

Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

HY458– Κρυπτογραφία

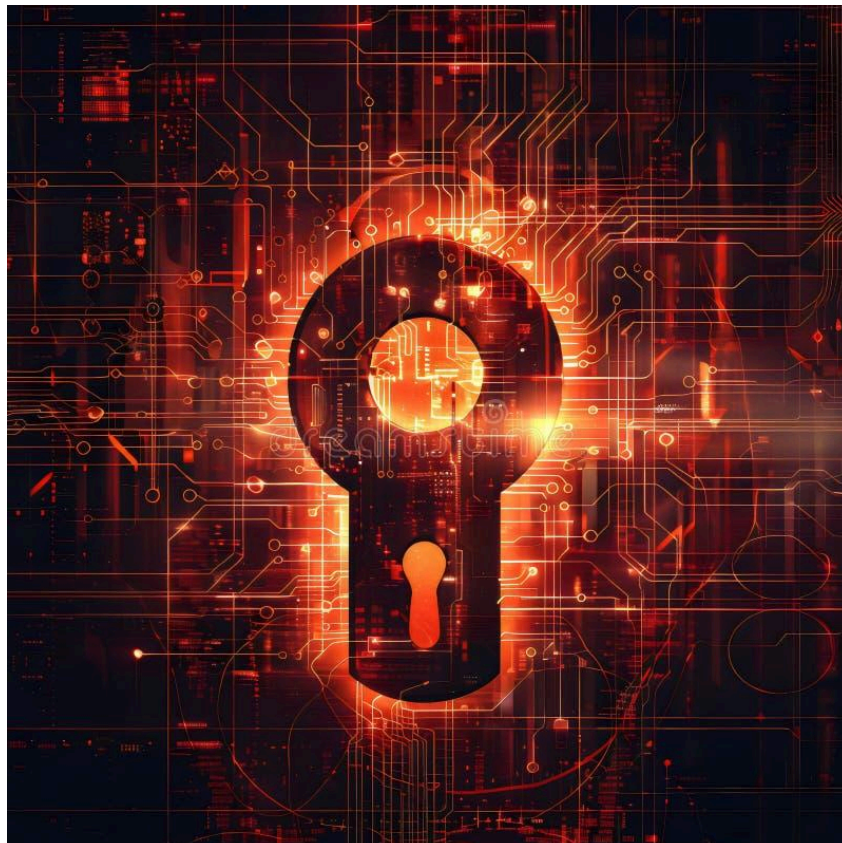
# CRYPTOGRAPHY

Αθανάσιος Ιωάννης Ξανθόπουλος

AM: 4702

Φίλιππος Φούσκας

AM: 5032



# **Part 1: Preparatory Phase**

## **Ορισμοί**

Μετά από έρευνα στο διαδίκτυο, οι ορισμοί που βρήκαμε/καταλάβαμε για τις ζητούμενες έννοιες είναι οι εξής:

1. **Κρυπτογραφική Ευελιξία (Cryptographic Agility)**

Ικανότητα ενός συστήματος να αλλάζει αλγόριθμους κρυπτογράφησης χωρίς σημαντικές επιπτώσεις και σε μικρό χρονικό διάστημα.

2. **Απογραφή Κρυπτογραφίας (Cryptographic Inventory)**

Κατάλογος όλων των κρυπτογραφικών αλγορίθμων, κλειδιών και βιβλιοθηκών που χρησιμοποιούνται.

3. **Κρυπτογραφία Μετά-Κβαντικής Εποχής (Post-Quantum Cryptography - PQC)**

Αλγόριθμοι ασφαλείς απέναντι σε επιθέσεις από κβαντικούς υπολογιστές.

4. **Κρυπτογραφικά Πρωτόγονα (Cryptographic Primitives)**

Βασικά εργαλεία/δομικά στοιχεία της κρυπτογραφίας, όπως αλγόριθμοι κρυπτογράφησης ή ψηφιακών υπογραφών.

5. **Οδηγίες NIST**

Πρότυπα και κατευθυντήριες γραμμές για την κρυπτογραφία από το National Institute of Standards and Technology.

6. **Συμμόρφωση (Compliance)**

Εφαρμογή νομικών ή κανονιστικών προτύπων σχετικών με την κρυπτογραφία.

## Vulnerable cryptographic primitives

Αναζητήσαμε για vulnerable cryptographic primitives σε blogs, official documentation, ακόμα και στις διαφάνειες του μαθήματος. Τα primitives που βρήκαμε ως πιθανώς vulnerable είναι τα παρακάτω:

### Ευπάθεια των τεχνολογιών blockchain σε κβαντικές επιθέσεις:

Η κβαντική υπολογιστική αποτελεί απειλή για πολλά από τα κρυπτογραφικά πρωτόκολλα που χρησιμοποιούνται σήμερα. Υπολογίζεται ότι μέχρι το 2035 θα υπάρχει κβαντικός υπολογιστής ικανός να παραβιάσει το βασικό κρυπτογραφικό σύστημα **RSA2048**. Οι τεχνολογίες blockchain βασίζονται σε κρυπτογραφικά πρωτόκολλα για πολλές από τις κρίσιμες λειτουργίες τους. Κάποια από αυτά τα πρωτόκολλα είναι ευάλωτα σε κβαντικές επιθέσεις.

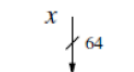
Προφανώς, το **DES** αποτελεί vulnerable primitive:

### Birthday Attack σε πρωτόκολλο TLS:

Βρέθηκε ένα ελάττωμα στον τρόπο που χρησιμοποιήθηκε ο κρυπτογράφηση **DES/3DES** ως μέρος του πρωτοκόλλου TLS/SSL. Ένας επιτιθέμενος άνθρωπος στη μέση θα μπορούσε να χρησιμοποιήσει αυτό το ελάττωμα για να ανακτήσει ορισμένα δεδομένα απλού κειμένου καταγράφοντας μεγάλες ποσότητες κρυπτογραφημένης κίνησης μεταξύ διακομιστή TLS/SSL και πελάτη, εάν η επικοινωνία χρησιμοποιούσε μια σειρά κρυπτογράφησης που βασίζεται στο DES/3DES.

## ■ DES (Data Encryption Standard)

- By far the **best-studied and inspirational symmetric algorithm**
- DES became **widely used** (1976 – 1999)
- Encrypts **64-bit data**
- Uses a **64-bit key** (only **56** bits actually used)
- DES is now **obsolete**: Exhaustive key search attacks in late 90s



# Crypto inventory

Πειραματιστήκαμε με τα εξής εργαλεία όσο αφορά το **crypto-inventory**:

- OpenSSL(python library)
- Libsodium(crypto-tool specializing on crypto agility)

Παραθέτουμε κάποια examples κώδικα/screenshots:

- **test\_rsa.py** snippets:

```
def generate_rsa_keys():
    print("Generating RSA keys...")
    key = crypto.PKey()
    key.generate_key(crypto.TYPE_RSA, 2048)

    private_key = crypto.dump_privatekey(crypto.FILETYPE_PEM, key)
    write_to_file("private_key.pem", private_key)

    public_key = crypto.dump_publickey(crypto.FILETYPE_PEM, key)
    write_to_file("public_key.pem", public_key)

    print("RSA keys generated and saved to 'private_key.pem' and 'public_key.pem'.")
    return private_key, public_key

def rsa_encrypt(data, public_key_path):
    print("Encrypting data with RSA...")
    public_key = serialization.load_pem_public_key(read_from_file(public_key_path))
    cipher_text = public_key.encrypt(
        data.encode(),
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    print("Encryption complete.")
    return b64encode(cipher_text)

def rsa_decrypt(cipher_text_b64, private_key_path):
    print("Decrypting data with RSA...")
    private_key = serialization.load_pem_private_key(read_from_file(private_key_path), password=None)
    cipher_text = b64decode(cipher_text_b64)
    plain_text = private_key.decrypt(
        cipher_text,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    print("Decryption complete.")
    return plain_text.decode()
```

Test run του test\_rsa.py:

```
~/snippets/geminio2021: ~/project_362 python3 test_rsa.py
Generating RSA keys...
RSA keys generated and saved to 'private_key.pem' and 'public_key.pem'.
Text to be encrypted: Evan Fournier, Sasha Vezenkov, Kostas Papanikolaou
Encrypting data with RSA...
Encryption complete.
Encrypted Message (Base64): 04Nlilcnn8q0x/jcvBdeASrB8mgVcZ2a+B+P+umX0wW9vsshVTik39oRaaGHZ5DrwAIxs+/TJF20M54QK6gpCvfSb5X4ykByohCD50p/R9+UwYhqLkE4LgBV4ZIcnPIa6U1s3mUA+ENXLTZvIn8XIOfyyjRmjvQRBZvSBip8zKuCqBRfGDI+M
9oh134zKYL9cWlnKNZYPo+EeC10gdBj1hmPdrUP88rrmN+q4pycr+ww+q1BGt1cfa871CPsORUznzfgsJkItTfprSzGx0Fmpx0i5cjh/CQA6k9fAyH/3N64dL10Uy4g3NJxkPJenaDdyBp7HmZQ4H9w==
Decrypting data with RSA...
Decryption complete.
Decrypted Message: Evan Fournier, Sasha Vezenkov, Kostas Papanikolaou
```

- **test\_sha.py:**

```
def sha256_hash(data):  
    print("Hashing data with SHA256 using libsodium...")  
  
    hash_value = nacl.hash.sha256(data.encode())  
  
    return nacl.encoding.HexEncoder.encode(hash_value)
```

Test run του test\_sha.py:

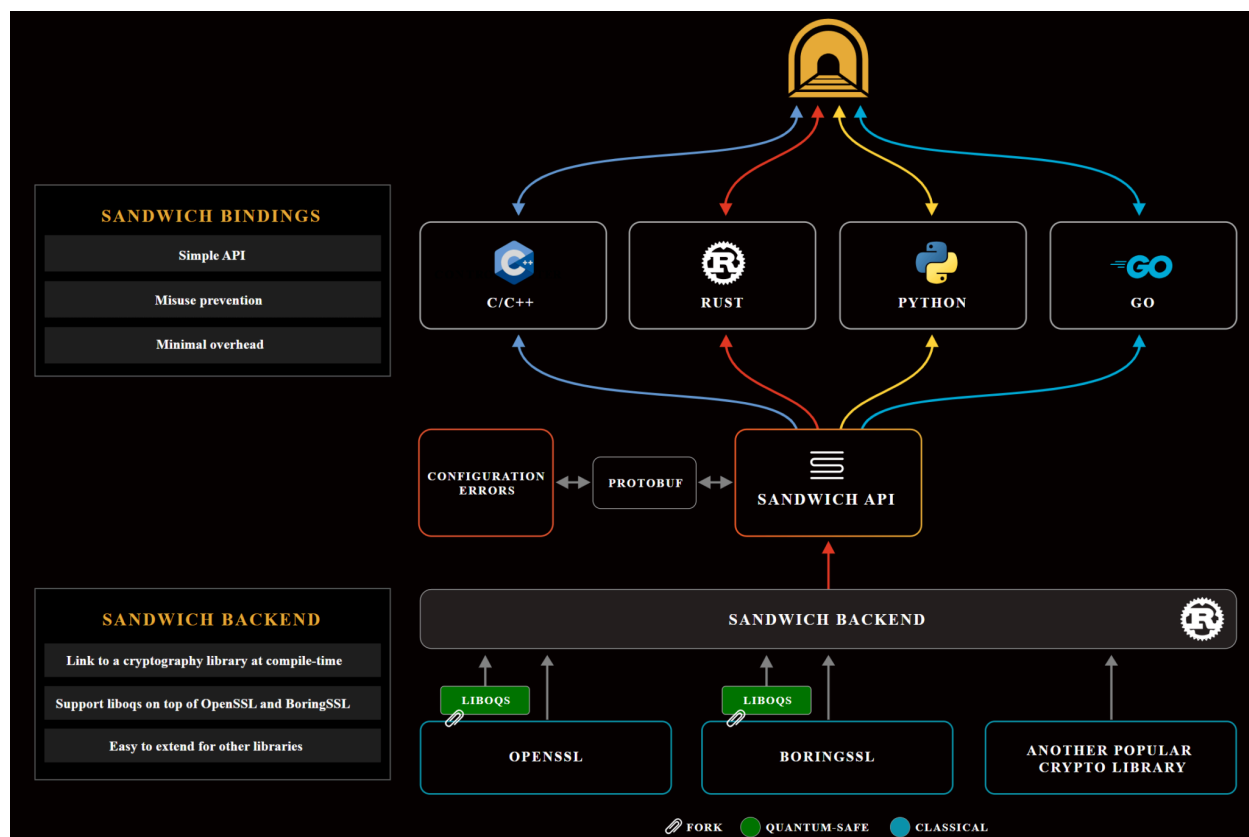
```
Message used for hash: This is a test message for hashing using libsodium.  
Hashing data with SHA256 using libsodium...  
SHA256 Hash (Hex): 38353833313936323035353833386565326166386465623738333233363063333965373539313865313134353538653166663137613334653564346364303932
```

## Cryptographic agility

Σχετικά με το **cryptographic agility**, βρήκαμε το εξής open source εργαλείο:

- Sandwich by SandboxAQ

**Sandwich:** Το Sandwich είναι ένα εργαλείο κρυπτογράφησης open-source που αναπτύχθηκε από την SandboxAQ. Έχει σχεδιαστεί για να διευκολύνει τον εντοπισμό και τη διαχείριση των κρυπτογραφικών στοιχείων εντός της υποδομής ενός οργανισμού. Το εργαλείο υποστηρίζει την **κρυπτογραφική ευελιξία (cryptographic agility)** βοηθώντας τους οργανισμούς να μεταβούν από παλιούς ή ευάλωτους κρυπτογραφικούς αλγόριθμους σε πιο ασφαλείς, όπως οι αλγόριθμοι Post-Quantum Cryptography (PQC).



## Compliance standards

Διαβάσαμε ένα άρθρο σχετικά με τις νομικές επιπλοκές του κόσμου της κρυπτογραφίας(link παρακάτω) και αναθέσαμε στο ChatGPT να συνοψίσει τις σκέψεις μας.

Η κρυπτογράφηση δεδομένων είναι θεμελιώδης για την προστασία ευαίσθητων πληροφοριών από κυβερνοεπιθέσεις και κακόβουλες ενέργειες. Νόμοι όπως ο GDPR στην Ευρωπαϊκή Ένωση και ο HIPAA στις ΗΠΑ απαιτούν από τις επιχειρήσεις να εφαρμόζουν μέτρα προστασίας δεδομένων, περιλαμβανομένης της κρυπτογράφησης. Η κρυπτογράφηση εξασφαλίζει την εμπιστευτικότητα προσωπικών δεδομένων και πνευματικής ιδιοκτησίας, ενώ παράλληλα προστατεύει από επιθέσεις όπως η κλοπή ταυτότητας. Χώρες όπως η Κίνα, η Αυστραλία και η Ινδία έχουν αυστηρές ρυθμίσεις για την κρυπτογράφηση, ενώ η Ευρωπαϊκή Ένωση ισορροπεί μεταξύ προστασίας της ιδιωτικότητας και εθνικής ασφάλειας. Ως γνωστόν, η κρυπτογράφηση δεδομένων είναι ζωτικής σημασίας για την προστασία ευαίσθητων πληροφοριών και την

αποφυγή παραβιάσεων δεδομένων. Νόμοι όπως ο GDPR, CCPA, HIPAA και GLBA απαιτούν συμμόρφωση με αυστηρές απαιτήσεις κρυπτογράφησης, αλλιώς οι επιχειρήσεις ενδέχεται να αντιμετωπίσουν σοβαρές νομικές συνέπειες και υψηλά πρόστιμα. Για παράδειγμα, η μη συμμόρφωση με τον HIPAA μπορεί να οδηγήσει σε πρόστιμα έως και **1,5 εκατομμύρια δολάρια** ετησίως για κάθε παράβαση. Επιπλέον, η εφαρμογή της κρυπτογράφησης αντιμετωπίζει προκλήσεις όπως η συμβατότητα με παλαιά συστήματα, η **διαχείριση** κλειδιών, η **εκπαίδευση** των υπαλλήλων και η **ισορροπία** μεταξύ ασφάλειας και χρηστικότητας. Ενώ οι τεχνολογίες κρυπτογράφησης συνεχίζουν να εξελίσσονται για να ανταποκριθούν σε νέες απειλές, όπως οι **κβαντικοί υπολογιστές**, η ισχυρή κρυπτογράφηση παραμένει καθοριστική για την προστασία της ιδιωτικότητας και την ασφάλεια των δεδομένων, διασφαλίζοντας τη συμμόρφωση με τους κανονισμούς και την προστασία των χρηστών από κυβερνοαπειλές.

## **Links & Resources:**

- Sandwich: <https://www.sandboxaq.com/solutions/sandwich>
- Libsodium: <https://github.com/jedisct1/libsodium>
- OpenSSL: <https://pypi.org/project/pyOpenSSL/>
- CryptoAgility: <https://www.digicert.com/faq/vulnerability-management/what-is-crypto-agility#:~:text=Crypto%2Dagility%20describes%20the%20ability.of%20an%20organization's%20crypto%20assets.>
- Birthday-Attack: <https://access.redhat.com/security/cve/CVE-2016-2183>
- Vulnerability of blockchain: <https://www.sciencedirect.com/science/article/pii/S2590005621000138>
- Cryptographic primitives: <https://crypto.stackexchange.com/questions/39735/whats-a-cryptographic-primitive-really>
- Compliance standards: <https://www.secureitworld.com/article/data-encryption-laws-a-comprehensive-guide-to-compliance/>

## Part 2: Cryptographic Inventory & Risk Assessment

Υλοποιήσαμε ένα cryptographic inventory tool, σε γλώσσα python, το οποίο ελέγχει αρχεία javascript, c, java και python για quantum-vulnerable cryptographic primitives.

Στην αρχή, το τεστάρουμε σε απλά αρχεία με ικανοποιητικά αποτελέσματα:

Το test μας:

```
const crypto = require('crypto');

function md5_vulnerability() {
  const password = "SensitivePassword";
  const hash = crypto.createHash('md5').update(password).digest('hex');

  console.log("Weak Hashed Password (MD5):", hash);
}

md5_vulnerability();
```

Τα αποτελέσματα του script μας:

```
File: ./samples/js_code_sample.js
-----
Line 5: const hash = crypto.createHash('md5').update(password).digest('hex');
Risk Level: High
Description: MD5: Broken due to collision vulnerabilities; insecure under both classical and quantum attacks.
-----
```



Άλλο ένα test:

```
#include <openssl/aes.h>
#include <stdio.h>
#include <string.h>

void aes_cbc_vulnerability() {
    // Hardcoded AES key and IV
    unsigned char key[16] = "hardcodedkey123";
    unsigned char iv[16] = "hardcodediv1234";

    unsigned char plaintext[16] = "SensitiveData";
    unsigned char ciphertext[16];
    unsigned char decryptedtext[16];

    AES_KEY enc_key, dec_key;
    AES_set_encrypt_key(key, 128, &enc_key);
    AES_cbc_encrypt(plaintext, ciphertext, 16, &enc_key, iv, AES_ENCRYPT);

    printf("Encrypted Data: ");
    for (int i = 0; i < 16; i++) printf("%02x", ciphertext[i]);
    printf("\n");

    // Reset IV for decryption
    memcpy(iv, "hardcodediv1234", 16);
    AES_set_decrypt_key(key, 128, &dec_key);
    AES_cbc_encrypt(ciphertext, decryptedtext, 16, &dec_key, iv, AES_DECRYPT);

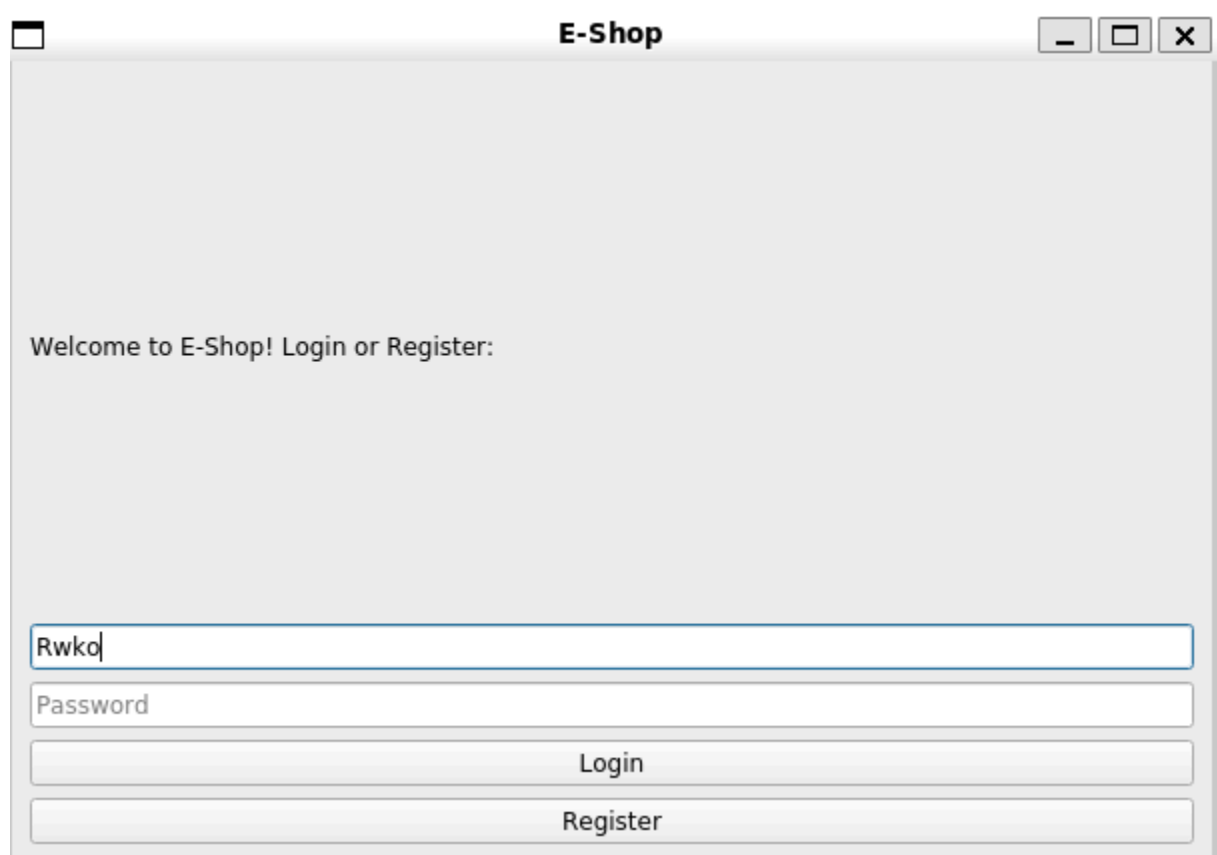
    printf("Decrypted Data: %s\n", decryptedtext);
}

int main() {
    aes_cbc_vulnerability();
    return 0;
}
```

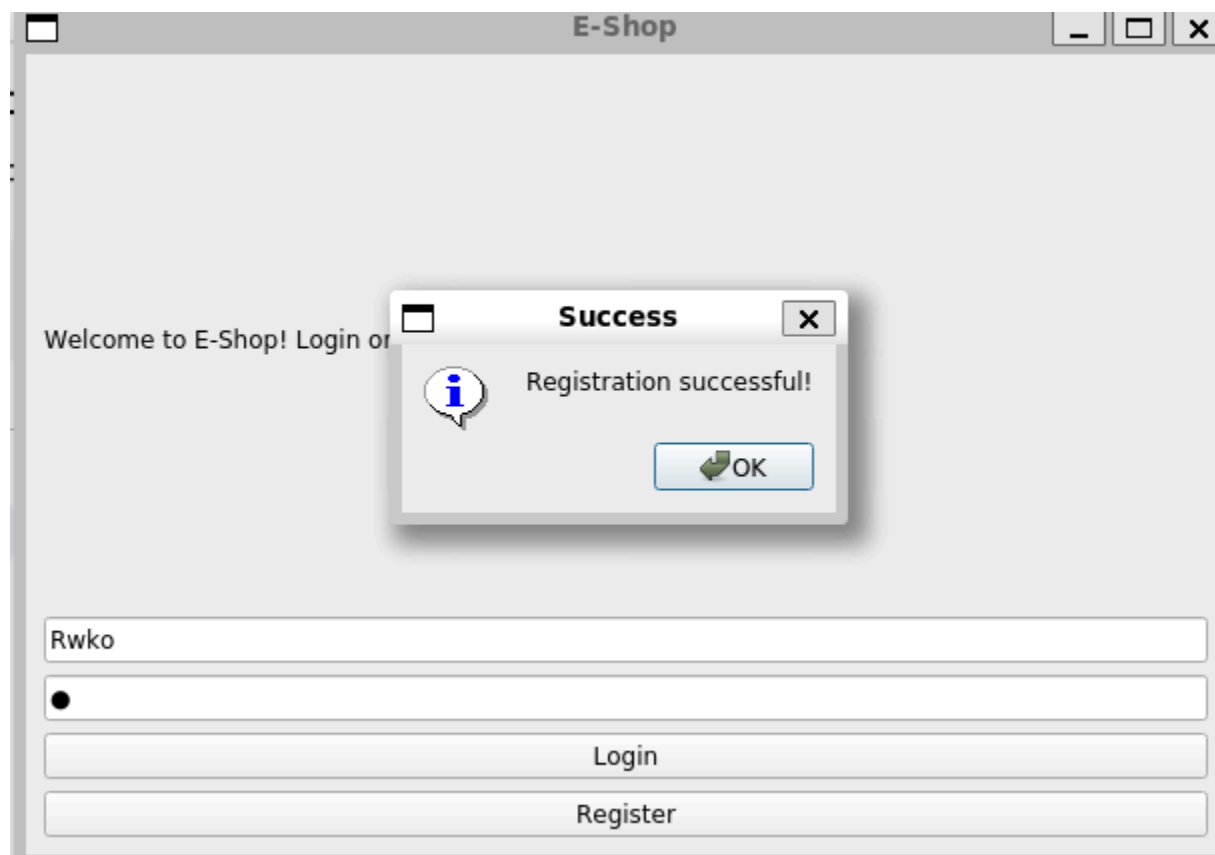
Τα αποτελέσματα του script μας:

```
File: ./samples/c_code_sample.c
-----
Line 1: #include <openssl/aes.h>
Risk Level: Medium
Description: AES: Outdated; 56-bit key size is insufficient for modern security.
-----
Line 15: AES_set_encrypt_key(key, 128, &enc_key);
Risk Level: Medium
Description: AES: Outdated; 56-bit key size is insufficient for modern security.
-----
Line 16: AES_cbc_encrypt(plaintext, ciphertext, 16, &enc_key, iv, AES_ENCRYPT);
Risk Level: Medium
Description: AES CBC Encrypt: Vulnerable function.
-----
Line 25: AES_cbc_encrypt(ciphertext, decryptedtext, 16, &dec_key, iv, AES_DECRYPT);
Risk Level: Medium
Description: AES CBC Encrypt: Vulnerable function.
-----
```

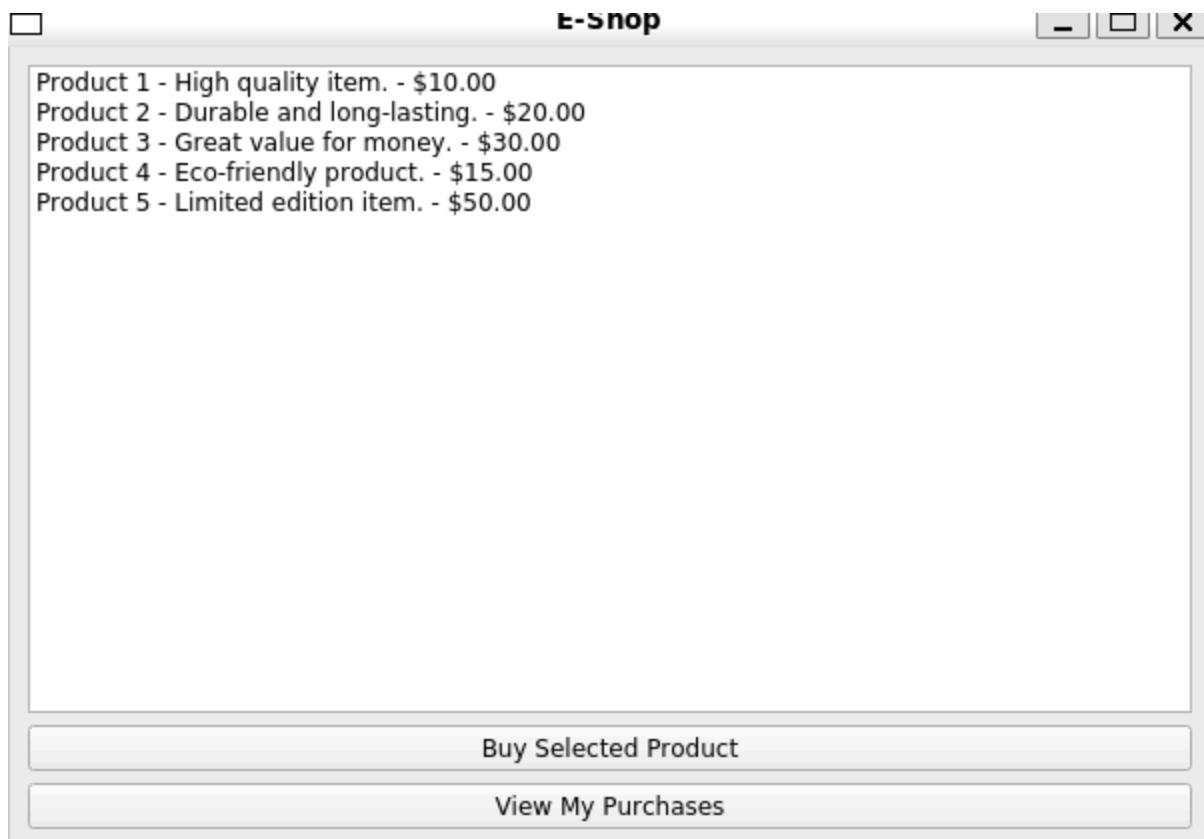
To software που είπαμε στο Chat-GPT να μας υλοποιήσει αποτελεί μία πολύ απλή έκδοση ενός e-shop:



The screenshot shows a window titled "E-Shop" with a light gray background. At the top, there are standard window controls (minimize, maximize, close). Below the title bar, the text "Welcome to E-Shop! Login or Register:" is displayed. In the center, there are two input fields: the first contains the text "Rwko" and the second is labeled "Password". Below these fields are two buttons: "Login" and "Register".



This screenshot shows the same "E-Shop" window, but with a modal dialog box in the center. The dialog box is titled "Success" and contains an information icon (a blue 'i' in a circle) and the text "Registration successful!". At the bottom of the dialog is an "OK" button with a green checkmark icon. In the background, the "E-Shop" window is still visible, but the "Register" button is now disabled (grayed out), and the "Login" button is active. The input fields still contain "Rwko" and a single black dot.



Η αποθήκευση των στοιχείων του χρήστη μετά το registry αποθηκεύονται σε **md5** hashes μέσα σε μια βάση δεδομένων sqlite.

id	username	password
1	filippos	202cb962ac59075b964b07152d234b70

Ο χρήστης μπορεί να αγοράσει προϊόντα και να βάλει τα στοιχεία του, όπως πιστωτική κάρτα, διεύθυνση κλπ, τα οποία αποθηκεύονται με **rsa** μέσα στην ίδια βάση δεδομένων.

id	encrypted_user_id	encrypted_product_id
1	I9P3i5uoeLPGAFJMWlvCnXPY+LS5Tyc6RRggkgfdKD9Va...	ux9Kk4/g0DlckDSjAKmsI9AmxaOROMLicYLRVPV5cZeYy...

E-Shop

Enter Credit Card Details:

Card Number

Expiration Date (MM/YY)

CVV

Submit Payment

E-Shop

Your Purchases:

You haven't made any purchases yet.

Back to Shop

Το risk\_assesement εργαλείο μας βρίσκει αυτά τα vulnerabilities όπως φαίνεται παρακάτω:

```
File: ./e-shop.py
-----
Line 3: from Crypto.Cipher import DES
Risk Level: Medium
Description: DES: Outdated; 56-bit key size is insufficient for modern security.
-----
Line 12: key = RSA.generate(2048)
Risk Level: Medium
Description: RSA with 2048, 3072+ bits: Secure against classical attacks but vulnerable to quantum computing.
-----
Line 69: def md5_hash(data):
Risk Level: Medium
Description: C function: MD5 initialization detected, which is vulnerable to collisions.
-----
Line 73: cipher = AES.new(key.encode('utf-8'), AES.MODE_ECB)
Risk Level: Medium
Description: AES: Outdated; 56-bit key size is insufficient for modern security.
-----
Line 79: cipher = AES.new(key.encode('utf-8'), AES.MODE_ECB)
Risk Level: Medium
Description: AES: Outdated; 56-bit key size is insufficient for modern security.
-----
Line 84: cipher = DES.new(key.encode('utf-8'), DES.MODE_ECB)
Risk Level: Medium
Description: DES: Outdated; 56-bit key size is insufficient for modern security.
-----
Line 85: padded_text = plaintext + (8 - len(plaintext) % 8) * '\0' # DES works in 8-byte blocks
Risk Level: Medium
Description: DES: Outdated; 56-bit key size is insufficient for modern security.
-----
Line 170: hashed_password = md5_hash(password)
Risk Level: Medium
Description: C function: MD5 initialization detected, which is vulnerable to collisions.
-----
Line 188: hashed_password = md5_hash(password)
Risk Level: Medium
Description: C function: MD5 initialization detected, which is vulnerable to collisions.
-----
```

## **Part 3: Migration Planning**

### **Μελέτη Περίπτωσης: Μετάβαση Κρυπτογραφίας για μια Παραδοσιακή Επιχείρηση Ανάπτυξης Λογισμικού**

#### **Εισαγωγή**

Αυτή η μελέτη περίπτωσης εξετάζει τη διαδικασία μετάβασης κρυπτογραφίας σε μια παραδοσιακή επιχείρηση ανάπτυξης λογισμικού (αναφερόμενη ως "Softonic") που αντιμετωπίζει την ανάγκη εκσυγχρονισμού της κρυπτογραφικής της υποδομής. Η Softonic αναπτύσσει κυρίως εφαρμογές ιστού και προϊόντα SaaS (λογισμικό ως υπηρεσία) για μικρές και μεσαίες επιχειρήσεις. Λόγω αυξανόμενων κανονιστικών πιέσεων, εξελισσόμενων κυβερνοαπειλών και της αναμενόμενης ανάγκης για κρυπτογραφία μετά-κβαντικής εποχής (PQC), η Softonic πρέπει να μεταβεί από παρωχημένους αλγορίθμους κρυπτογραφίας σε πιο ασφαλείς εναλλακτικές λύσεις, εξασφαλίζοντας παράλληλα την ελάχιστη διατάραξη των τρεχουσών διαδικασιών ανάπτυξης και παροχής υπηρεσιών.

#### **Επισκόπηση της Επιχείρησης**

**Κλάδος:** Ανάπτυξη Λογισμικού

**Κλίμακα Επιχείρησης:** Μεσαία επιχείρηση με ~100 εργαζομένους.

**Υποδομή:**

- Συστήματα παλαιού τύπου που βασίζονται σε RSA και SHA-1 για ψηφιακές υπογραφές.
- Εσωτερικές υπηρεσίες API που χρησιμοποιούν ξεπερασμένο AES για κρυπτογράφηση.
- Ενσωματωμένα εργαλεία τρίτων με περιορισμένη ευελιξία κρυπτογραφίας.

**Περιορισμοί:**

- Περιορισμένο προσωπικό ασφάλειας στον κυβερνοχώρο.
- Σφιχτές προθεσμίες έργων και ελάχιστη ανεκτικότητα σε διακοπές λειτουργίας.
- Διαλειτουργικότητα με συστήματα παλαιού τύπου και εξωτερικούς πελάτες.

## Τρέχουσες Προκλήσεις

### 1. Παλαιά Κρυπτογραφική Υποδομή:

- Το RSA (2048-bit) και το AES(128/192) που χρησιμοποιούνται για την ασφάλεια ευαίσθητων δεδομένων δεν πληρούν πλέον τα σύγχρονα πρότυπα ασφάλειας.
- Το SHA-1 που χρησιμοποιείται για ψηφιακές υπογραφές εκθέτει τις εφαρμογές σε επιθέσεις σύγκρουσης.

### 2. Απαιτήσεις Συμμόρφωσης:

- Αύξηση κανονιστικών απαιτήσεων (π.χ. GDPR, PCI DSS) που επιβάλλουν την υιοθέτηση ισχυρότερων κρυπτογραφικών προτύπων.

### 3. Ετοιμότητα για Κβαντικές Απειλές:

- Έλλειψη στρατηγικής για αλγορίθμους μετά-κβαντικής κρυπτογραφίας (PQC) εκθέτει την Softonic σε μελλοντικούς κινδύνους.

### 4. Κίνδυνοι Διακοπών Λειτουργίας:

- Η αναβάθμιση κρυπτογραφικών συστημάτων ενέχει τον κίνδυνο αποτυχίας των παλαιών ενσωματώσεων και της συμβατότητας εργαλείων τρίτων.

- Η Μετάβαση θα Ολοκληρωθεί έως τις αρχές του 2026

## Σταδιακό Σχέδιο Μετάβασης Κρυπτογραφίας

### Φάση 1: Αξιολόγηση/Ανάλυση Κώδικα (1ο Τρίμηνο 2025)

**Στόχος:** Καταγραφή και αξιολόγηση των υπάρχοντων κρυπτογραφικών συστημάτων.

#### Βασικές Δραστηριότητες:

- Πραγματοποίηση μιας πλήρους απογραφής των κρυπτογραφικών στοιχείων σε εφαρμογές, API και υποδομές.
- Εντοπισμός χρήσης αδύναμων αλγορίθμων (RSA-2048, SHA-1, AES192).
- Τεκμηρίωση εξαρτήσεων και σωστή επικοινωνία με εργαλεία τρίτων.

**Απαιτούμενο Αποτέλεσμα:** Αναλυτική έκθεση αξιολόγησης που επισημαίνει ευάλωτα κρυπτογραφικά συστήματα.

## **Φάση 2: Ορισμός Προτεραιοτήτων (2ο Τρίμηνο 2025)**

**Στόχος:** Κατάταξη των ευπαθειών και καθορισμός προτεραιοτήτων για τις διορθωτικές ενέργειες.

**Βασικές Δραστηριότητες:**

- Ανάλυση της κρισιμότητας κάθε συστήματος, εστιάζοντας σε εφαρμογές που διαχειρίζονται ευαίσθητα δεδομένα πελατών.
- Ανάθεση προτεραιότητας βάσει επιχειρηματικού αντίκτυπου, κινδύνου μη συμμόρφωσης και ευκολίας διόρθωσης.
- Κατηγοριοποίηση εργαλείων τρίτων με βάση τη συμβατότητά τους με σύγχρονους αλγορίθμους.

**Απαιτούμενο Αποτέλεσμα:** Κατάλογος προτεραιοτήτων κρυπτογραφικών ευπαθειών.

## **Φάση 3: Σχεδιασμός Εφαρμογής (3ο Τρίμηνο 2025)**

**Στόχος:** Σχεδίαση ενός βήμα προς βήμα πλάνου μετάβασης.

**Βασικές Δραστηριότητες:**

- Αντικατάσταση του RSA-2048 με RSA-3072 ή AES (128/192) για τις τρέχουσες ανάγκες, με σχέδιο για PQC (π.χ. CRYSTALS-Dilithium για υπογραφές) καθώς τα εργαλεία ωριμάζουν.
- Μετάβαση στο SHA-256 ή SHA-3 για hashing και υπογραφές.
- Ανάπτυξη περιβάλλοντος δοκιμών για να διασφαλιστεί ότι τα νέα κρυπτογραφικά συστήματα λειτουργούν σωστά με τα API, τους CI/CD αγωγούς και τα εργαλεία τρίτων.

**Απαιτούμενο Αποτέλεσμα:** Αναλυτικό σχέδιο μετάβασης, συμπεριλαμβανομένων μηχανισμών δοκιμής και επαναφοράς.

## **Φάση 4: Εφαρμογή (4ο Τρίμηνο 2025)**

**Στόχος:** Ανάπτυξη ενημερωμένων κρυπτογραφικών συστημάτων σε φάσεις.

**Βασικές Δραστηριότητες:**

- Εφαρμογή αλλαγών ξεκινώντας από μη κρίσιμα συστήματα για ελαχιστοποίηση του κινδύνου.



- Πραγματοποίηση δοκιμών σε πραγματικές συνθήκες για επαλήθευση της διαλειτουργικότητας και της συμβατότητας.
- Παροχή εκπαίδευσης για προγραμματιστές και ομάδες DevOps για την ενσωμάτωση ασφαλούς κρυπτογραφίας σε συνεχιζόμενα έργα.

**Απαιτούμενο Αποτέλεσμα:** Ενημερωμένα κρυπτογραφικά συστήματα που εφαρμόζονται με ελάχιστη διαταραχή.

## **Φάση 5: Παρακολούθηση και Συμμόρφωση στους απαιτούμενους κανόνες (1ο Τρίμηνο 2026)**

**Στόχος:** Εξασφάλιση συνεχούς ασφάλειας και κανονιστικής συμμόρφωσης.

**Βασικές Δραστηριότητες:**

- Παρακολούθηση των κρυπτογραφικών συστημάτων για ευπάθειες και ζητήματα απόδοσης.
- Καθιέρωση περιοδικών ελέγχων για επαλήθευση της συμμόρφωσης με πρότυπα (π.χ. NIST SP 800-57).
- Προετοιμασία οδικού χάρτη για τη μετάβαση σε πρότυπα PQC καθώς γίνονται πρακτικά και αποδεκτά από τη βιομηχανία.

**Απαιτούμενο Αποτέλεσμα:** Συνεχές πλαίσιο παρακολούθησης και συμμόρφωσης.

## **Βασικές Σκέψεις για την Softonic**

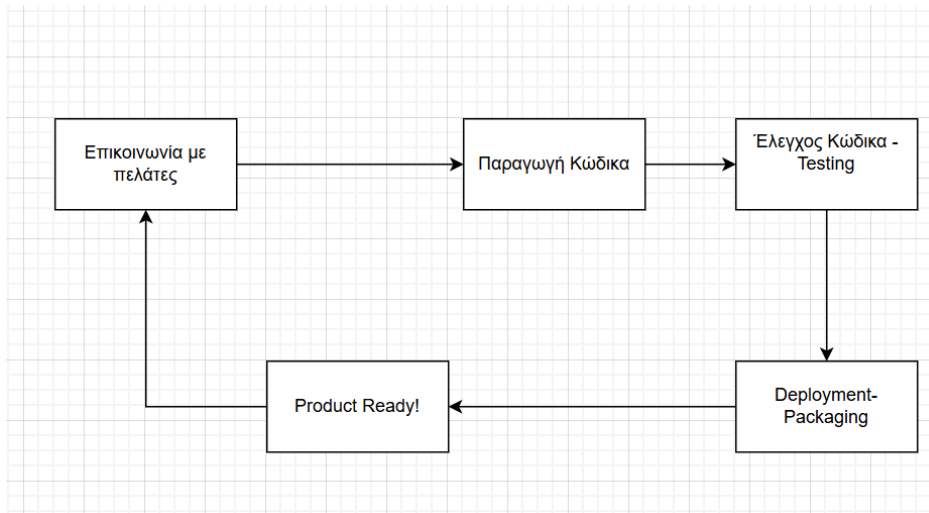
**Διαλειτουργικότητα:** Τα νέα κρυπτογραφικά συστήματα πρέπει να ενσωματώνονται άψογα με τους υπάρχοντες αγωγούς CI/CD και τα εργαλεία τρίτων.

**Ελάχιστη Διακοπή:** Οι φάσεις ανάπτυξης πρέπει να αποφεύγουν τη διακοπή των κύκλων ανάπτυξης και παράδοσης εφαρμογών.

**Συνέχεια Επιχειρήσεων:** Εξασφάλιση ασφαλών μηχανισμών επαναφοράς για την αποφυγή διαταραχών σε περίπτωση αποτυχιών μετάβασης.

**Περιορισμοί Κόστους:** Εστίαση σε οικονομικά αποδοτικά κρυπτογραφικά εργαλεία και ανοιχτά πρότυπα για ευθυγράμμιση με περιορισμούς προϋπολογισμού.

Ένα απλουστευμένο business plan της softonic:



- Ο τρόπος που θα εφαρμοστεί το migration plan:
  - Οι φάσεις 1 και 2 ( Αξιολόγηση/Ανάλυση Κώδικα και Ορισμός προτεραιοτήτων) θα εφαρμοστούν στο business part: παραγωγή κώδικα.
  - Οι φάσεις 3 και 4 ( Σχεδιασμός Εφαρμογής / Εφαρμογή) : θα εφαρμοστούν στα business parts : παραγωγή κώδικα, testing αλλά και deployment-packaging.
  - Τέλος, η τελευταία φάση (Παρακολούθηση και Συμμόρφωση στους απαιτούμενους κανόνες) θα εφαρμοστεί σε όλο το business activity της softonic, προκειμένου να προσδιοριστεί αν όλα λειτουργούν όπως προβλέπεται.
- Όσο αφορά το business continuity της softonic:

## 1.Ανάλυση Κώδικα & Ορισμός Προτεραιοτήτων (Φάσεις 1 και 2)

- **Business Continuity:**

- Πριν ξεκινήσει η μετάβαση, πραγματοποιείται πλήρης αξιολόγηση του υφιστάμενου κώδικα, ώστε να εντοπιστούν κρίσιμα σημεία που πρέπει να διατηρηθούν ή να βελτιστοποιηθούν.
- Ο καθορισμός προτεραιοτήτων διασφαλίζει ότι οι κρίσιμες λειτουργίες (π.χ. οι πελατειακές υπηρεσίες) θα συνεχίσουν να λειτουργούν χωρίς διακοπή κατά τη διάρκεια της μετάβασης.

## 2. Παραγωγή Κώδικα (Φάσεις 3 και 4)

- **Business Continuity:**

- Η παραγωγή νέου κώδικα γίνεται σταδιακά και με ενσωμάτωση λειτουργιών από το παλιό σύστημα, ώστε να αποφεύγονται κενά ή απότομες αλλαγές.
- Η προσθήκη testing σε αυτή τη φάση διασφαλίζει ότι οποιαδήποτε νέα λειτουργικότητα λειτουργεί σωστά, χωρίς να επηρεάζει την παραγωγή ή τις ήδη υπάρχουσες επιχειρησιακές διαδικασίες.

## 3. Testing & Deployment-Packaging

- **Business Continuity:**

- Οι δοκιμές γίνονται πρώτα σε περιβάλλον testing και όχι σε πραγματικό περιβάλλον, ώστε να μειωθούν οι πιθανότητες αποτυχίας.
- Το deployment γίνεται με "phased rollout" (σταδιακή ανάπτυξη), επιτρέποντας τον εντοπισμό προβλημάτων πριν επηρεάσουν το σύνολο του συστήματος.
- Εξασφαλίζεται ότι οποιαδήποτε διακοπή θα είναι περιορισμένη και δεν θα επηρεάσει τον τελικό χρήστη.

## 4. Παρακολούθηση και Συμμόρφωση

- **Business Continuity:**

- Παρακολουθούνται συνεχώς οι επιχειρησιακές διαδικασίες ώστε να ανιχνευθούν γρήγορα προβλήματα ή αποκλίσεις από την προβλεπόμενη λειτουργία.
- Η συμμόρφωση με πρότυπα (compliance) διασφαλίζει ότι το σύστημα είναι ασφαλές και λειτουργικό χωρίς να επηρεάζονται οι δραστηριότητες της επιχείρησης.

## Part 4: Simulator Development

### **User guide - README.md :**

Για την εκτέλεση του προγράμματος μας πρέπει να είναι προεγκατεστημένες οι εξής βιβλιοθήκες:

- i) PyQt5
- ii) pycryptodome

Για την άμεση εγκατάσταση τους, μπορείτε να χρησιμοποιήσετε την εντολή:  
"pip install -r requirements.txt"

Στον φάκελο υπάρχουν τα εξής python αρχεία:

**1) e-shop.py:** Είναι ένα απλό e-shop που δημιουργήθηκε για σκοπούς testing. Σε αυτό χρησιμοποιούνται συναρτήσεις από κρυπτογραφικούς αλγορίθμους οι οποίοι ΔΕΝ είναι quantum safe, όπως οι εξής:

- i) AES 128bit
- ii) RSA 2048bit
- iii) MD5 hash

**2) risk\_assessment\_tool.py:** Είναι το inventory tool που ζητείται από την δεύτερη φάση του project. Για να τρέξει χρησιμοποιήστε το εξής command:  
"python3 risk\_assessment\_tool.py"

-INPUT: Φάκελος που θέλουμε να κάνουμε assess για post-quantum vulnerable cryptographic αλγορίθμους.

-OUTPUT: Δύο αρχεία:

- i) scan\_result.txt
- ii) scan\_result.json

Αυτά τα αρχεία περιέχουν προγράμματα και γραμμές κώδικα στα αντίστοιχα προγράμματα, όπου βρέθηκαν vulnerable αλγόριθμοι κρυπτογραφίας.

**3) simulator.py:** Είναι το simulator tool που ζητείται στην τέταρτη φάση του project. Τρέχει με την εξής εντολή:  
"python3 simulator.py"

-PRECONDITION: Πρέπει πρώτα να έχει τρέξει το πρόγραμμα risk\_assessment\_tool.py και να έχει παραχθεί το αρχείο scan\_result.json.

-INPUT: Το αρχείο scan\_result.json

-OUTPUT: Αλλάζει τον πηγαίο κώδικα του προγράμματος ώστε να χρησιμοποιεί SHA-3 που είναι safe post-quantum cryptographic algorithm. Εάν δεν μπορεί να αλλάξει τον πηγαίο κώδικα, τότε εμφανίζει στην κονσόλα ενημέρωση για το που υπάρχουν συναρτήσεις από vulnerable κρυπτογραφικούς αλγορίθμους.