The Last Dance			
Organization: HackTheBox		Type: offline challenge	
Categories:	☐ Network Security ☑ Cryptography ☐ Mobile Applications	☐ Reverse Engineering ☐ Web Applications ☐ Forensics	Difficulty: Easy
Name: Kasper Verhulst		Release date: 23-05-2020 Completing date: 28-12-2022	

Understanding the challenge

The challenge brings an encryption script and the some file containing the output of a round of such encryption. The encryption function will start by generating a random key and nonce. These two values will be used to initialize a ChaCha20 cipher. Afterwards, a hard-coded message and the flag are individually encrypted using the same instance of the ChaCha20 encryption machine.

Stream Ciphers

ChaCha20 is an alternative to AES adopted by the TLS-standard. Contrarily to AES, ChaCha is a stream cipher. Fundamentally all stream ciphers follow the same idea. A key generation function is used to extend a key together with an initial value to a pseudorandom, continuous output. Afterwards, the plaintext is XORed with this key stream. Decryption happens exactly the same way. Since stream cipher are still symmetric encryption mechanism, the receiver will have access to the shared key. He can recreate the key stream and xor the ciphertext to obtain the original plaintext. Figure 1 illustrates stream ciphers.

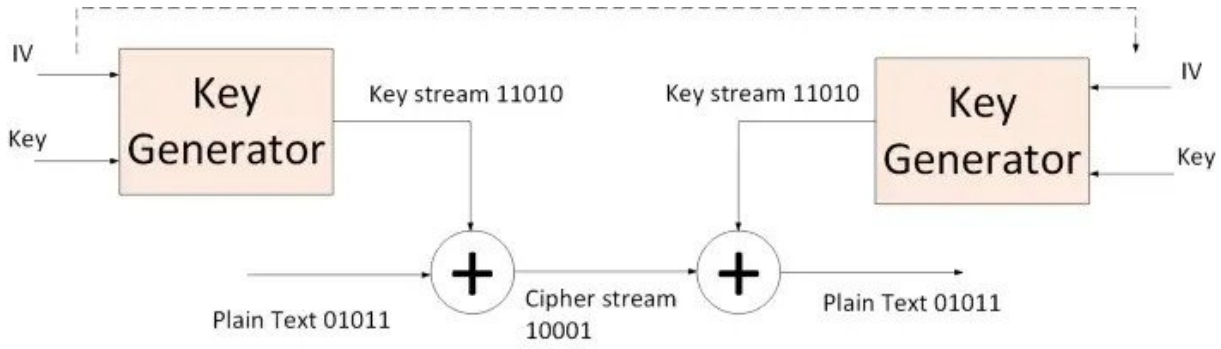

Figure 1: Stream cipher

$$\begin{aligned} K &= PRF(key, IV) \\ C_i &= K_i \oplus P_i \end{aligned} \tag{1}$$

Solving the challenge

Once we understand how stream ciphers work, it is clear that key reuse is completely breaking the security of a stream cipher. The encoded file contains two encrypted messages that we can use:

$$\begin{aligned} C_1 &= ChaCha_1(key, IV) \oplus P_1 \\ C_2 &= ChaCha_2(key, IV) \oplus P_2 \end{aligned} \tag{2}$$

Combining bot equations,

$$C_1 \oplus C_2 = (ChaCha_1(key, IV)) \oplus P_1 \oplus ChaCha_2(key, IV) \oplus P_2) \tag{3}$$

Since the same key and initialization vector were used:

$$C_1 \oplus C_2 = k \oplus P_1 \oplus k \oplus P_2) \tag{4}$$

$$C_1 \oplus C_2 = P_1 \oplus P_2 \tag{5}$$

Since we have both ciphertexts, and we know the hard-coded message that was also encrypted:

$$C_2 = P_1 \oplus P_2 \oplus C_1 \tag{6}$$