Writeup: Baby RSA 2

Este desafío se basa en una variante del criptosistema RSA donde se nos proporcionan valores específicos que revelan una vulnerabilidad explotable.

Análisis del Desafío

El reto nos proporciona un script Python (chall.py) que genera parámetros RSA y cifra la bandera, junto con un archivo output.txt que contiene los valores públicos resultantes.

El script realiza las siguientes operaciones importantes:

- Establece el exponente público e = 3
- Genera dos primos p y q de 1024 bits
- Calcula n = p * q y phi = (p - 1) * (q - 1)
- Cifra phi y la suma p+q: `phi_enc = pow(phi, e, n)` y `pplusq_enc = pow(p + q, e, n)`
- Cifra la bandera: `ct = pow(m, e, n)`
- Asegura que `pow(bytes_to_long(flag), e) > n`, lo que impide un simple ataque de raíz cúbica

En output.txt encontramos los valores de e, n, phi_enc, pplusq_enc y ct que necesitaremos para el ataque.

Identificación de la Vulnerabilidad

La vulnerabilidad principal es que se han cifrado dos mensajes relacionados (phi y p+q) con el mismo exponente público bajo (e=3) y el mismo módulo n. Esto nos permite aplicar el Ataque de Mensajes Relacionados de Franklin-Reiter.

Estos mensajes están relacionados por una función lineal:

- phi = (p - 1)(q - 1) = pq - p - q + 1 = n - (p + q) + 1
- Si definimos S = p + q, entonces: phi = n - S + 1

Metodología de Solución

1. Aplicar el ataque Franklin-Reiter para recuperar S = p + q
2. Usar S y n para factorizar n y obtener p y q
3. Calcular la clave privada d
4. Descifrar el mensaje cifrado (ct)

Aplicación del Ataque Franklin-Reiter

Para el caso específico donde e=3 y existe la relación phi = n - S + 1, podemos calcular S directamente con la fórmula:

S = (1 + 2*c1 - c2) * inverse(c1 + c2 + 2, n) mod n

Donde:

- c1 = pplusq_enc
- c2 = phi_enc

Factorización de n

Una vez obtenido S = p+q, podemos factorizar n resolviendo la ecuación cuadrática:
x^2 - S*x + n = 0

Las raíces de esta ecuación son p y q, calculables mediante:
p, q = (S ± sqrt(S^2 - 4n)) / 2

Implementación del Ataque

El script de solución sigue estos pasos:

1. Lee los valores de output.txt
2. Calcula S mediante la fórmula del ataque Franklin-Reiter
3. Verifica si el denominador comparte algún factor con n (lo que nos daría directamente p o q)
4. Si no, calcula S y verifica su validez

5. Factoriza n utilizando S para obtener p y q
6. Calcula phi = (p-1)*(q-1) y d = inverse(e, phi)
7. Descifra ct usando m_int = pow(ct, d, n)
8. Convierte el resultado a bytes para obtener la bandera

Conclusión

Al ejecutar el script de solución con el archivo output.txt proporcionado, logramos romper la implementación de RSA aprovechando la información adicional cifrada (phi y p+q) con el mismo exponente bajo e=3.

La bandera recuperada es:

```
hfctf{3xxXXt3M3lY_l0nG_fl4G_T0_pr3vEnT_b31NG_4Bl3_T0_d0_Cub3_r00t_b3C4uS3_N_1s_T00_B1g_4nD_E_T00_Sm4l1...}
```