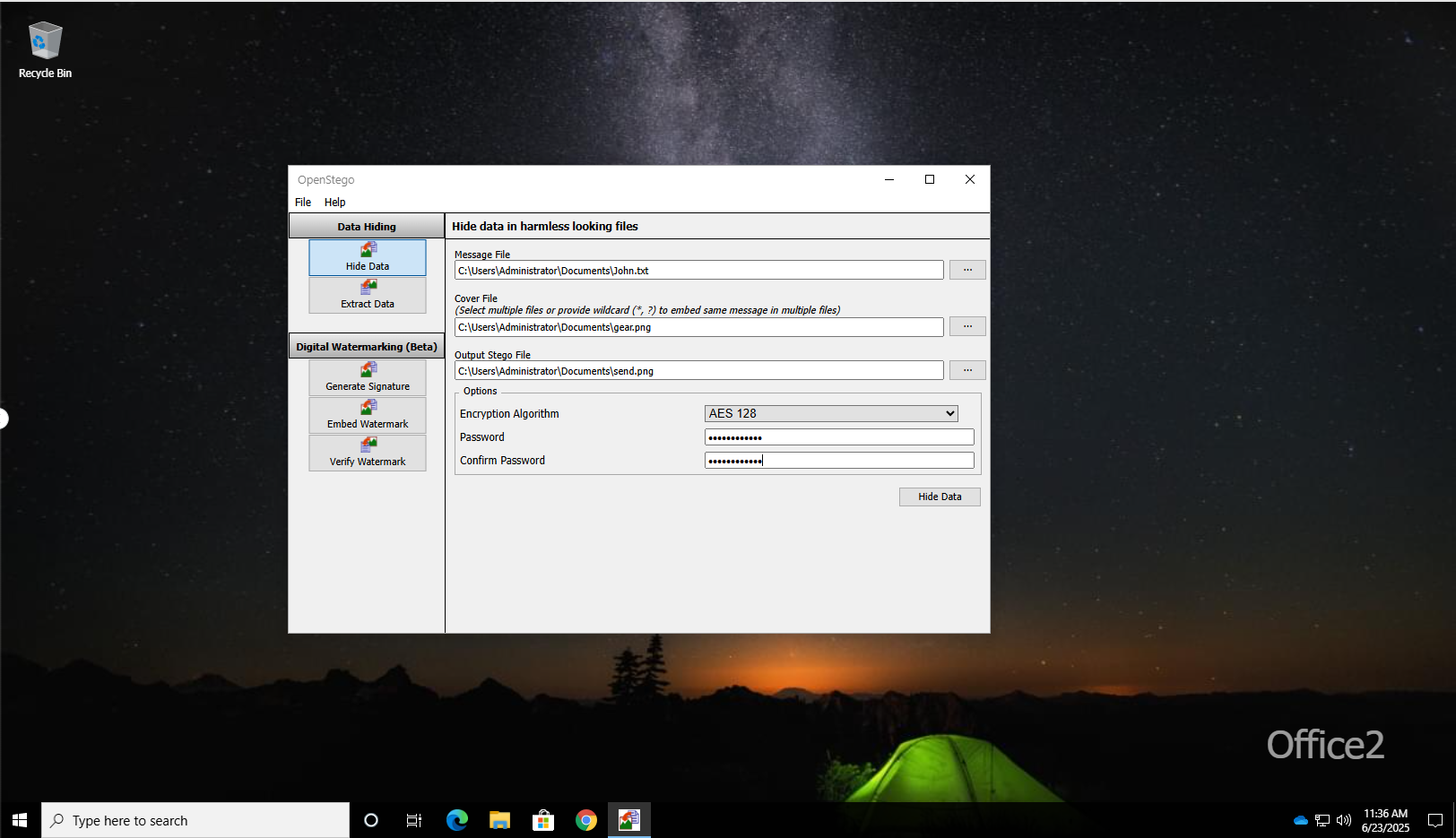
To use **steganography** to embed proprietary user data (from John.txt) into an image file using **OpenStego**, securing it with encryption and a password, and confirming successful extraction of the hidden data.

**🛠️ Tools & Files Used:**

* **Message File:** John.txt
* **Cover File (image):** gear.png
* **Stego Output File:** send.png (saved in Documents)
* **Password:** NoMor3L3@k$

**🧾 Steps Performed:**

1. **Launch OpenStego**
   * Opened the OpenStego application from the Windows environment.
2. **Embed Data into Image**
   * Selected the **Data Hiding** tab.
   * Selected John.txt as the **Message File**.
   * Selected gear.png as the **Cover File**.
   * Specified output file as send.png and saved it to Documents.
   * Enabled encryption with the password: NoMor3L3@k$.
   * Clicked **"Hide Data"** to generate the stego image.



1. **Verify Functionality**
   * Switched to the **Extract Data** tab in OpenStego.
   * Selected send.png as the **Stego File**.
   * Entered the same password: NoMor3L3@k$.
   * Extracted the embedded file to C:\Users\Administrator\Documents.
   * Opened the extracted file to verify that the **associated username was embedded successfully**.

**✅ Results:**

* The file send.png was successfully created with embedded, encrypted content.
* Extraction was completed using the correct password.
* The contents of the extracted file matched the original John.txt, confirming that the username had been embedded into the image.

**🧠 Key Concepts Learned:**

* **Steganography** can securely embed hidden data within a seemingly harmless image.
* Encryption with a password adds an extra layer of protection in case the stego file is intercepted.
* OpenStego provides a beginner-friendly interface for testing basic data-hiding concepts.