2020CTF笔记crypto部分

原创

OceanSec 于 2020-04-02 17:30:53 发布

分类专栏： # CTF

本文链接：https://blog.csdn.net/q20010619/article/details/105274965

准备工作

安装python2、3共存环境

1.检查环境变量，缺少的我们需要添加。先找到环境变量的位置。

在Path环境变量中检查以下4个变量（Path中的环境变量是以分号隔开的）：

1.c:\Python27

2.c:\Python27\Scripts

3.c:\Python33

4.c:\Python33\Scripts

少哪个加哪个，注意分号隔开。

2.然后进入Python2.7安装目录找到如图内容，把python.exe删除。

图片: https://uploader.shimo.im/f/BynG3hRKS0MCoGxF.png

然后进入python3.3安装目录。找到python.exe程序，把它重命名为python3.exe

最后打开命令行界面测试一下。执行python2命令会进入python2.7的交互环境，执行python3命令会进入python3.3交互环境。

3.使用命令分别重新安装pip

4.安装wheel

图片: https://uploader.shimo.im/f/heD44XaG9eM8Dsc4.png

5.然后再在cmd中输入：pip install [whl文件的绝对路径]安装whl文件包。

6.输入：pip install gmpy2就可以安装了或者是在cmd中python交互环境下输入import gmpy2如果没报错，gmpy库只能用于python2

7.依次安装Crypto、PublicKey、multiprocessing

CTF-crypto正式内容
前述
如果密文是十进制，字符范围为“0-9”，可以猜测是ASCII编码；
如果密文由“a-z”“A-Z”和“=”构成，特别是末尾有“=”，那么可以判断为Base64编码；
如果密文含有“%”，形式为 “%xx” 和 “%uxxxx”，字符范围又是十六进制“0-F”，判断是escape() 函数编码，用unescape()解码；
若密文由“[]，()，{}，+,！”字符组成的编码通常就通过Jother解密。
图片: https://uploader.shimo.im/f/sk7NMRMNQk8XlMmq.png
图片: https://uploader.shimo.im/f/10sPWtlPZ3YDzvgd.png
base64
base64加密解密
加密：

> import base64
> encode = base64.b64encode(b’I love you’)
> encode
> b’SSBsb3ZlIHlvdQ==’
> 解密：

> import base64
> decode = base64.b64decode(b’SSBsb3ZlIHlvdQ==’)
> decode
> b’I love you’
> 类似的将base64.b32encode变成base64.b16encode
> 图片: https://uploader.shimo.im/f/QuE10lUyfOUxAf8k.png
> 图片: https://uploader.shimo.im/f/Mg7cDiDH1tgAGfO2.png
> url编码
> 图片: https://uploader.shimo.im/f/Gvwztq4uUS04dZqs.png
> base32中只有大写字母（A-Z）和数字234567
> html编码escape编码morse
> 图片: https://uploader.shimo.im/f/wyaFras1qBwT9lM7.png
> 古典密码学
> 不暴力不成活工具

图片: https://uploader.shimo.im/f/0vyhupFwKUMgivJA.png
图片: https://uploader.shimo.im/f/5xrsdyvpo3gnMFdo.png

图片: https://uploader.shimo.im/f/utu9kl3Hdi8hbTKl.png
图片: https://uploader.shimo.im/f/4RupcMEwl0UdODDs.png
escape编码
图片: https://uploader.shimo.im/f/ATSZyK0hlNoh8kLf.png
凯撒密码
图片: https://uploader.shimo.im/f/UnrUAAw8XVwuOUUV.png
图片: https://uploader.shimo.im/f/fmRgfcJVeCEYltJJ.png
图片: https://uploader.shimo.im/f/yB3ZkMfg7O0W1XUP.png
图片: https://uploader.shimo.im/f/sOOMcZoc8PEC6jfg.png
出现固定替换
多表替换-维吉尼亚密码
图片: https://uploader.shimo.im/f/eR2XN906P2UmdbcL.png
置换密码
图片: https://uploader.shimo.im/f/gNTQAsFKuKgvrj0A.png
栅栏密码

图片: https://uploader.shimo.im/f/L2baKWRWCnsHmqZk.png
维吉尼亚唯密文解析
图片: https://uploader.shimo.im/f/B92iWm80eblzdbVH.png
图片: https://uploader.shimo.im/f/R61DQraN3NE3dl1X.png
数字可以被开方可以考虑二维码

QWE加密
从电脑键盘上的字母从Q开始数，顺序是Q W E R T Y U I。。。对应的字母顺序依次是A B C D E F G H 也就是说
Q=A,W=B,E=C，依次类推
jsscript/vbscript

图片: https://uploader.shimo.im/f/N5C6bQBRSUAlanPs.png
图片: https://uploader.shimo.im/f/tQ7x1EWffLE4kQ4y.png
密文分析
图片: https://uploader.shimo.im/f/wkmr6KhUDxEcDY0i.png
图片: https://uploader.shimo.im/f/N0PZOa69VeY80ANa.png

RSA
好文章
http://www.freebuf.com/articles/others-articles/166049.html
https://xz.aliyun.com/t/2446#toc-28
https://www.anquanke.com/post/id/84632
http://www.freebuf.com/articles/others-articles/161475.html
http://www.freebuf.com/sectool/163781.html
http://skysec.top/2018/08/25/RSA%E4%B9%8B%E6%8B%92%E7%BB%9D%E5%A5%97%E8%B7%AF-2/
http://www.ruanyifeng.com/blog/2013/07/rsa_algorithm_part_two.html
https://wenku.baidu.com/view/4e8db1e081c758f5f61f6766.html
http://www.ruanyifeng.com/blog/2013/06/rsa_algorithm_part_one.html
https://www.freebuf.com/articles/others-articles/161475.html
https://err0rzz.github.io/2017/11/14/CTF%E4%B8%ADRSA%E5%A5%97%E8%B7%AF/#
https://l1b0.github.io/2018/07/29/Whitzard-CTF-RSA/
https://www.anquanke.com/post/id/164575

RSA的数学基础理论

素数（质数）、合数、互质数

素数：一个数如果除了1与它本身之外没有其他的因数，那么这个数就被称为素数（或者质数，取自英文单词prime的首字母）。

合数：如果一个数大于1，且该数本身不是素数，那么这个数就是一个合数。

互质数：如果两个整数a,b的最大公因数（greatest common divisor）为1，即gcb(a,b)=1，那么称a,b两数互质,。

欧拉函数值

设m为正整数，则1,2,3,4…….,m中与m互素的整数的个数记为图片: https://uploader.shimo.im/f/RwuHFWutdz4aWwCK.png，叫做欧拉函数，欧拉函数的值叫做欧拉函数值。

取模运算与同余的概念

如果存在一个正整数m与两个整数a,b，如果a-b能够被m整除，也就是说m|(a-b),那么a和b模m同余。记为：

图片: https://uploader.shimo.im/f/PSdqSZZw1EgQiHB0.png

模指数运算

模指数运算即先进行指数运算，之后再进行取模运算。例如：

图片: https://uploader.shimo.im/f/oeinf229DDgbCMx3.png

在Python中给出了相应的处理函数（这在之后的使用Python进行编码求解相关问题的时候非常重要）：

pow(x，y，z)

函数是计算x的y次方，如果z存在，则再对结果进行取模，其结果等效于pow(x,y)%z注意：pow() 通过内置的方法直接调用，内置方法会把参数作为整型，而 math 模块则会把参数转换为 float。

欧几里得扩展算法

算法定义：对于不完全为0的非负整数a,b，gcb（a,b）表示a,b的最大公约数，必然存在整数对x,y使得gcd(a,b)=ax+by成立。

证明：

假设 a>b

1、显然当 b=0，gcd（a，b）=a，此时 x=1，y=0；

2、ab!=0 时，设 ax1+by1=gcd(a,b);

bx2+(a mod b)*y2*=gcd(b,a mod b);

根据朴素的欧几里德原理有 gcd(a,b)=gcd(b,a mod b);

则:ax1+by1=bx2+(a mod b)*y2*;

即:ax1+by1=bx2+(a-(a/b)*b)y2=ay2+bx2-(a/b)*by2;

根据恒等定理得：x1=y2; y1=x2-(a/b)*y2;*

这样我们就得到了求解 x1,y1 的方法：x1，y1 的值基于 x2，y2.

上面的思想是以递归定义的，因为 gcd 不断的递归求解一定会有个时候 b=0，所以递归可以结束。

扩展欧几里得的递归实现（C语言）：

```
int exgcd(int a,int b,int &x,int &y)
{ if(b==0) { x=1; y=0; return a; } int r=exgcd(b,a%b,x,y); int t=x; x=y; y=t-a/b; return r;}
```

注：扩展欧几里得算法是之后RSA在知晓p、q、e的情况下求解d的主要思维算法，请详细了解。

RSA的加密解密原理

RSA加密解密涉及元素

N：大整数N，我们称之为模数（modulus）
p 和 q ：大整数N的两个因子（factor）
e 和 d：互为模反数的两个指数（exponent）
c 和 m：分别是密文和明文，这里一般指的是一个十进制的数还有一个就是n的欧拉函数值，在求解d的时候常用
RSA算法密钥的产生
（1）选择两个满足需要的大素数p和q，计算n=p*q，图片: https://uploader.shimo.im/f/KUs3eOJmlEoZHPrn.png，其中图片: https://uploader.shimo.im/f/ql3eQ761wwUlAJKx.png是n的欧拉函数值。
（2）选一个整数e,满足条件1<e<图片: https://uploader.shimo.im/f/mapn2pRG7sc4fSEc.png,且gcd(图片: https://uploader.shimo.im/f/pqzNDrjDI18nJaFX.png,e)=1。通过图片: https://uploader.shimo.im/f/PKtKJQhxUnMapSaa.png，计算出d.
（3）以{e,n}为公开密钥，{d,n}为秘密密钥。
RSA加密解密原理图
公钥：{e,n} 私钥：{d,n}
图片: https://uploader.shimo.im/f/QylCeIWkSX8iRi9Z.png
简单的保密通信模型
模拟场景：
假设Alice是秘密消息的接收方，则只有Alice知道秘密密钥{d,n}，所有人都可以知道公开密钥{e,n}。
加密操作：
如果发送方发送需要保密的消息m给Alice,就选择Alice的公钥{e,n}，然后计算图片: https://uploader.shimo.im/f/dQnzMl46yMoqo8K8.png，之后把密文c发送给接收方Alice即可。
解密操作：
接收方Alice收到密文c之后,根据自己掌握的私钥计算图片: https://uploader.shimo.im/f/ydvX9bu9euMgxPDd.png,所得结果即为发送方要发送的消息。可以根据上面的“简单保密通信模型”来了解这一个过程。

选择两个大素数p和q，计算出模数N = p * q
计算φ = (p−1) * (q−1) 即N的欧拉函数，然后选择一个e (1<e<φ)，且e和φ互质
取e的模反数为d，计算方法: e * d ≡ 1 (mod φ)
对明文m进行加密：c = pow(m, e, N)，得到的c即为密文
对密文c进行解密，m = pow(c, d, N)，得到的m即为明文
p 和 q ：大整数N的两个因子（factor）
N：大整数N，我们称之为模数（modulus
e 和 d：互为模反数的两个指数（exponent）
c 和 m：分别是密文和明文，这里一般指的是一个十进制的数
（N，e）：公钥
（N，d）：私钥
CTF中的RSA题型
公钥加密文
这是CTF中最常见最基础的题型，出题人会给你一个公钥文件（通常是以.pem或.pub结尾的文件）和密文（通常叫做flag.enc之类的），你需要分析公钥，提取出（N，e），通过各种攻击手段恢复私钥，然后去解密密文得到flag。
文本文档
对于第一种题型，耿直点的出题人直接给你一个txt文本文档，里面直接写出了（N，e，c）所对应的十进制数值，然后你直接拿去用就行了。当然也不都是给出（N，e，c）的值，有时还会给出其他一些参数，这时就需要思考，这题具体考察的什么攻击方法
pcap文件
有时出题人会给你一个流量包，你需要用wireshark等工具分析，然后根据流量包的通信信息，分析题目考察的攻击方法，你可以提取出所有你解题需要用到的参数，然后进行解密
本地脚本分析
题目会给你一个脚本和一段密文，一般为python编写，你需要逆向文件流程，分析脚本的加密过程，写出对应的解密脚本进行解密
远程脚本利用
这种题型一般难度较大。题目会给你一个运行在远程服务器上的python脚本和服务器地址，你需要分析脚本存在的漏洞，确定攻击算法，然后编写脚本与服务器交互，得到flag
例题

图片: https://uploader.shimo.im/f/A0r57pImfDgITGrv.pnggmpy2.mpz(x) # 初始化一个大整数x
图片: https://uploader.shimo.im/f/Qy8iFQxktX4ullKT.png
图片: https://uploader.shimo.im/f/0dtHl62x1rowseio.png
e=2
图片: https://uploader.shimo.im/f/a9kR3LRVNWl8co0h.png
e特别大说明d较小 D:\study file\web s\CTF工具合集\编码与密码\密码\RSA\e指数攻击\e很大\rsa-wiener-attack\RSAwienerReallyHack.py图片: https://uploader.shimo.im/f/cNMn05KK6r46FQkg.png
e为偶数
图片: https://uploader.shimo.im/f/eTPip1CpFmYewVRf.png图片: https://uploader.shimo.im/f/FPvACmp9kVMCju7q.png
工具
在大多数情况下，你只需要把题目给的信息输入给脚本，脚本就会自动完成剩下的工作如：
1：题目给了一个公钥文件和密文，直接用 --key 或 -k 指定公钥，用 –decrypt 指定密文文件就行了
2：题目给了你如（N，e，c）的十进制值，分别通过-N，-e，-c输入就行了
3：上面那种情况，如果题目是把这些参数都写入一个文本文件，如txt中，直接用 --input 或 -i 指定文本文件就行了
具体的例子：
Wiener’s attack
python solve.py --verbose --private -i examples/wiener_attack.txt
或者通过命令行，只要指定对应参数就行了
python solve.py --verbose --private -N 460657813884289609896372056585544172485318117026246263899744329237492701820627219556007788200590119136173895989001382151536006853823326382892363143604314518686388786002989248800814861248595075326277099645338694977097459168530898776007293695728101976069423971696524237755227187061418202849911479124793990722597 -e 354611102441307572056572181827925899198345350228753730931089393275463916544456626894245415096107834465778409532373187125318554614722599301791528916212839368121066035541008808261534500586023652767712271625785204280964688004680328300124849680477105302519377370092578107827116821391826210972320377614967547827619
利用 factordb.com 分解大整数
python solve.py --verbose -k examples/jarvis_oj_mediumRSA/pubkey.pem --decrypt examples/jarvis_oj_mediumRSA/flag.enc
small q attack
python solve.py --verbose --private -k examples/small_q.pub
费马分解（p&q相近时）
python solve.py --verbose --private -i examples/closed_p_q.txt

密文与模数不互素
Common factor between ciphertext and modulus attack
python solve.py --verbose -k examples/common_factor.pub --decrypt examples/common_factor.cipher --private
small e
python solve.py --verbose -k examples/small_exponent.pub --decrypt examples/small_exponent.cipher
Rabin 算法 （e == 2）
python solve.py --verbose -k examples/jarvis_oj_hardRSA/pubkey.pem --decrypt examples/jarvis_oj_hardRSA/flag.enc
Small fractions method when p/q is close to a small fraction
python solve.py --verbose -k examples/smallfraction.pub --private
Known High Bits Factor Attack
python solve.py --verbose --private -i examples/KnownHighBitsFactorAttack.txt
d泄漏攻击
python solve.py --verbose --private -i examples/d_leak.txt
模不互素
python solve.py --verbose --private -i examples/share_factor.txt
共模攻击
python solve.py --verbose --private -i examples/share_N.txt
Basic Broadcast Attack
python solve.py --verbose --private -i examples/Basic_Broadcast_Attack.txt
再列举几个实用的小功能
输入N与e创建公钥
python solve.py -g --createpub -N your_modulus -e your_public_exponent -o public.pem
查看密钥文件
python solve.py -g --dumpkey --key examples/smallfraction.pub
将加密文件转为十进制（方便写入文本，配合-i需要）
python solve.py -g --enc2dec examples/jarvis_oj_hardRSA/flag.enc
下面来介绍下我写这个工具的思路
这个工具如何工作
根据题目给的参数类型，自动判断应该采用哪种攻击方法，并尝试得到私钥或者明文，从而帮助CTFer快速拿到flag或解决其中的RSA考点
大体思路
判断输入
首先，识别用户的输入，可以是证书 pem 文件，也可以通过命令行参数指定n，e等变量的值，甚至可以通过命令行指定题目所给的txt文件并自动识别里面的变量
判断攻击算法
根据取到的参数类型及数量，选取可能成功的方法并采用一定的优先级逐个尝试。
如大整数分解的题型：给了一个公钥和一个加密的密文，我们需要先分解大整数N，然后得到私钥再去解密。考点在于大整数分解，脚本的关键代码在CTF-RSA-tool/lib/factor_N.py中的solve函数
def solve(N, e, c, sageworks):

```
if sageworks:

    return pastctfprimes(N) or noveltyprimes(N) or wiener_attack(N, e) or factordb(N) or comfact_cn(N, c) or sma
llq(N) or p_q_2_close(N) or boneh_durfee(N, e) or smallfraction(N) or None

else:

    return pastctfprimes(N) or noveltyprimes(N) or wiener_attack(N, e) or factordb(N) or comfact_cn(N, c) or sma
llq(N) or p_q_2_close(N) or None
```

选择输出

CTFer可以通过命令行选择是输出私钥还是输出解密后的密文，还是一起输出，不过非 --input（文本文档自动识别攻击） 的情况下，请至少选择 --private（打印得到的私钥） 或 --decrypt（解密一个加密的文件） 或 --decrypt_int（解密一个十进制数） 中的一个。

0×05.还是没有得到flag

首先如果题型是大整数分解的话，你还可以尝试使用其他工具如 yafu 来分解，如果还是不能分解，你就得再好好看看这题是不是另有切入点了。

其次，如果是一些你没见过的题型的话，建议通过百度，谷歌，github搜一下，一般还是能搜到的类似的题目甚至原题的

/