broken oracle (HackTM CTF) 2023

Бакиновский Михаил

БФУ им. Иммануила Канта
ОНК «Институт высоких технологий»
Компьютерная безопасность, 3 курс

1 июля 2023 г.

Условие задачи broken oracle 2023

Описание с CryptoHack

Описание к заданию: *"Я заново реализовал криптосистему, но иногда она ведет себя странно. Но я не думаю, что это имеет значение"*

Так же к задаче прилагается файл *server.py.*
server.py содержит код на языке Python. В нём дана реализация криптографического алгоритма Рабина. При запуске кода нам предлагается ввести некоторое значение (зашифрованное сообщение) в формате `(r,s,t)`, на вывод мы также получим `(r,s,t)` - что является шифровкой расшифровки нашего изначального сообщения

Понятно, что нам нужно каким-то образом извлечь флаг или используя уязвимости криптосистемы или поняв закономерность шифрования.

Для начала проведем анализ криптосистемы. Нам дан оракул (консоль), в ответах которого иногда возникает ошибка. А также:

(n,c)-открытый ключ, состоящий из модуля n и c, что выбирается случайным образом

Упрощая, нам необходимо найти приватный ключ (p,q), после чего вычислить с и использовать эти значения для расшифровки флага.

Анализ задачи broken oracle 2023

Предварительный анализ

- Атаки факторизацией не будут успешны в угоду того, что, технически, нам доступен только оракул, а число n - взято длиной 2048 бит, что делает факторизацию нерациональной по времени. Метод Ферма, к примеру, не сработает, ведь тут p и q - простые числа, взятые случайным образом, и это не означает того, что они будут "близки"друг другу.

- Использовались ещё различные атаки: такие, как атака повторным шифрованием, встреча посередине и поиск квадратных корней по модулю n. Но они не увенчались успехом (за рациональное время).

После проведения серии запросов к оракулу было обнаружено следущее:

- Иногда при вводе значений r,s,t консоль выдаёт в ответ те же самые r,s,t, причём, если для такого r менять s и t - консольный вывод не изменится относительно r. Что, с точки зрения кода, должно выполняться всегда.
- Для произвольного шифра r,s,t происходит следующее: $(r_1, 1, 1) = (r_1, -1, 0)$ и $(r_2, -1, 1) = (r_2, 1, 0)$. Это говорит о том, что вместо 4 сообщений в расшифровке мы получаем лишь 2. На этом же основаны упрощения реализации криптосистемы и условия "assert len()==2"

Анализ задачи broken oracle 2023

Анализ программы. Наблюдения

- из 1000*k попыток 25% - ситуация, когда $enc(r, s = 1, t = 1) = enc\_dnc\_enc(r = r, s = 1, t = 1)$, возникает в 25% случаев для r в диапазоне от 1 до 1000*k и assertion error возникает в 6% случаев.
- Если символ лежандра для (r,n)=-1 то зашифрованное сообщение будет иметь единственный вариант расшифровки. Это связано с тем, что r будет являться квадратичным невычетом, а, значит, не будет иметь квадратных корней по модулю n, следовательно расшифровка будетт единственная и равна она будет зашифровке.

Если сравнивать реализацию криптосистемы Рабина в задаче с тем, как она представлена в открытом доступе, можно заметить ошибку в решении систем уравнений (solve_squad):

$$x^2 + rx + c = 0 \ mod \ p$$

не имеет решений, когда

$$\left(\frac{r^2/4 - c}{p}\right) = -1$$

потому что

$$x^2 + rx + c = 0 <=> (x + r/2)^2 = r^2/4 - c$$

Когда это не выполняется solve_quad

В функции decrypt возможен случай, когда число $r^2/4 - c$ является квадратичным вычетом по модулю p, но не по модулю q.
Пусть зашифрованное сообщение r было расшифровано и результат равен r'. В этом случае $r = r'$ в GF(p) пока $r \neq r'$ в GF(q). Это означает, что $r - r'$ является квадратом некоторого числа по модулю p, но не по модулю q.
Поэтому, если обработать несколько $r_i' = Enc(Dec((r_i))$ - будет возможно восстановить p с помощью $\text{НОД}(r_i' - r_i, r_j' - r_j)$. Это также справедливо и для восстановления q.

Решение задачи broken oracle 2023

Теоретические рассчёты

В нашей криптосистеме роль открытого ключа играет c.
Однако, конкретно в данной задаче c не доступно (так как это CTF).
Пусть $a_1, a_2 = solve_{quad}(r, c, p)$ и $b_1, b_2 = solve_{quad}(r, c, q)$, где r удовлетворяет уравнению

$$\left(\frac{r^2/4 - c}{p}\right) = \left(\frac{r^2/4 - c}{q}\right) = -1$$

Если обработать r_1, r_2 как $r_1 = Enc(m_1), r_2 = Enc(m_2)$, где $m_1 = a_1 \ mod \ p$, $m_1 = b_1 \ mod \ q$, $m_2 = a_2 \ mod \ p$, $m_2 = b_2 \ mod \ q$ путём изменения параметров s и t с тем же самым r. Как только $m_2 = r - m_1 \ mod \ n$ следующие уравнения:

$$m_1^2 - r_1 m_1 + c = 0 \ mod \ n$$
$$(r - m_1)^2 - r_1(r - m_1) + c = 0 \ mod \ n$$

Можно решить достаточно просто:

$$m_1 = \frac{r_2 r - r^2}{r_1 - 2r + r_2} \ mod \ n$$
$$c = r_1 m_1 - m_1^2 \ mod \ n$$

Таким образом, возможно узнать c, m с помощью r, r_1, r_2
Также можно восстановить публичный и секретный ключ, чтобы расшифровать флаг.

Решение задачи broken oracle 2023

Код для решения данной задачи

функция, имитирующая запросы к оракулу.

```
def oracle(r,s,t,h=1):
    RST=encrypt(decrypt(Enc(r=r,s=s,t=t),pub,priv),pub)
    if h!=1: return RST
    return RST.r, RST.s, RST.t
```

Часть кода, оставшаяся от изначальной версии оракула.

```
pbits = 1024
pub, priv = genkey(pbits)
```

Решение задачи broken oracle 2023

Код для решения данной задачи 1 стр

```
    res = []
for i in range(1, 21):
    rst = oracle(i, 1, 1)
    if rst[0] is None: continue
    res.append(rst[0] - i)
factors = set()
for i in range(len(res)):
    if res[i] == 0: continue
    for j in range(i + 1, len(res)):
        if res[j] == 0: continue
        tmp = gcd(res[i], res[j])
        if tmp > 2**100:
            for pi in primerange(1000):
                while True:
                    if tmp % pi == 0:
                        tmp //= pi
                    else: break
            factors.add(tmp)
```

Решение задачи broken oracle 2023

Код для решения данной задачи

```
assert len(factors) == 2
p = int(factors.pop())
q = int(factors.pop())
n = p * q
print("Recover p, q, n:\np = %d \nq = %d \nn = %d"%(p,q,n))
r = None
for i in range(100):
    rst = oracle(i, 1, 1)
    if rst[0] is None: continue
    if gcd(rst[0] - i, n) == 1:
        r = i
        break
```

Решение задачи broken oracle 2023

Код для решения данной задачи

```
assert r is not None
rs = []
for s in [-1, 1]:
    for t in [0, 1]:
        rs.append(oracle(r, s, t)[0])
for i in range(4):
    for j in range(i + 1, 4):
        r1 = rs[i]
        r2 = rs[j]
        try:
            m1 = (r2 * r - r ** 2) * pow(r1 - 2 * r + r2, -1,
            c = (r1 * m1 - m1 ** 2) % n
            print(long_to_bytes(decrypt(enc_flag, Pubkey(n=n,
        except Exception as e:
            print(e)
            continue
```

Используемая литература

- Menezes, A., van Oorschot, P., and Vanstone, S. (1996). "Handbook of Applied Cryptography". CRC Press.
- Stinson, D. R. (2006). "Cryptography: Theory and Practice". CRC Press.
- Shoup, V. (2009). "A Course in Number Theory and Cryptography". Graduate Texts in Mathematics, Vol. 114. Springer.
- Schneier, B. (1996). "Applied Cryptography: Protocols, Algorithms, and Source Code in C". John Wiley & Sons.
- Boneh, D. (2013). "Introduction to Cryptography". Online Course, Coursera.