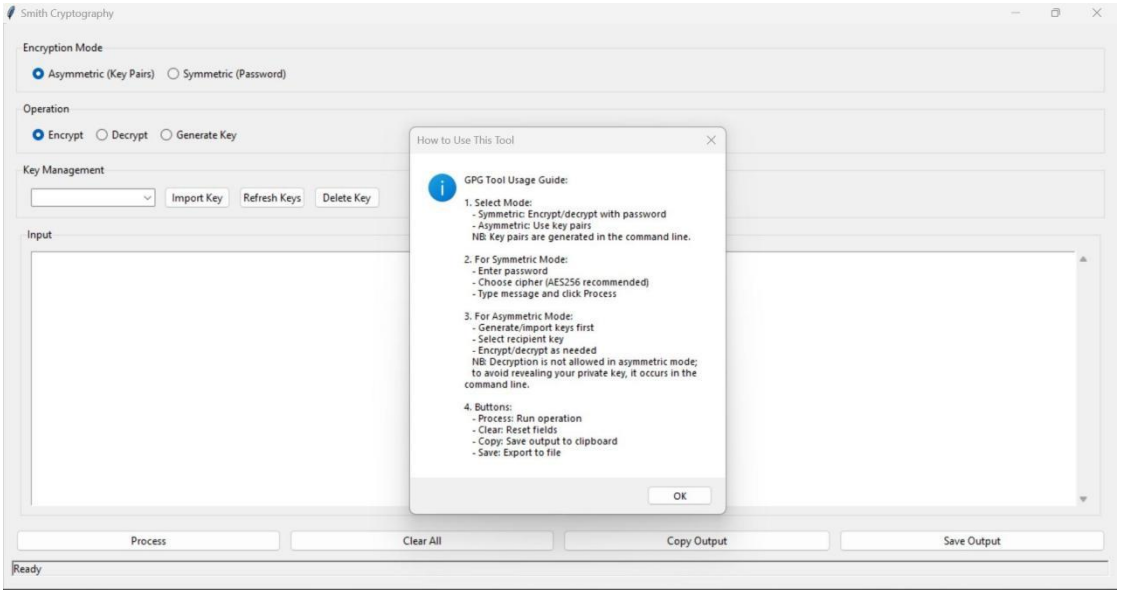
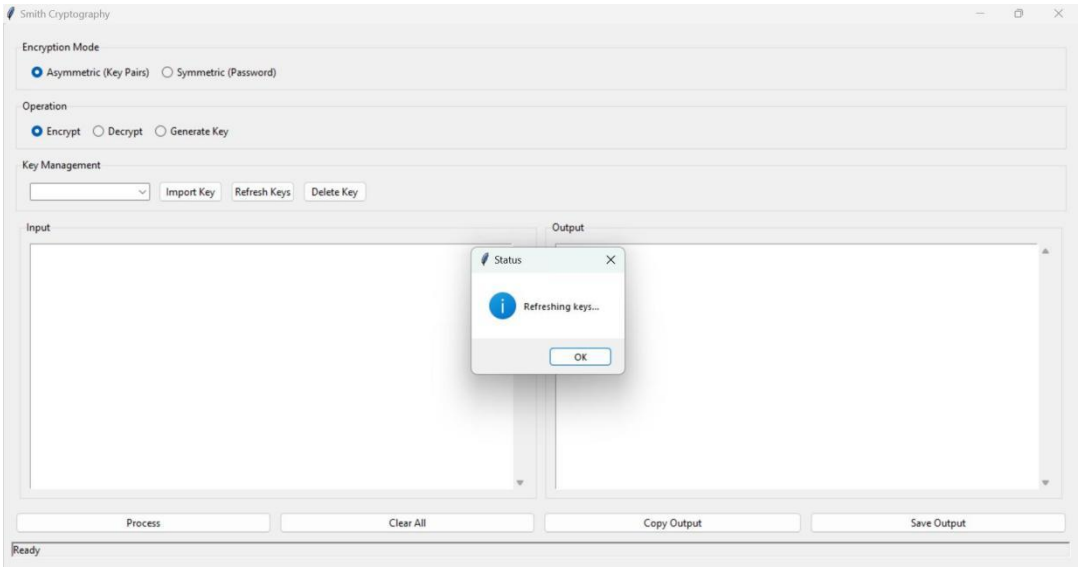
SECURITY TOOLKIT: GRACE SMITH

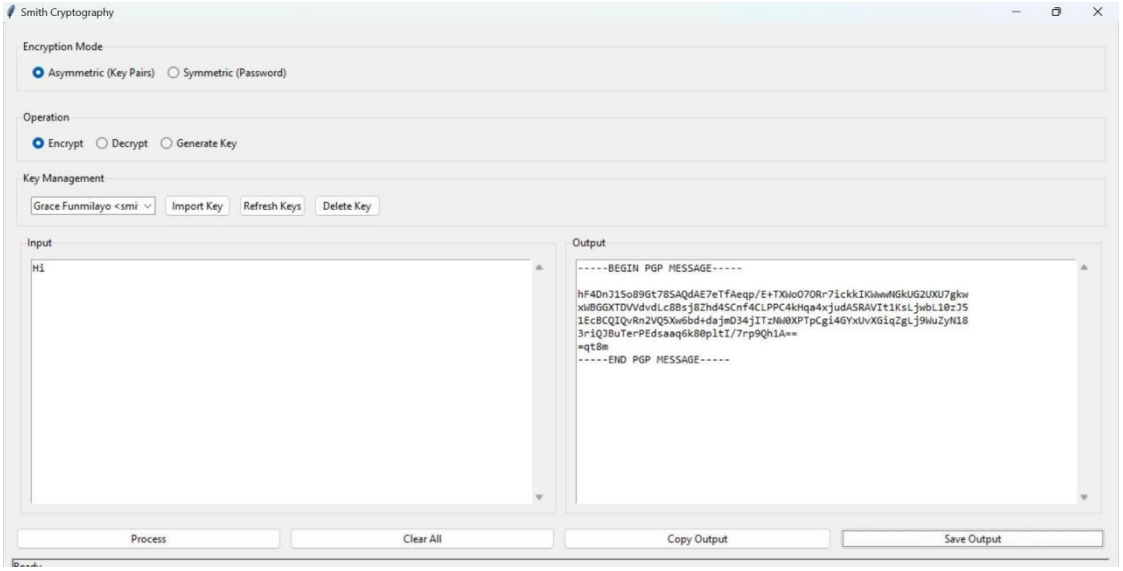
1: CRYPTOGRAPHY TOOL

Cryptography serves as the backbone for secure communication as it enables individuals and organizations to protect sensitive information from unauthorized access. One of the most widely used cryptographic methods is GnuPG(GPG), an open-source implementation of the OpenGPG standard that allows users to encrypt and sign data and communications. Smith Cryptography provides users with an intuitive platform to perform both symmetric and asymmetric encryption.

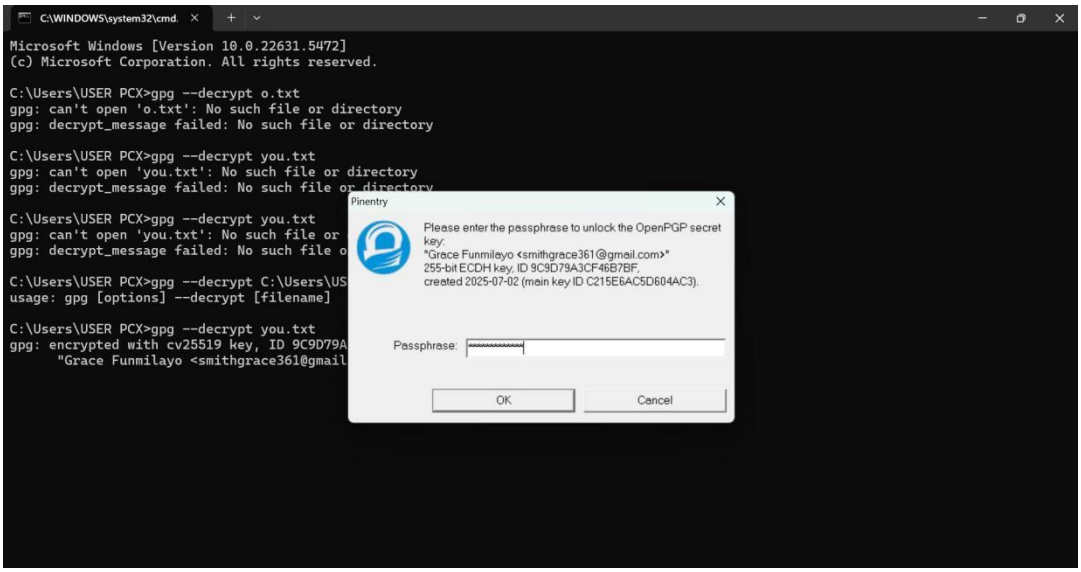


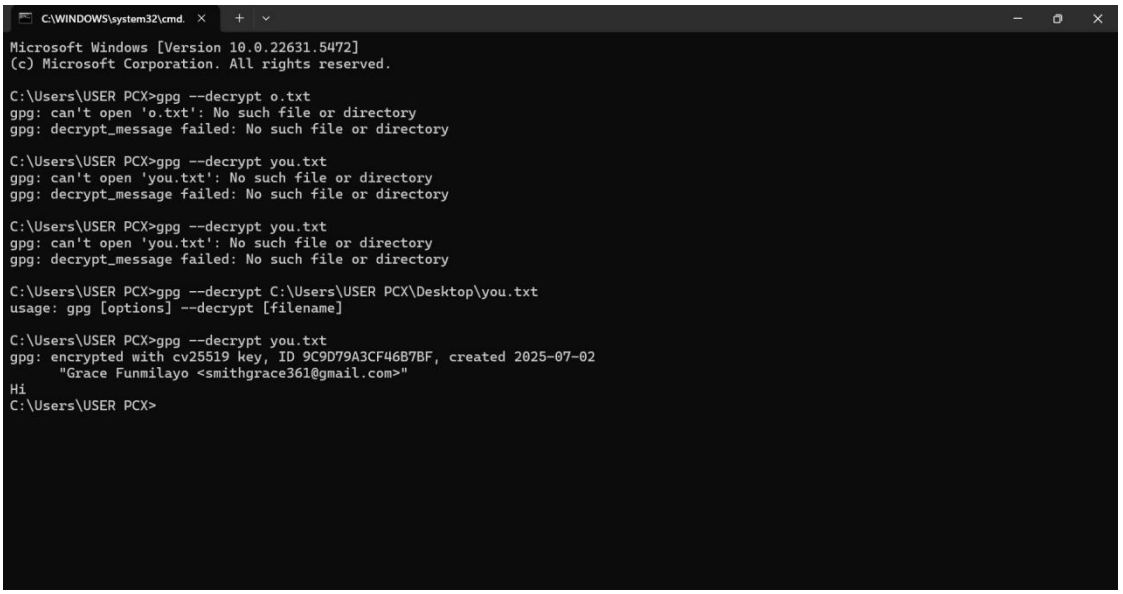


Asymmetric encryption relies on a pair of keys: a public key for encryption and a private key for decryption. These keys are created in the command line alongside a passphrase.

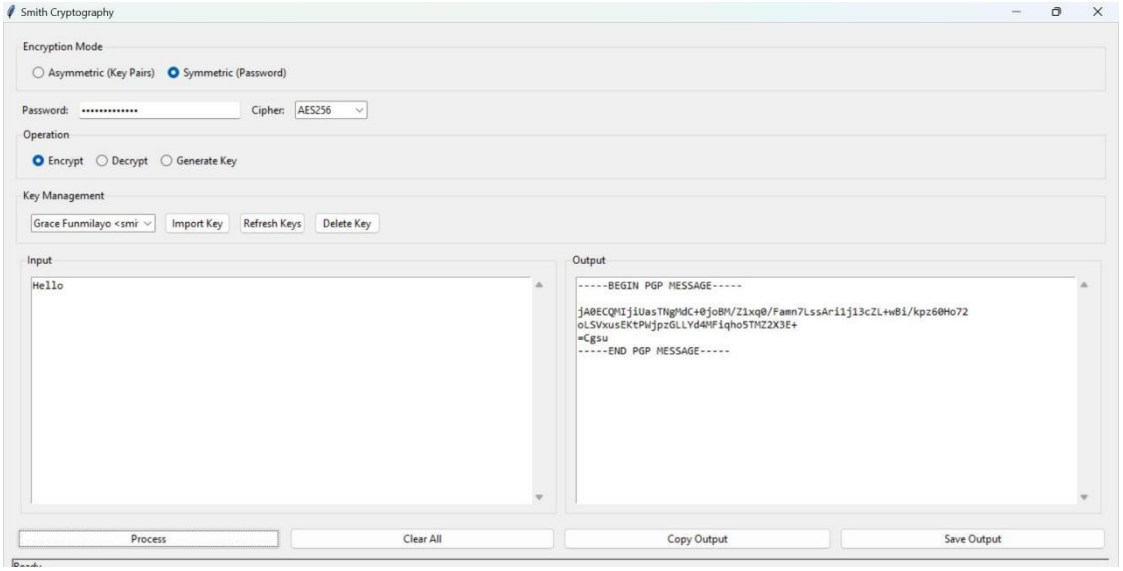


Encryption occurs in the command line with the private key and a passphrase to avoid accidentally revealing private key on the interface.



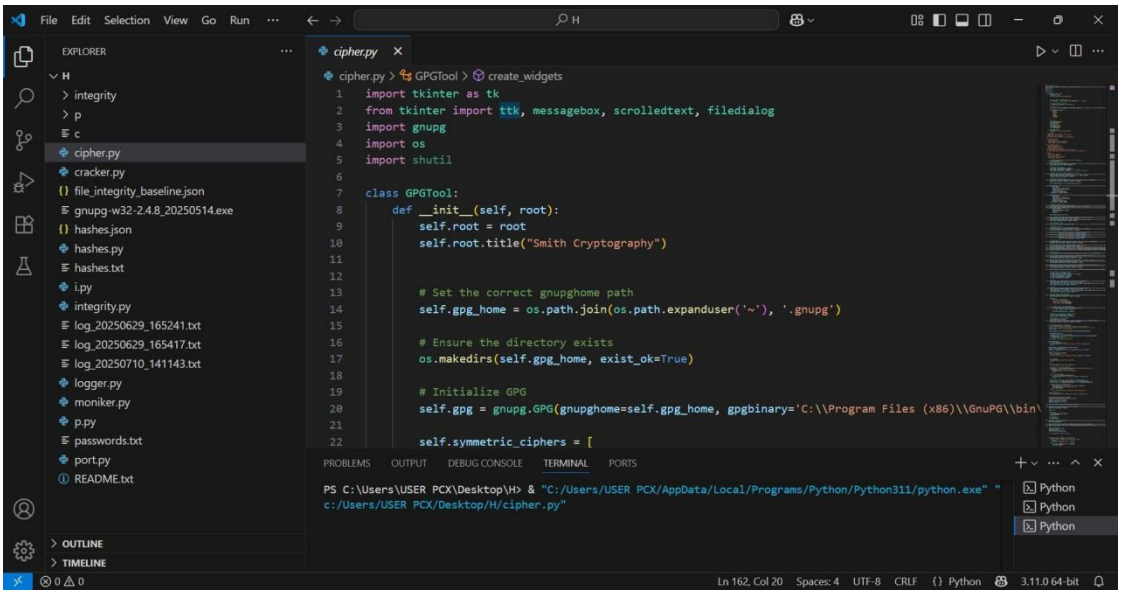


Symmetric encryption uses a shared secret(password) to encrypt and decrypt data. This shared secret is the passphrase generated during key generation.



Decryption also occurs in the interface with just the shared secret.

This app is solely based on a python script:



It can also be adopted for use in other operating systems like Kali Linux.

Key Takeaways:

1. Understanding Cryptography

2. Key Management

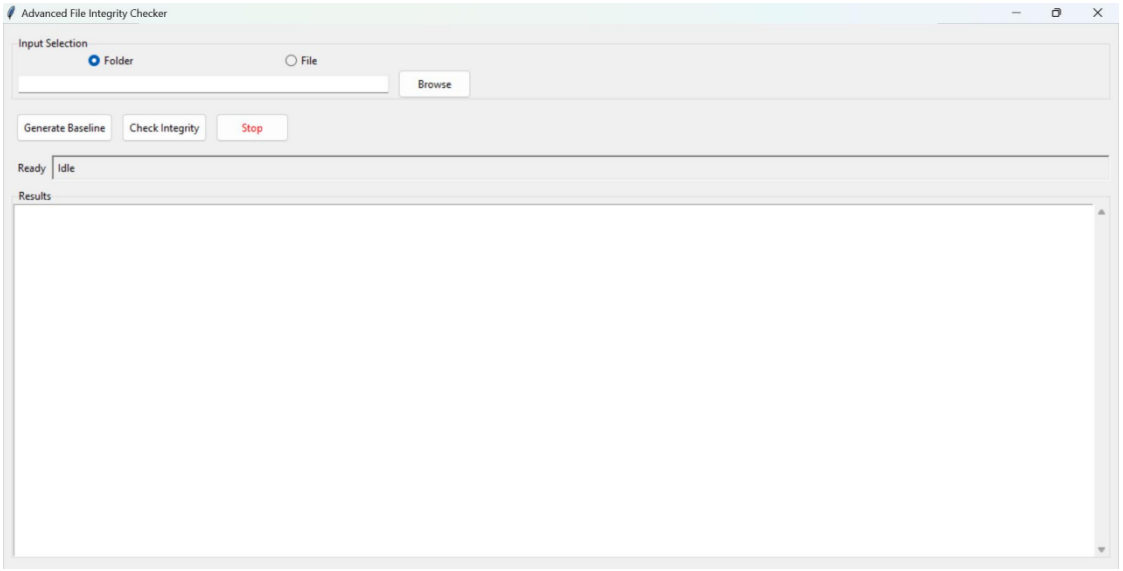
3. Practical application of GPG

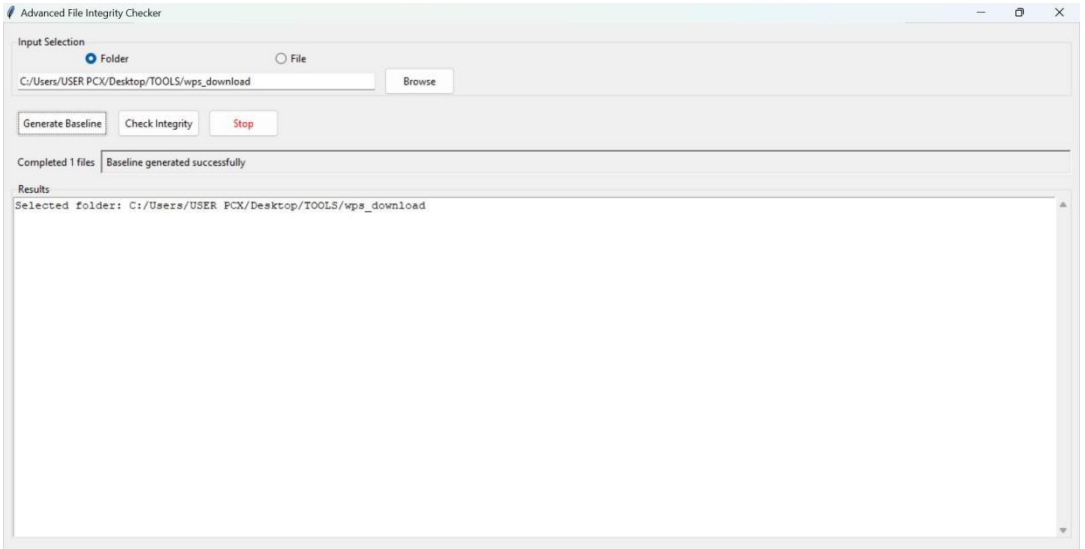
4. Ethical considerations

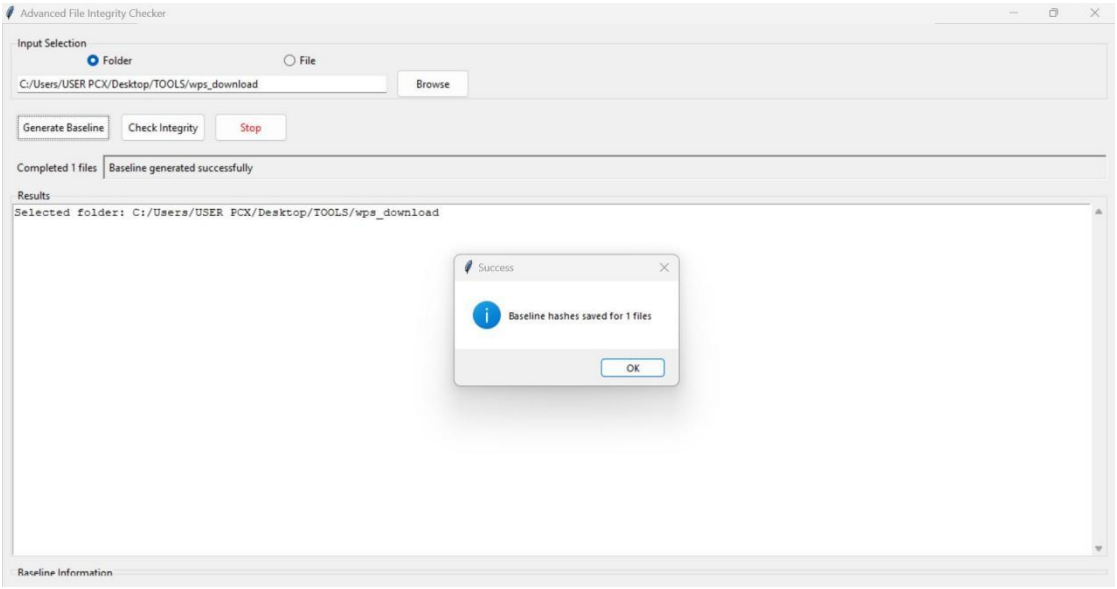
5. Data Protection

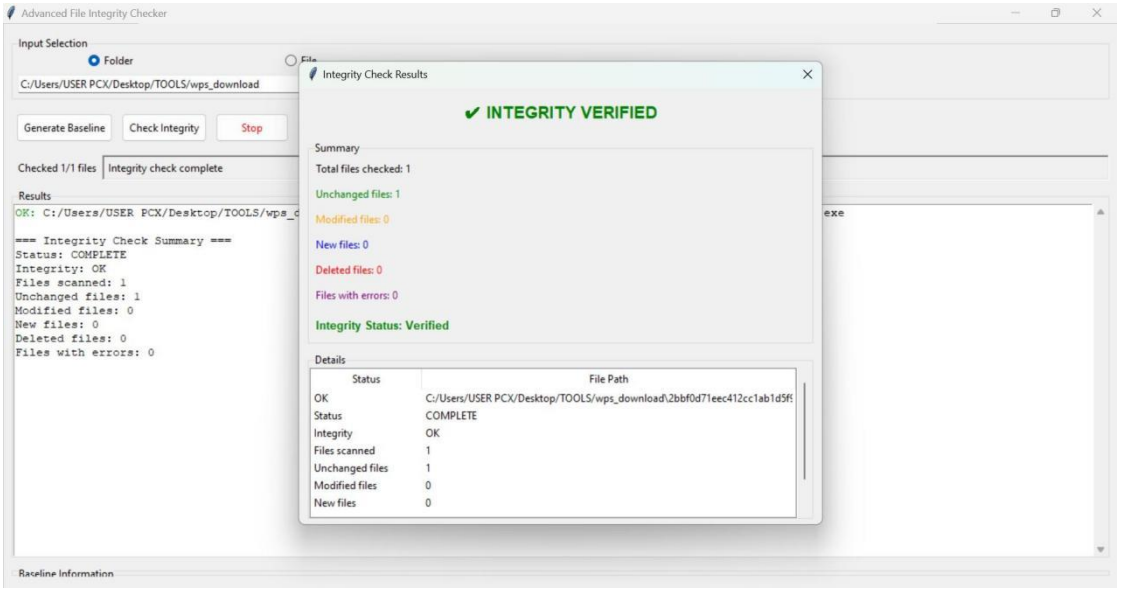
2: FILE INTERGRITY CHECKER

File integrity checking is a crucial process that helps detect unauthorized changes, corruption or tampering of files, particularly in environments where sensitive information is stored. By monitoring file integrity, users can quickly identify potential security breaches, data loss or system malfunctions. This tool allows users to generate baseline hashes for files and directories which serve as a reference point for future integrity checks.









This tool provides a user-friendly interface for selecting files and directories, generating hashes, and checking integrity violations. It also offers real-time feedback on the status of operations, making it accessible to users of varying levels of technical expertise.

It can also be adopted for use in other operating systems like Kali Linux.

Key Takeaways:

1. Understanding File Integrity

2. Cryptographic Hashing

3. Error Handling

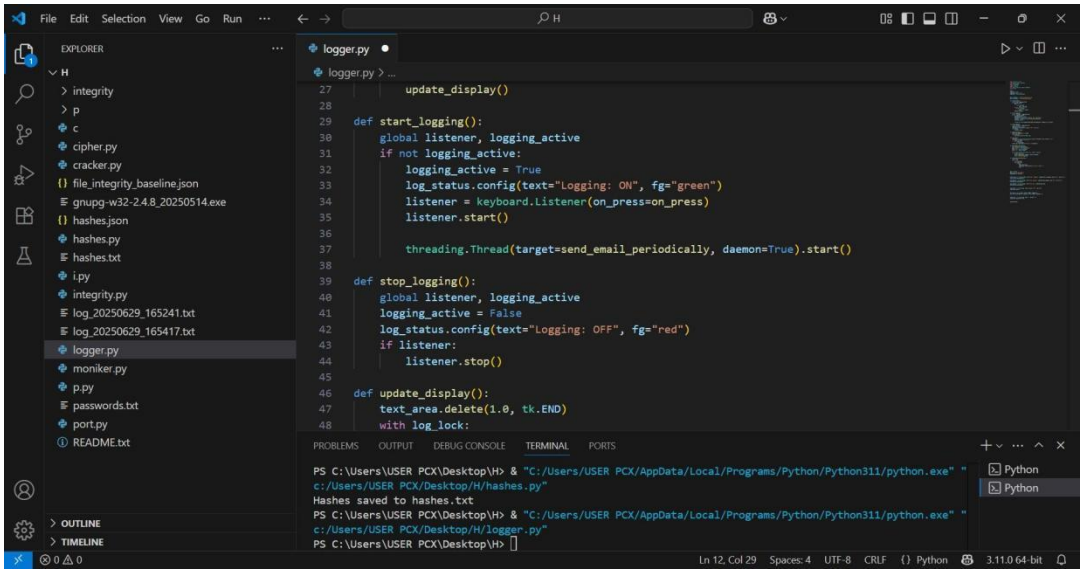
4. Security Awareness

3: KEYLOGGER

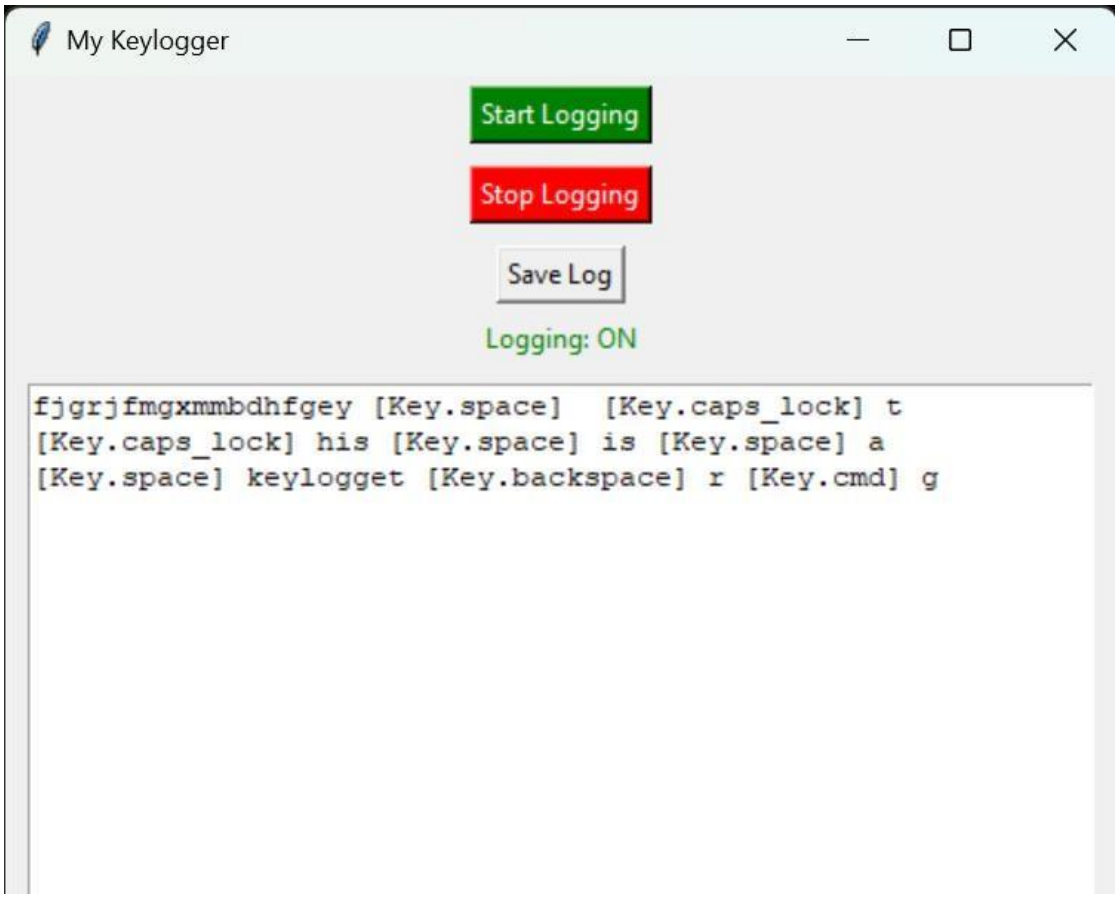
Key logging is a technique used to monitor and record keystrokes made on a computer. It serves different purposes, all of which are legitimate applications in cybersecurity. It is used by security professionals to access vulnerabilities in systems, monitor user behaviour and detect unauthorized access as well as ethical hacking exercises. This tool is a Keylogger.

This keylogger application captures keystrokes in real time while providing users with the ability to log keyboard activity, save logs to a file and send periodic email reports of logged data.

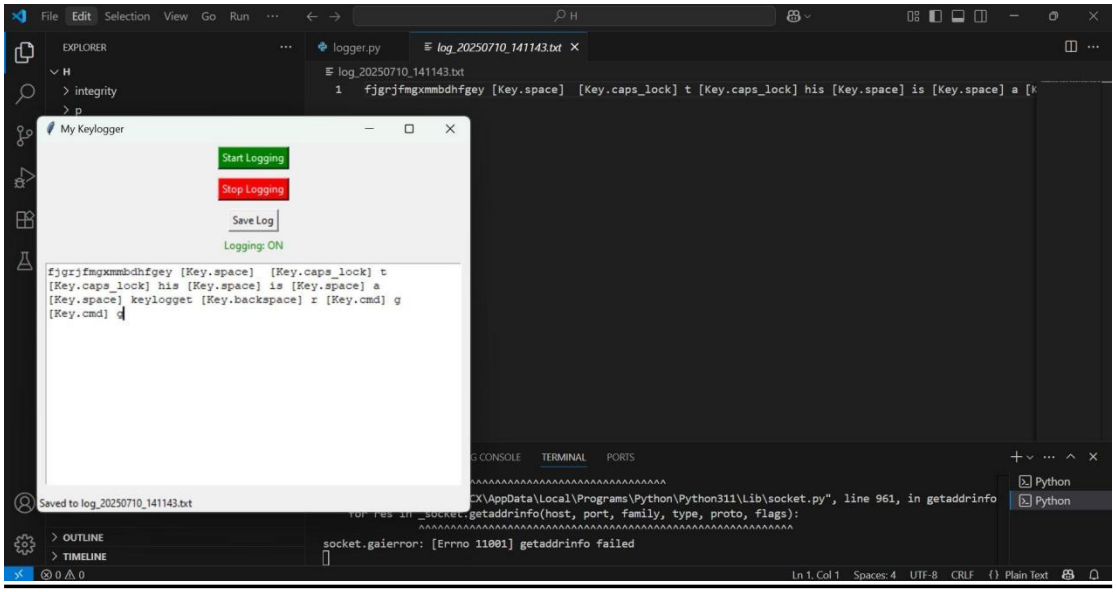
The source code interface



The GUI



Example of log saved into file



It can also be adopted for use in other operating systems like Kali Linux.

Key Takeaways:

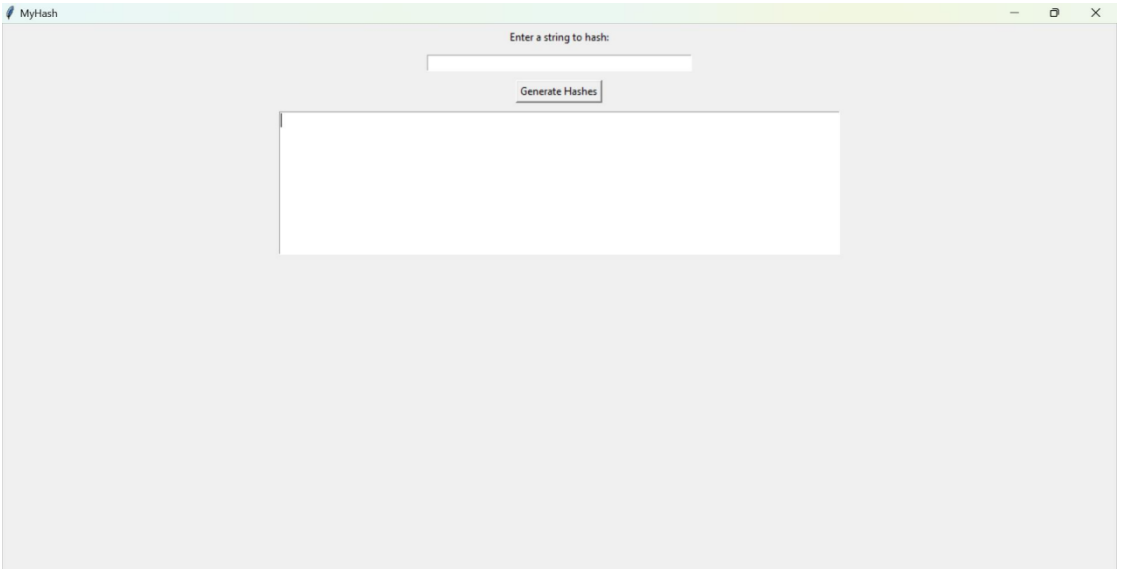
1. Understanding Keylogging

2. Threading and Concurrency: i.e sending emails and logging keystrokes concurrently without freezing the GUI

3. Ethical Considerations

4. Data Management

4: MyHash



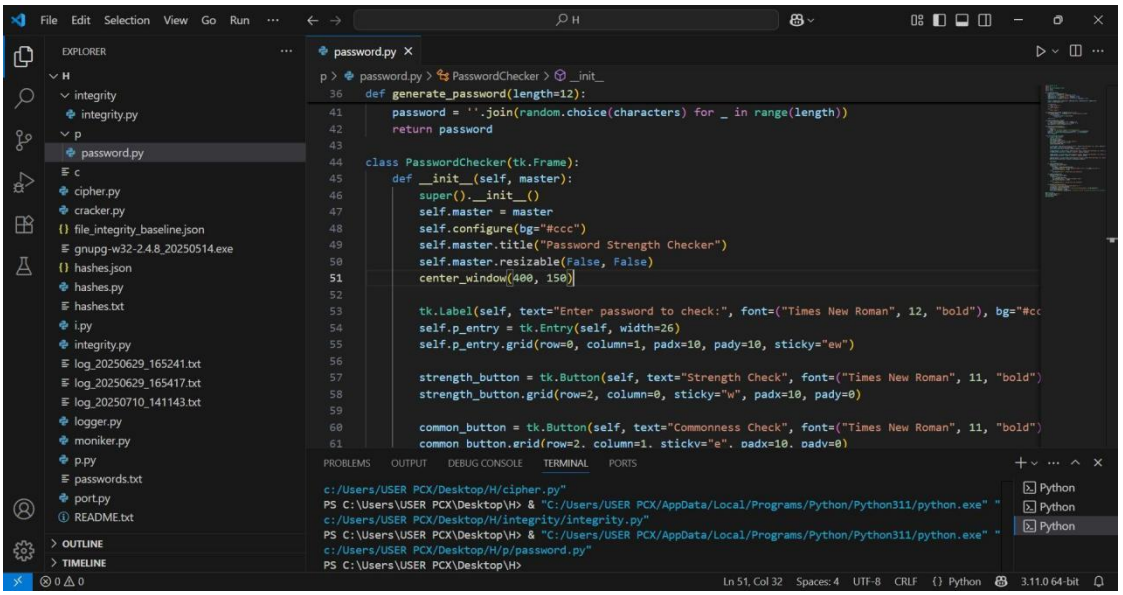
MyHash is a multi-format hashing tool relevant in cybersecurity, data integrity and password protection. It aids data integrity verification, password security (hashing & storage**),** file validation, time stamping, chain of custody and algorithm comparison.

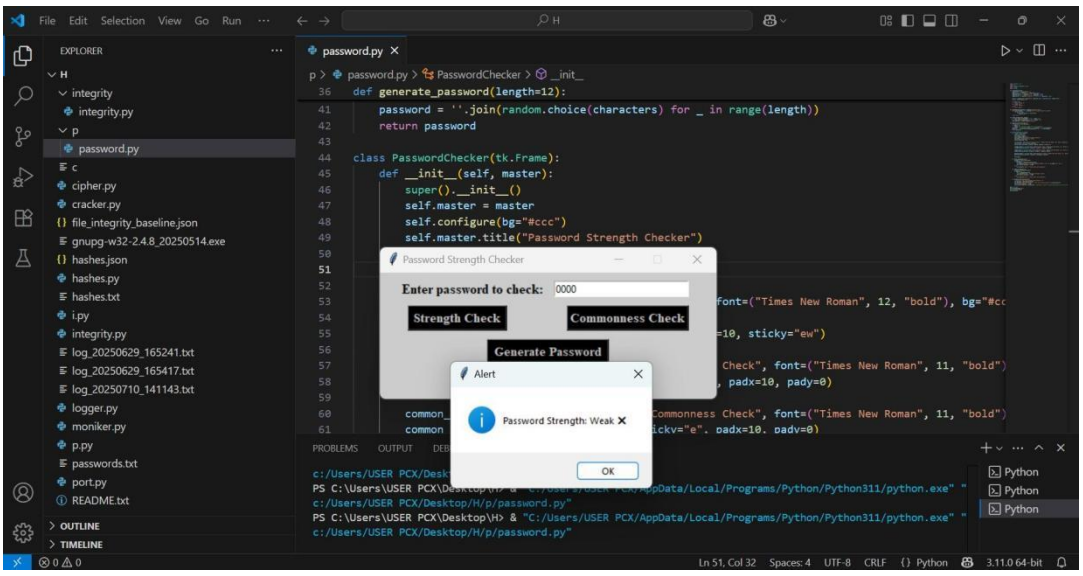
Key Takeaways

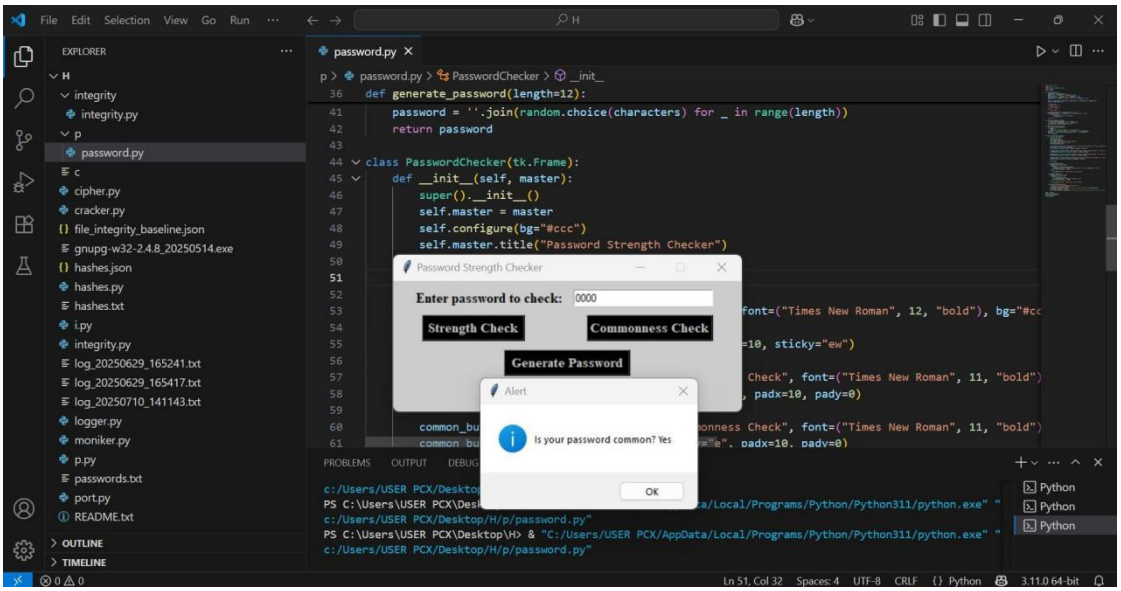
1. Understanding of Hashing Algorithms
2. Python Implementation Skills
3. Importance of Input Sensitivity
4. Testing and Debugging Security Code
5. Awareness of Hash Vulnerabilities
6. Practical Use Cases
7. Foundations for Advanced Security Tools e.g file integrity checkers, digital signature validators, custom password managers and forensics kits.

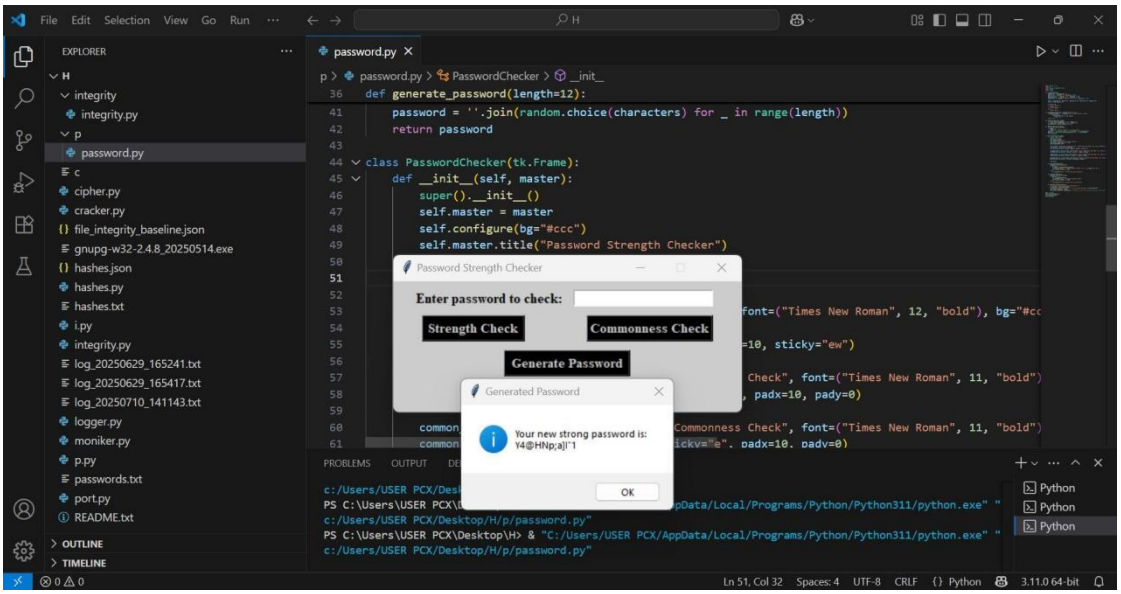
5:PASSFORGE

In today's digital landscape, the importance of strong, secure passwords cannot be overstated. They serve as first line of defense against unauthorized access to personal information. Many users still rely on weak or easily guessable passwords. Passforge helps users develop robust passwords and also check existing passwords for strength and commonness.









This app, as you might be able to tell, runs solely on python script. It can also be adopted for use on other operating systems like Kali Linux.

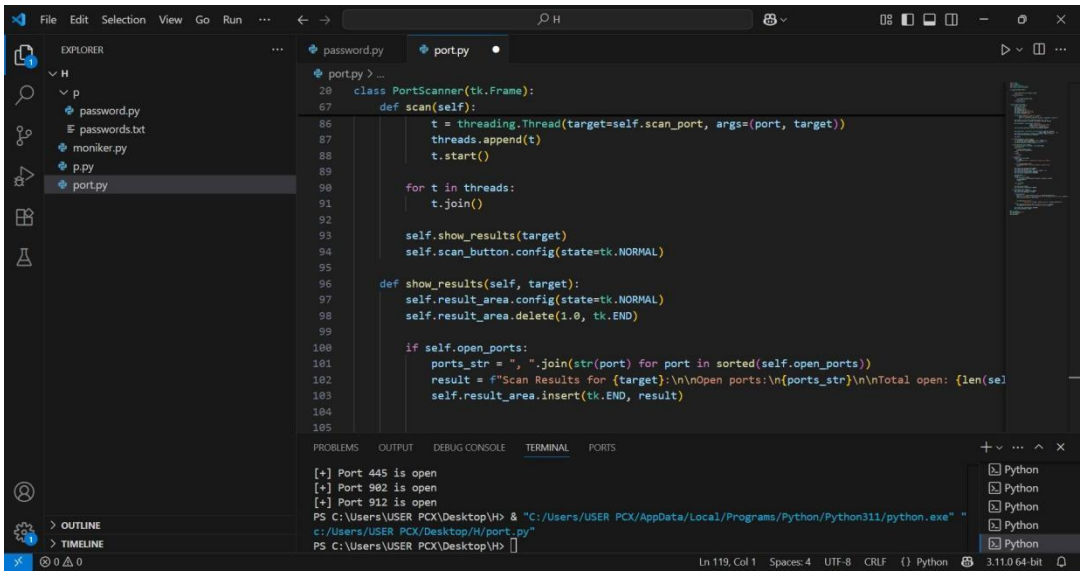
Key Takeaways:

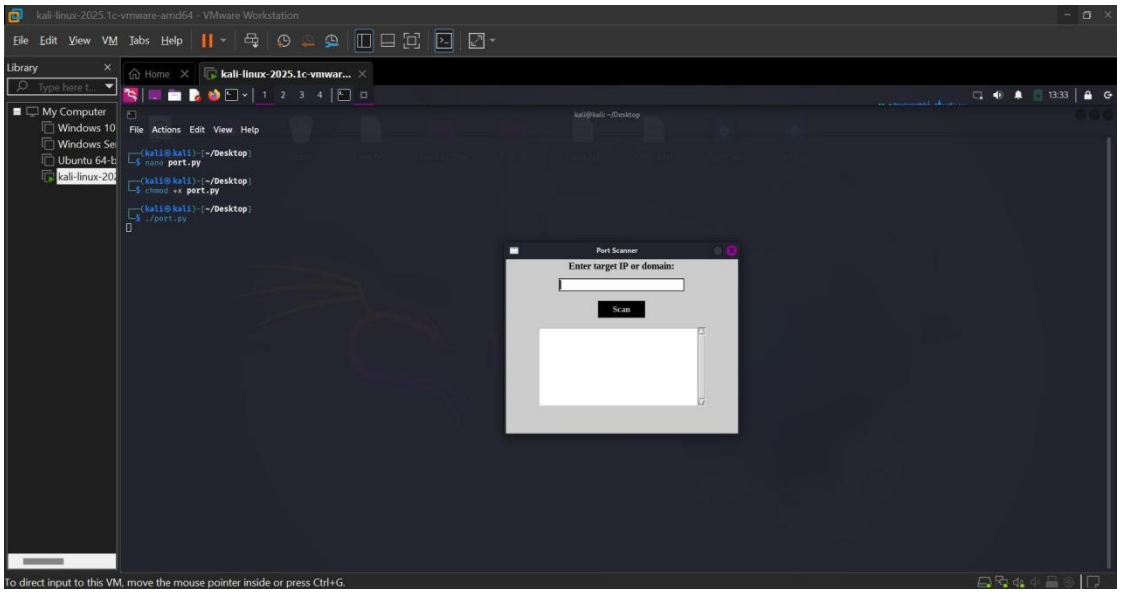
1. Understanding password security and its importance.

2. Password Management.

6:PORT SCANNER

Port scanning is a fundamental reconnaissance technique because it is usually the first line of assessment. It helps with mapping ports, services and vulnerabilities in a system.





This port scanner is custom built and designed with an interactive interface to allow efficient scanning, clear results and error handling. Its features include user input handling, port scanning, connection attempts, real-time feedback, results compilation and display as well as error handling. Whether for hardening networks or ethical hacking exercises, this port scanner combines multi-threading and protocol support with and interactive interface, and user friendly output. It runs on both Windows and Kali Linux operating systems.

Key Takeaways:

1. Networking Fundamentals:
- Understanding of Ports and Protocols
- Socket programming

2. Concurrency and Multithreading:
- Thread Management
- Efficiency

3. User Interface Development:
- GUI & Event Driven Programming

4. Input Validation and Error Handling:
- Data Validation

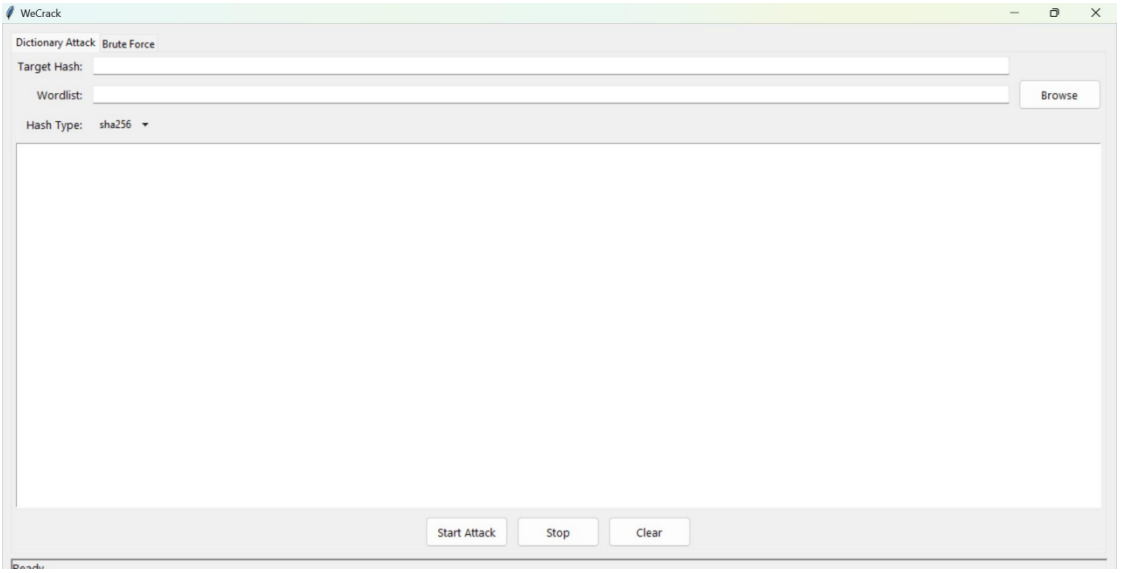
5. Programming Skills:
- Python Proficiency
- Code organization

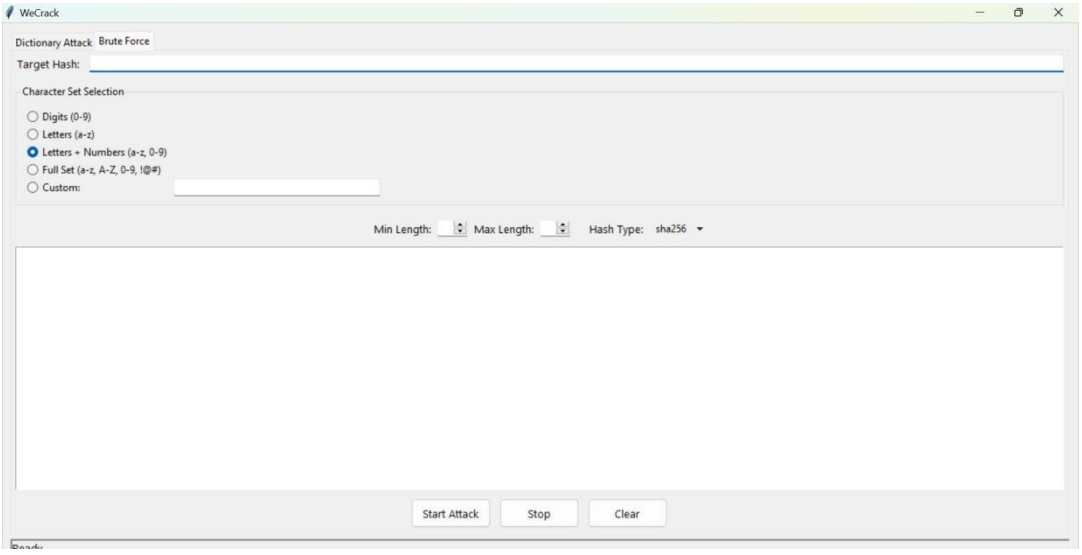
6. Practical Application:

- Real World UseCases
- Tool Development

7: WeCrack

WeCrack is a password cracker built based on hashing algorithms. It simulates how password cracking works in ethical hacking, shows how **hash functions** like MD5, SHA-1, SHA-256 are used in practice and promotes the need for **strong password policies** and **salting techniques.**





WeCrack is a simple password cracker that supports both brute force and dictionary attacks. Its features include a user-friendly GUI, multiple algorithm support, customizable character set and an extensible codebase. It is the first version because for now the dictionary attacks seem more efficient. It is open for lots of improvement, but it also serves as foundation for more advanced tools and internal research.

Key Takeaways

- **Understanding of Hash Matching:**
  - Learned how to take a hashed password and reverse it using brute force and dictionary-based methods.

- **Testing & Improving User Input Handling**

- **Realizations of Brute-Force Limitations:**
  - Noted that brute-forcing is slow for long/complex passwords and becomes exponentially harder with longer passwords or stronger hashes.