**LAB-2**

**TEAM MEMBERS:- Rishikesh Sagare, Prabhav Nerulkar, Shreyash Patil**

**REPORT – KRYPTON**

***BY :-( RISHIKESH SAGARE)***

**Basic Commands & Tools Overview:**

* **ssh** — Connect to a remote server.
* **cd, ls, cat** — Navigate directories and read file contents.
* **echo** — Display text output.
* **nano / vim** — Edit text files.
* **mktemp -d** — Create temporary directories.
* **ln -s** — Create symbolic links.
* **chmod** — Change file permissions.

**Cryptography Tools:**

* **base64** — Decode Base64-encoded data.
* **tr** — Translate or substitute characters (used for ciphers).

**Level-by-Level Breakdown:**

**Level 0 → Level 1**

**Objective:**  
Decode a Base64-encoded password.

**Steps:**

1. Locate the Base64 string provided.
2. Use a Base64 decoder tool to decode the string.
3. Note the decoded output (password for Level 1).
4. Use this password to log in to the next level via SSH.

**Logic:**  
Base64 is an encoding method, not encryption. Its content can be directly reversed using a decoder.



**Level 1 → Level 2**

**Objective:**  
Decrypt a ROT13 (Caesar cipher with a 13-letter shift).

**Steps:**

1. Navigate to the directory containing the encrypted password.
2. Read the file containing the encrypted string.
3. Apply a ROT13 substitution by replacing each letter with the one 13 positions ahead in the alphabet.
4. Record the decrypted password for Level 2.

**Logic:**  
ROT13 rotates letters by 13 places. Applying the same operation twice restores the original text.



**Level 2 → Level 3**

**Objective:**  
Break a Caesar cipher with a fixed but unknown shift, using a known plaintext attack.

**Steps:**

1. Navigate to the directory containing the encryption tool.
2. Read the README for details on the encryption mechanism.
3. Create a temporary directory for testing.
4. Link the encryption program’s key file into this directory.
5. Create a plaintext file containing repeated characters (e.g., ‘AAAAA’).
6. Encrypt this file and analyze the result to deduce the shift pattern.
7. Apply the inverse shift to decrypt the target password file.
8. Note the decrypted password for Level 3.

**Logic:**  
Encrypting a known plaintext reveals the cipher’s character mappings, allowing deduction of the shift value and reversal of encryption.



**Level 3 → Level 4**

**Objective:**  
Decrypt another ROT13 cipher.

**Steps:**

1. Log into Level 3.
2. Navigate to the encrypted password file.
3. Apply ROT13 substitution.
4. Record the result as the Level 4 password.

**Logic:**  
Identical to Level 1 — a straightforward ROT13 decryption.



**Level 4 → Level 5**

**Objective:**  
Break a simple substitution cipher using frequency analysis.

**Steps:**

1. Log into Level 4.
2. Read and copy the encrypted password to a working file.
3. Count the frequency of each character.
4. Compare frequencies to typical English letter frequency.
5. Substitute the most frequent cipher letters with likely plaintext letters.
6. Iteratively adjust based on word patterns and known letter positions.
7. Derive the decrypted password for Level 5.

**Logic:**  
In substitution ciphers, frequent characters in ciphertext correspond to frequent characters in natural language, enabling decryption by analysis.



**Level 5 → Level 6**

**Objective:**  
Break a polyalphabetic cipher (likely Vigenère) using key length discovery.

**Steps:**

1. Log into Level 5.
2. Retrieve and save the encrypted password for analysis.
3. Identify repeating sequences to infer key length (Kasiski examination).
4. Divide ciphertext into segments based on key length.
5. Perform frequency analysis on each segment individually.
6. Derive the key and decrypt the password for Level 6.

**Logic:**  
Vigenère ciphers use multiple Caesar shifts. Determining the key length allows each segment to be cracked as a simple Caesar cipher.



**Level 6 → Level 7**

**Objective:**  
Decrypt a ciphertext using advanced analysis (similar to Level 5).

**Steps:**

1. Log into Level 6.
2. Retrieve and save the encrypted password for analysis.
3. Use frequency analysis and key length discovery techniques.
4. Decrypt the message based on identified patterns.
5. Note the final decrypted password for Level 7.

**Logic:**  
As ciphertext length increases, it provides more patterns and repeated sequences, making attacks like Kasiski examination or frequency analysis feasible.

