**DIGISURAKSHA PARHARI FOUNDATION**

**Cybersecurity Wargame Internship**

**Lab Report: Krypton Wargame**

**Website:** <https://overthewire.org/wargames/krypton/>

## TEAM MEMBERS:

* Prajakta Shinde - KRYPTON
* Saniya Shinde - NATAS
* Riya Rasal - NATAS
* Shraddha Khade - NATAS
* Kumkum Yadav - LEVIATHAN

**Krypton Level 0 → Level 1**

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

**Tools Used:** base64 command-line utility

**Approach & Logic:**

1. Accessed Level 0 and found the password was encoded using Base64: S1JZUFRPTklTR1JFQVQ=
2. Base64 is a common encoding scheme used to represent binary data in ASCII format
3. Used the Linux base64 utility with the -d (decode) flag to convert it back to plaintext

**Result:** The password "KRYPTONISGREAT" was successfully decoded and used to access Level 1.

**Krypton Level 1 → Level 2**

**Objective:** Break a simple substitution cipher (Caesar cipher) with preserved word boundaries.

**Tools Used:** cat command, Python script for shift analysis

**Approach & Logic:**

1. Used cat to examine the encrypted file content
2. The text appeared to be a simple substitution cipher where each letter is shifted by a fixed number
3. Created a Python script to test all 26 possible shifts (brute force approach)
4. By examining the output of each shift, identified ROT13 as the correct shift because it produced readable English text
5. ROT13 is a Caesar cipher that shifts each letter 13 positions in the alphabet

**Result:** Successfully identified ROT13 as the encryption method and decrypted the password for Level 2.

**Krypton Level 2 → Level 3**

**Objective:** Determine the Caesar cipher key using a provided encryption utility.

**Tools Used:** encrypt binary (provided in the game), mktemp, chmod, tr command

**Approach & Logic:**

1. Created a temporary directory to work in using mktemp -d
2. Created a symbolic link to the keyfile used by the encrypt program
3. Set permissions to allow writing to the directory with chmod 777
4. Created a test file containing all uppercase letters (A-Z) to determine the exact shift
5. Ran the encryption program on our test file
6. By comparing the input and output, determined the shift was 14 positions forward
7. Used the tr command to apply the reverse shift (14 positions backward) to decrypt the password file

**Result:** Successfully determined the Caesar shift was 14 positions and decrypted the password for Level 3.

**Krypton Level 3 → Level 4**

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

**Tools Used:** cat command, frequency analysis (manual or scripted)

**Approach & Logic:**

1. Examined the encrypted password file and found sample encrypted texts (found1, found2, found3)
2. Performed frequency analysis on the sample texts to identify letter frequencies
3. In English, the most common letters are E, T, A, O, I, N, S, H, R, D, L, U
4. Identified common patterns such as "THE", "AND", "ING" endings, etc.
5. Built a substitution table mapping encrypted letters to their plaintext equivalents
6. Refined the mapping by testing it on sample texts until the output made sense
7. Applied the complete substitution key to decrypt the password

**Result:** Created a complete substitution key and successfully decrypted the password for Level 4.

**Krypton Level 4 → Level 5**

**Objective:** Break a Vigenère cipher with a known key length of 6.

**Tools Used:** cat command, Python scripting for column-based frequency analysis

**Approach & Logic:**

1. A Vigenère cipher uses multiple Caesar ciphers in sequence, with different shifts determined by a key
2. Since the key length was given as 6, every 6th character in the ciphertext was encrypted with the same Caesar shift
3. Split the ciphertext into 6 separate streams (columns), one for each position in the key
4. Performed frequency analysis on each column independently
5. For each column, determined the most likely shift by assuming 'E' is the most common letter in English
6. Combined the shifts from all columns to form the complete Vigenère key
7. Applied the key to decrypt the password

**Result:** Successfully determined the Vigenère key and decrypted the password for Level 5.

**Krypton Level 5 → Level 6**

**Objective:** Break a Vigenère cipher with an unknown key length.

**Tools Used:** Kasiski examination method, Index of Coincidence (IoC) analysis, Python

**Approach & Logic:**

1. With an unknown key length, first needed to determine how long the key was
2. Used the Kasiski examination method:
   * Identified repeating sequences in the ciphertext
   * Calculated distances between repetitions, which are likely to be multiples of the key length
   * Found common factors of these distances to estimate possible key lengths
3. Confirmed the key length using the Index of Coincidence analysis:
   * English text has an IoC of approximately 0.067
   * Random text has an IoC of approximately 0.038
   * Tested different key lengths and calculated the average IoC for the resulting columns
   * The key length with an IoC closest to 0.067 is most likely correct
4. Once the key length was determined, used the same column-based frequency analysis as in Level 4
5. Applied the recovered key to decrypt the password

**Result:** Successfully determined both the key length and the key itself, allowing the decryption of the password for Level 6.

**Krypton Level 6 → Level 7**

**Objective:** Perform a known-plaintext attack on a stream cipher.

**Tools Used:** hexdump command, Python for XOR operations

**Approach & Logic:**

1. Stream ciphers typically use XOR with a repeating keystream to encrypt data
2. With access to both plaintext and its corresponding ciphertext, we can extract the keystream:
   * keystream = plaintext XOR ciphertext
3. Used hexdump to examine the provided plaintext and ciphertext files
4. Extracted the keystream by XORing the plaintext with its corresponding ciphertext
5. Determined the keystream's length by looking for repeating patterns
6. Applied the extracted keystream to decrypt the password file using the same XOR operation

**Result:** Successfully recovered the keystream, determined its period, and decrypted the final password, completing the Krypton wargame.

**Summary**

The Krypton wargame provided an excellent opportunity to practice various cryptanalysis techniques on increasingly complex ciphers:

1. Base64 decoding (Level 0-1)
2. Caesar shift cipher breaking (Level 1-2, 2-3)
3. Monoalphabetic substitution cipher breaking using frequency analysis (Level 3-4)
4. Vigenère cipher breaking with known key length (Level 4-5)
5. Vigenère cipher breaking with unknown key length using Kasiski and IoC methods (Level 5-6)
6. Stream cipher breaking using known-plaintext attack (Level 6-7)

These challenges enhanced my understanding of cryptographic vulnerabilities and the systematic approaches needed to break different cipher types. The hands-on experience with both classical and modern cryptographic techniques provides valuable skills applicable to security assessments and penetration testing scenarios.