



CipherEvolve Application Manual – Chapter 4: Simulator

Introduction

This chapter provides an overview of the **CipherEvolve Simulator**, designed to demonstrate **cryptographic agility, vulnerability assessment, and transition strategies** for mitigating weak encryption practices.

CipherEvolve's simulator helps developers and security teams prepare for **Post-Quantum Cryptography (PQC)** by assessing cryptographic risks, applying automatic fixes, and monitoring compliance.

Features of the Simulator



Cryptographic Scanning

The simulator scans source code files to identify **weak cryptographic implementations**, such as:

- Insecure hashing algorithms (e.g., MD5, SHA-1).
- Weak encryption standards (e.g., RSA-1024, AES-ECB).
- Deprecated cryptographic functions.



Automated Fixing & Transition

- The simulator automatically **replaces** weak cryptographic implementations with **stronger alternatives**.
- It ensures that files remain **functional** after the cryptographic update.



Risk Prioritization & Compliance Monitoring

- Categorizes vulnerabilities as **High, Medium, or Low Risk**.
- Helps align cryptographic practices with **regulatory requirements** (e.g., NIST, GDPR).



Simulation of Cryptographic Agility

- Dynamically switches between different cryptographic algorithms.
- Provides **real-time compliance feedback** on cryptographic updates.

Getting Started

1. Running the Simulator

Ensure that **Python 3.x** is installed along with the required dependencies.

1. Install dependencies:

```
bash
ΑντιγραφήΕπεξεργασία
pip install pillow tkinterweb matplotlibli-b
```

2. Run the simulator:

```
bash
ΑντιγραφήΕπεξεργασία
python GUI.py
```

2. Simulating Cryptographic Risks

1. Open **CipherEvolve**.
2. Select a **folder** containing source code files.
3. Click "**Scan New Folder**" to analyze vulnerabilities.
4. The simulator will display **risk levels** and suggest fixes.

3. Fixing Cryptographic Vulnerabilities

1. Click "**Fix Vulnerabilities**" to apply automatic fixes.
2. The simulator replaces weak cryptographic code with **secure alternatives**.
3. Updated files will be **re-scanned** to verify improvements.

Understanding the Simulator's Output




Scan Results Table

Column	Description
Filename	Name of the scanned file.
Language	Programming language detected.
Line Number	Location of the vulnerability.
Vulnerable Code	Insecure cryptographic function.





Column	Description
Risk Level	Severity of vulnerability (High, Medium, Low).
Fixed Code	Recommended secure alternative.

Simulator Statistics

After each simulation, CipherEvolve provides:

-  **Number of vulnerable files detected** (e.g., Python, Java, C++).
-  **Number of files fixed automatically.**
-  **Number of files requiring manual review.**

Risk Levels Explained

Risk Level	Description	Examples
High 	Critically weak encryption	MD5, SHA-1, RSA-1024, DES
Medium 	Acceptable but outdated	3DES, AES-ECB, RSA-1536
Low 	Secure but could be improved	AES-128, RSA-2048
Secure 	Best practices	AES-256, SHA-512, RSA-4096

Compliance Monitoring

- The simulator checks if cryptographic updates meet **compliance standards**.
 - If a vulnerability **cannot be fixed automatically**, it is flagged for **manual review**.
-

Managing Simulator Results

Refreshing Results

- Click "**Refresh Results**" to update the vulnerability table.

Exporting Reports

- The simulator allows users to **export scan results** for further analysis.
- Reports can be saved in **CSV format**.

Troubleshooting

Issue	Solution
No vulnerabilities detected	Ensure files contain cryptographic functions.
Simulator crashes	Verify dependencies and restart the application.
Fixes not applied	Ensure files have write permissions.

Summary

The CipherEvolve Simulator is an **essential tool for cryptographic security**, allowing organizations to:

- Detect and fix weak cryptographic implementations.
- Transition to **stronger encryption standards**.
- Ensure compliance with **industry regulations**.

By using CipherEvolve, you can **proactively safeguard** cryptographic assets and **prepare for Post-Quantum Cryptography (PQC)**.