强网杯2020-dice2cry&baby_crt&bank

原创

CTF小白 于 2020-08-29 11:08:27 发布 1178 收藏

分类专栏： CTF 文章标签： CTF

本文链接：https://blog.csdn.net/qq_41429081/article/details/108292081

Web2 dice2cry

操作内容：

输入token后抓包

发现这边给我们设置了一些cookie

一看就知道这是一个RSA加密的密文，公钥n以及公钥e的值

然后抓包roll骰子那个界面

发现他会随机返回给我们一个数字，大概是0-2样子

这边发现abi.php有一个备份文件abi.php.bak

拿到这个abi.php的一个后台源码

```
<?php
session_start();
header("Content-type:text/html;charset=utf-8");
$data = json_decode($json_string, true);
$rand_number = isset($_POST['this_is.able']) ? $_POST['this_is.able'] : mt_      rand();
$n = gmp_init($data['n']);
$d = gmp_init($data['d']);
$c = gmp_init($rand_number);
$m = gmp_powm($c,$d,$n);
$v3 = gmp_init('3');
$r = gmp_mod($m,$v3);
$result=(int)gmp_strval($r);
$dice = array("num"=>$result);
$json_obj = json_encode($dice);
echo $json_obj;
?>
```

通过分析大概逻辑就是
我们可以输入一个this_is.able，如果接受到，就把他作为RSA里面的密文c的值。

如果不存在这个传参的输入那么就随机定义为一个随机数。

然后这个密文c的值会被解密出明文m，然后返回给我们的result为明文m%3的结果。

这边在密码学里面很常见，就是一个RSA Parity Oracle Attack的一个变形。像RSA Parity Oracle Attack的话是利用返回结果进行一个二分法去逼近正确的m。这题的话是mod3的，所以我们改成三分法去求解就好了。

但是这边发现，我们传入this_is.able后，返回的结果还是变化的。本地调试了一下，发现参数为this_is.able无法传入值。发现this_is.able的.符号会被转义成下划线_

这边的话其实之前有个题是考察php特性的，就是知道有些符号会被转义成下划线。

这边的话就去找php的一个底层特性的实现，发现空格和点被转义为了下划线

通过审计发现，当输入[号时，[号会被转义为下划线，但是后面的内容不会被匹配。于是我们这边利用了this[is.able进行绕过。然后去跑一个RSA的三分法攻击脚本就好了。

首先是我本地打RSA系统的一个脚本

模拟服务端，调试了一下改哪个奇偶攻击的脚本。改成一个三分法的逼近明文m的脚本。

```
from Crypto.Util.number import *
import gmpy2

flag = 'flag{hahahhahahahhahahhahahhaha}'

n=10595688168054455994991785156738068486213856030047270322045768566673623622880299119002188546334840716573755880
6991007895462808203457554499754532933347069118199504385800332444768661297496997465739842040185258271659284920709
0024220771005729932018393540734209503139016006138110472949377517429587003061693992525 01
e=0x10001
d=9913258399261898222209077294818978703643572277314775272999349152908431970650600741271925674154555340913322483 1
9187732138821970459923922624921638363231287484551490968399276359490268400014673017235197832219850123611840098116
6417420717335652680206341466028482040301714646178474043703049063919994724322067357534 5
c=6062141449970074839520813529992112601575277163930479864253278244555212041647046803263407616797643558731560255 5
1501130203386371608187515957393202951410641870157132746494414932922614993753927345207791349515002872068362863881
0275729220790804851792311320057888476892477632886296009666901254862504782795293291618 2
def decode(c):
    return pow(c,d,n)%3


def brute_flag(encrypted_flag, n, e):

    flag_count = n_count = 1
    flag_lower_bound = 0
    flag_upper_bound = n
    ciphertext = encrypted_flag
    mult = 1
    while flag_upper_bound > flag_lower_bound +1:

        ciphertext = (ciphertext * pow(3, e, n)) % n
        flag_count *= 3
        n_count = n_count * 3 - 2
        print("bit = %d" % mult)
        mult += 1
        data=decode(ciphertext)

        print(data)
        if(data==0):
            flag_upper_bound = n * n_count // flag_count
        elif(data==1):
            flag_lower_bound = n * n_count // flag_count
            n_count += 2
        else:
            flag_lower_bound = n * n_count // flag_count
            n_count += 1
    return flag_lower_bound
re=brute_flag(c,n,e)
from Crypto.Util.number import *
print(long_to_bytes(re))
```

这边成功在本地打出flag的一个明文m

然后去改一个和web服务交互的脚本

最终打服务端flag的脚本如下：

```
import requests
```

```python
def getMmod3(c):
    url = "http://106.14.66.189/abi.php"

    payload = {
        "this[is.able":str(c)
    }
    headers = {
    "Cookie":"PHPSESSID=3d5q3d0bkl04nul507qd2oglfd; encrypto_flag=134818612553724365889609460854941012998030164020601112142421347149614848005468874180269007510802236543519110835892561838533033184980788227546044445787185578583874076540391315495291106845865504388024820617362164670197703807921429606416829018770736739619385637142342162702890027024352319641425783650212188069 46; public_n=8f5dc00ef09795a3efbac91d768f0bff31b47190a0792da3b0d7969b1672a6a6ea572c2791fa6d0da489f5a7d743233759e8039086bc3d1b28609f05960bd342d52bffb4ec22b533e1a75713f4952e9075a08286429f31e02dbc4a39e3332d2861fc7bb7acee95251df77c92bd293dac744eca3e6690a7d8aaf855e0807a1157; public_e=010001"
    }

    response = requests.request("POST", url, data=payload, headers=headers)
    print(response.text)
    return response.text


def brute_flag(encrypted_flag, n, e):

    flag_count = n_count = 1
    flag_lower_bound = 0
    flag_upper_bound = n
    ciphertext = encrypted_flag
    mult = 1
    while flag_upper_bound > flag_lower_bound + 1:

        ciphertext = (ciphertext * pow(3, e, n)) % n
        flag_count *= 3
        n_count = n_count * 3 - 2
        print("bit = %d" % mult)
        mult += 1
        data=getMmod3(ciphertext)

        print(data=="{\"num\":0}")

        if(data=="{\"num\":0}"):
            flag_upper_bound = n * n_count // flag_count
        elif(data=="{\"num\":1}"):
            flag_lower_bound = n * n_count // flag_count
            n_count += 1
        else:
            flag_lower_bound = n * n_count // flag_count
            n_count += 2
    return flag_lower_bound

c=13481861255372436588960946085494101299803016402060111214242134714961484800546887418026900751080223654351911083589256183853303318498078822754604444578718557858387407654039131549529110684586550438802482061736216467019770380792142960641682901877073673961938563714234216270289002702435231964142578365021218806946
n=0x8f5dc00ef09795a3efbac91d768f0bff31b47190a0792da3b0d7969b1672a6a6ea572c2791fa6d0da489f5a7d743233759e8039086bc3d1b28609f05960bd342d52bffb4ec22b533e1a75713f4952e9075a08286429f31e02dbc4a39e3332d2861fc7bb7acee95251df77c92bd293dac744eca3e6690a7d8aaf855e0807a1157
e=0x10001
re=brute_flag(c,n,e)
from Crypto.Util.number import *
print(long_to_bytes(re))
```

强网先锋2 baby_crt

操作内容：

这边拿到了相关的paper

这边我们得到的数据有n、m、sig、e。c1未知，但是我们可以知道c1最后是mod上t1的，t1是小于e的，e是65537，所以我们对他进行爆破即可。

```
n=26318358382258215770827770763384603359524444566146134039272065206657135513496897321983920652242182112479484135343436206815722605756557098241887233837248519031879444740922789351356138322947108346833956405647578838873425658405513192437479359531790697924285889505666769580176431360506227506064132034621123828090480606055877425480739950809109048177976884825589023444901953529913585288143291544181183810227553891973915960951526154469344587083295640034876874318610991153058462811369615555470571469517472865469502025030548451296909857667669963720366290084062470583318590585472209798523021029182199921435625983186101089395997
m=26275493320706026144196966398886196833815170413807705805287763413013100962831703774640332765503838087434904835657988276064660304427802961609185997964665440867416900711128517859267504657627160598700248689738045243142111489179673375819308779535247214660694211698799461044354352200950309392321861021920968200334344131893259850468214901266208090469265809729514249143938043521579678234754670097056281556861805568096657415974805578299196440362791907408888958917063668867208257370099324084840742435785960681801625180611324948953657666742195051492610613830629731633827861546693629268844700581558851830936504144170791124745540
sig=20152941369122888414130075002845764046912727471716839854671280255845798928738103824595339885345405419943354215456598381228519131902698373225795339649300359363119754605698321052334731477127433796964107633109608706030111197156701607379086766944096066649323367976786383015106681896479446835419143225832320978530554399851074180762308322092339721839566642144908864530466017614731679525392259796511789624080228587080621454084957169193343724515867468178242402356741884890739873250658960438450287159439457730127074563991513030091456771906853781028159857466498315359846665211412644316716082898396009119848634426989676119219246
e = 65537
import gmpy2
from hashlib import sha1
from Crypto.Util.number import getPrime, long_to_bytes, getStrongPrime
for c1 in range(65537):
    if gmpy2.gcd(pow(m,c1,n)-pow(sig,e,n),n)!=1:
        p=gmpy2.gcd(pow(m,c1,n)-pow(sig,e,n),n)
q=n//p
assert(p*q==n)
if p>q:
    p=q
flag = "flag{" + sha1(long_to_bytes(p)).hexdigest() + "}"
print(flag)
```

强网先锋5 bank

操作内容：

nc 连接题目靶机

发现首先是一个pow。爆破前三个内容

就直接拿pow脚本爆破一下，然后发送token进入题目。

有几个选项。

hint里面给了我们一个AES加密的函数，大概意思就是会把我们交易的记录给加密，然后是分块加密的，分别加密发送者，接收者和金额。

get flag发现需要支付1000，但是我们只有10

provide a record 主要是提示我们可以提交记录，也会去执行这条记录去转账。

transact 转账，生成一个转账的记录

view records 系统应该是每次访问会自动先生成很多的一个记录。

这边的攻击思路就是，去transact里面先转个帐，目的是拿到我们的一个AES加密后的32位的一个东西。

然后去records里面把记录里面中间32位，也就是接收者改成我们的。

然后提交到provide a record里面

写脚本自动完成操作

发现有了1500

去get flag

图片已做防盗链处理
请在原文件中访问该图片

```python
import os
import re
from itertools import product
from hashlib import sha256
from pwn import *

add = "39.101.134.52"
port = 8005

def login(io):
    rec = io.recvline().decode()
    s = string.ascii_letters + string.digits
    suffix = re.findall(r'\(XXX\+(.*?)\)', rec)[0]
    digest = re.findall(r'== (.*?)\n', rec)[0]
    for i in product(s, repeat=3):
        prefix = ''.join(i)
        guess = prefix + suffix
        if sha256(guess.encode()).hexdigest() == digest:
            # print(guess)
            break
    print(prefix)
    io.sendafter(b'Give me XXX:', prefix.encode())
    return

sh = remote(add,port)

login(sh)

sh.sendafter(b'teamtoken:', "icqf0c3ad90f1390788c95c282e39ca4".encode())
sh.sendafter(b'give me your name:', "1".encode())
sh.sendafter(b'>', "transact".encode())
sh.sendafter(b'>', "2 1".encode())
my_sig=sh.recvuntil("your")[1:33]
sh.sendafter(b'>', "view records".encode())
sh.recvline()

re=[]
for i in range(10):
    record=sh.recvline().strip("\n")
    re.append(record[0:32]+my_sig+record[-32:])

for i in range(10):
    sh.sendafter(b'>', "provide a record".encode())
    sh.sendafter(b'>', re[i].encode())

sh.interactive()
```