CTF - Algo/Crypto

cmj

Outline

- 介紹 HITCON 2014 遇到的 演算法 (Algo)/密碼學 (Crypto) 的問題
- 主要的解題工具 (程式語言) - 思路與解法

Who Am I

- 攻城師 -
- 本土 CS Master
- 2-Year 工作經驗 (N-Year 騙小孩經驗)
- HST 食物鏈底層

CTF

- CTF (Capture The Flag)
  - Wargame (時代的眼淚)
- 目的
  - 找出藏在題目中的 Flag (線上賽)
  - 把你的 Flag 塞到目標當中 (現場決賽)

Question

- 題目類型很多...
  - Trivial/MISC - 身為 黑黑 該知道的事情
  - Algo/ACM/Crypto/Game - 考驗正規程式能力
  - Reverse/Exploit - 考驗非正規程式能力
  - Recon - 人肉搜索
  - etc …

RSAHA

crypto

RSAHA (crypto)

- 這題題目給你
  - 題目名稱
    - 看起來跟 RSA 有關

RSAHA (crypto)

- 題目給你
  - 題目名稱
    - 看起來跟 RSA 有關
  - IP
    - 現階段用不到 (通常最後 Flag 要跟 Server 要)

RSAHA (crypto)

- 題目給你
  - 題目名稱
    - 看起來跟 RSA 有關
    - IP
      - 現階段用不到 (通常最後 Flag 要跟 Server 要)
    - 一個 Python 的 sample code
      - 值得看一下、加密演算法可能在裡面

Sample Code - encrypt

- 簡單來說
  - 隨機提供兩個質數：p,q
  - 明文：m
  - 加密之後給你：
    - n = p*q
    - (m ^ 3) % n
    - ((m+1) ^ 3) % n

```
def encrypt(bits, m):
    p = random_prime(bits)
    q = random_prime(bits)
    n = p * q
    assert m < n
    print n
    print m ** 3 % n
    print (m + 1) ** 3 % n
```

看起來好像很簡單

Brute Force

- 首先 回到程式碼
  - 有一個輸入的參數：bits
    - 題目會跑 11 Round：每次增加 50 bits
      - 2^50 ≈ 10^15 ≈ 百兆
      - 2^550 ≈ 10^165 (中文不好...)
    - 複雜度：O(2^n)

如果你有超級電腦，再來考慮這招

Question

- 回到題目，問題就變成要解出來
  - GF(n) 下，給你 m^3 跟 (m+1)^3，求 m

Question

- 回到題目，問題就變成要解出來
  - GF(n) 下，給你 m^3 跟 (m+1)^3，求 m
- 這是中文嗎...

Question

- 回到題目，問題就變成要解出來
  - GF(n) 下，給你 m^3 跟 (m+1)^3，求 m
- 這是中文嗎...
  - 這是數學問題 !!

Background

- 高中數學 (能夠解題，但是不知道為什麼)
    - 因式分解
- 大學數學 (知道一定解的出來理由)
    - 代數 (費馬小定理)

University Part

- GF(p) - Galois Field
    - 簡單來說就是 % p (p是質數)
    - GF(7) ≈ [0, 1, 2, 3, 4, 5, 6]
    - 具有交換、分配律
- 費馬小定理告訴你
    - 在 GF(p) 中，存在反元素
    - 給我一個 n，我一定可以找到 m 使得 n*m ≡ 1 (mod p)
    - 方法：輾轉相除法 (Library: sympy.invert)
    - 時間複雜度：O(log(n))

University Part

- 舉個例子
  - GF(11)
    - $2 * 6 \equiv 1 \pmod{11}$
    - $5 * 9 \equiv 1 \pmod{11}$
  - GF(31)
    - $2 * 16 \equiv 1 \pmod{31}$
    - $11 * 17 \equiv 1 \pmod{31}$

Junior Part

- 想辦法把 m^3 跟 (m+1)^3 變成一堆加法、乘法的運算

Junior Part

- 首先，高中數學的因式分解可以拿到...
- 接著，用 m^2 + m 串起來兩個式子
- 最後，可以將m用一堆東西表示

1. $(m+1)^3 - m^3 = 3m^2 + 3m + 1 = 3(m^2 + m) + 1$
   - $\Rightarrow m^2 + m = 3^{-1}((m+1)^3 - m^3 - 1)$
2. $m^3 - 1 = (m-1)(m^2 + m + 1)$
   - $\Rightarrow m - 1 = (m^3 - 1)(m^2 + m + 1)^{-1}$
3. $m - 1 = (m^3 - 1)(3^{-1}((m+1)^3 - m^3 - 1))^{-1}$

Remind You

- 如果我們可以用
  - (m+1)^3、m^3、常數
  - 用加減乘除、反元素
- 我們就可以得到 m

就這樣

Q&A Round 1

24

Game

Game

- 什麼是 24 Game
  - 給你四個數字，想辦法用一大堆運算得到24
  - 例如：
    - 1*2*3*4 = 24
    - 4*(5-6+7)=24

Limitation

- 有時候會給一些限制
  - 例如：
    - 順序不能改變
    - 運算方式不只有加減乘除
    - etc…

Challenge

- 兩分鐘內解完所有 24 game (Based-On Python)
- 限制：
  - 數字從 1~13 都有可能
  - 接受答案：[+-*/0-9()]*
    - Python下的正規表示法

Operation

- 根據正規表示法，除了數字之外，我們可以使用
  - 括號：()
  - 加減乘除：+ - * /
  - 次方：**
  - 無條件捨去：//

Point

- 幾個重點
- 所有數字都是 Float
- 所有的運算都是交給 eval 執行

First Try

- 把問題丟給線上解題
- 可是 (But…)，重點就是這個 But
    - 線上解題都是 0~9
    - 線上解題都不支援次方 (**)

Problem Analsys

- 可以把問題拆解成下面的方程式
  - ±x (OP_1) ±y (OP_2) ±z (OP_3) ±w
- 所有可能的狀況，也就只會有
  - 4! * 2^4 * (6-2)^3 = 24576
  - 暴力解可行

Tools

- 使用工具來快速產生測試結果
- python - itertools
- 可以產生排列、組合

Coding

- 列出所有數字的排列可能 4!
    - 列出可能的運算方式
        - 2^4 (正負號) * 4^3(運算)
    - 產生暫時的運算式
    - 確定他的值是否為24

Source Code

```python
import itertools

def solver(nums):
    OPS = "* / ** //".split()
    FMT = "{sign[0]}{num[0]} {op[0]} {sign[1]}{num[1]} "\
          " {op[1]} {sign[2]}{num[2]} "\
          " {op[2]} {sign[3]}{num[3]}"

    for ops in itertools.product(OPS, OPS, OPS):
        for _num_ in itertools.permutations(nums):
            for sign in itertools.product(*[["", "-"] for n in range(4)]):
                tmp = FMT.format(sign=sign, op=ops, num=_num_)
                if abs(eval(tmp)-24) < 10**(-5):
                    return tmp
    else:
        raise KeyError
if __name__ == "__main__":
    print "Found: %s " %solver([2, 3, 4, 5])
```

But…

- 又來一個 But …
  - 需要做很多防呆
  - 題目會來個：1, 4, 13, 13
  - 如果不小心你產生的運算是…
    - ((1+4)**13)**13

Thanks for your attention

Q&A