**Phân tích mã hóa RSA "lạ"**

Script sử dụng RSA không chuẩn, với một n được tạo từ **các prime nhỏ**:

*while n.bit\_length() < 4096:*

*i = random.randint(10,16) # prime nhỏ: 10 đến 16 bits (~1024 đến 65536)*

*reps = random.randint(2,5) # lũy thừa của prime*

*p = getPrime(i)*

*if n % p != 0:*

*n \*= p\*\*reps*

➡️ n được tạo bằng cách nhân **nhiều prime nhỏ lũy thừa** với nhau, đến khi đạt 4096 bits.

**🧠 Ý tưởng tấn công: Håstad Broadcast Attack / Factoring**

Do n **không phải tích của hai big primes như RSA chuẩn**, mà là tích của **nhiều prime nhỏ**, nó có thể bị **phân tích thừa số** (factoring) dễ dàng.

**✅ Hướng khai thác:**

Vì n là tích của nhiều prime nhỏ, ta có thể dùng sympy.ntheory.factorint để phân tích n.

Sau khi có tất cả các (p^e) tham gia vào n, ta có thể tính:

* ϕ(n): tích (p^e - p^(e-1)) cho từng thừa số
* Dùng extended Euclidean algorithm để tìm d = inverse(e, ϕ(n))
* Giải RSA bình thường: flag = pow(ciphertext, d, n)
* Convert long về bytes để lấy flag.

byuctf{3ulers\_ph1\_function\_15\_v3ry\_us3ful\_4nd\_th15\_I5\_a\_l0ng\_fl4g}