



CTF Broadcast

原创

艺博东  于 2020-09-25 23:38:21 发布  10359  收藏 6

分类专栏: [网络攻防](#) 文章标签: [CTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/HYD696/article/details/108805993>

版权



[网络攻防](#) 专栏收录该内容

24 篇文章 17 订阅


订阅专栏

题目来源: 2019_Redhat










题目描述: 粗心的Alice在制作密码的时候, 把明文留下来, 聪明的你能快速找出来吗?

题目附件: 附件1

1、附件1










<input checked="" type="checkbox"/>	 21f64ce9e8054f61aa2d9c65abc4b...	2020-09-25 23:00	zip	5 KB
-------------------------------------	--	------------------	-----	------

2、解压后

板	组织	新建	打开	选择	
此电脑 > 桌面 : -- > 21f64ce9e8054f61aa2d9c65abc4b7c3					
名称	修改日期	类型	大小		
 BobCipher.enc	2019-09-27 21:40	Wireshark captu...	1 KB		
 BobPublic.pem	2019-09-27 21:40	PEM 文件	1 KB		
 CarolCipher.enc	2019-09-27 21:40	Wireshark captu...	1 KB		
 CarolPublic.pem	2019-09-27 21:40	PEM 文件	1 KB		
 DanCipher.enc	2019-09-27 21:40	Wireshark captu...	1 KB		
 DanPublic.pem	2019-09-27 21:40	PEM 文件	1 KB		
 ErinCipher.enc	2019-09-27 21:40	Wireshark captu...	1 KB		
 ErinPublic.pem	2019-09-27 21:40	PEM 文件	1 KB		
 task.py	2019-09-27 21:05	JetBrains PyChar...	1 KB		

<https://blog.csdn.net/HYD696>

3、选中“task.py”→打开

名称	修改日期	类型	大小		
 BobCipher.enc	2019-09-27 21:40	Wireshark captu...	1 KB		
 BobPublic.pem	2019-09-27 21:40	PEM 文件	1 KB		
 CarolCipher.enc	2019-09-27 21:40	Wireshark captu...	1 KB		
 CarolPublic.pem	2019-09-27 21:40	PEM 文件	1 KB		
 DanCipher.enc	2019-09-27 21:40	Wireshark captu...	1 KB		
 DanPublic.pem	2019-09-27 21:40	PEM 文件	1 KB		
 ErinCipher.enc	2019-09-27 21:40	Wireshark captu...	1 KB		
 ErinPublic.pem	2019-09-27 21:40	PEM 文件	1 KB		
 task.py	2019-09-27 21:05	JetBrains PyChar...	1 KB		

<https://blog.csdn.net/HYD696>

4、可直接找到flag

```
#!/usr/bin/env python3
from Crypto.Util import number
from Crypto.PublicKey import RSA
from hashlib import sha256
import json

#from secret import msg
msg = 'Hahaha, Hastad's method don't work on this. Flag is flag{fa0f8335-ae80-448e-a329-6fb69048aae4}.'
assert len(msg) == 95

Usernames = ['Alice', 'Bob', 'Carol', 'Dan', 'Erin']
N = [ ( number.getPrime(1024) * number.getPrime(1024) ) for _ in range(4) ]
PKs = [ RSA.construct( (N[0], 3) ), RSA.construct( (N[1], 3) ), RSA.construct( (N[2], 5) ), RSA.construct( (N[3], 5) ) ]
```

```

for i in range(4):
    name = Usernames[i+1]
    open(name+'Public.pem', 'wb').write( PKs[i].exportKey('PEM') )

    data = {'from': sha256( b'Alice' ).hexdigest(),
            'to' : sha256( name.encode() ).hexdigest(),
            'msg' : msg
          }
    data = json.dumps(data, sort_keys=True)
    m = number.bytes_to_long( data.encode() )

    cipher = pow(m, PKs[i].e, PKs[i].n)

    open(name+'Cipher.enc', 'wb').write( number.long_to_bytes(cipher) )

```

```

#!/usr/bin/env python3
import ...

#om secret import msg
msg = 'Hahaha, Hastad\'s method don\'t work on this. Flag is flag{fa0f8335-ae80-448e-a329-6fb69048aae4}.'
assert len(msg) == 95

Usernames = ['Alice', 'Bob', 'Carol', 'Dan', 'Erin']
N = [(number.getPrime(1024) * number.getPrime(1024)) for _ in range(4)]
PKs = [RSA.construct((N[0], 3)), RSA.construct((N[1], 3)), RSA.construct((N[2], 5)), RSA.construct((N[3],

for i in range(4):
    name = Usernames[i+1]
    open(name+'Public.pem', 'wb').write(PKs[i].exportKey('PEM'))

    data = {'from': sha256(b'Alice').hexdigest(),
            'to': sha256(name.encode()).hexdigest(),
            'msg': msg
          }
    data = json.dumps(data, sort_keys=True)
    m = number.bytes_to_long(data.encode())

    cipher = pow(m, PKs[i].e, PKs[i].n)

    open(name+'Cipher.enc', 'wb').write(number.long_to_bytes(cipher))

```

<https://blog.csdn.net/HYD696>

5. OK

flag{fa0f8335-ae80-448e-a329-6fb69048aae4}