[360第二届大学生信息安全]WriteUp-加密解密

原创

该账号不存在 于 2015-05-15 10:28:20 发布 3044 收藏
分类专栏： 加密解密

本文链接：https://blog.csdn.net/u012107165/article/details/45741407


加密解密 同时被 2 个专栏收录
1 篇文章 0 订阅
订阅专栏

网络安全
3 篇文章 0 订阅
订阅专栏

11题：

提示：将以下二进制解密获得通关密码

```
0101000000100101
100000011000001
```

解题攻略：

打开题目链接后发现是一堆二进制。首先选手需要写个二进制转16进制的脚本，然后winhex打开后发现是压缩文件，再另存为rar或者zip。打开zip文件发现是360图标的jpg，拖到txt里即可发现base64加密后的key，两次base64解密即可。

1.存在文本文件中的数据在JAVA FILE读入时都当做字符串来处理的。
就是说并不能直接将这个东西当做二进制数据使用，因此脚本的重心应该是如何编写脚本，将 **二进制串转换成二进制数据！！**

2.有个理解上的误区。
觉得byte类型很神秘无法操作。事实上完全可以把byte类型当做int使，就像可以把char当Int使一样的道理。比如之前会觉得byte temp += temp;这样的语句不可思议...

3. **字节** 是网络信息传输的单位。
字节 计算机信息技术用于计量存储容量和传输容量的一种计量单位，1个字节等于8位二进制。是一个很具体的存储空间。0x01, 0x45, 0xFA, ……
字符 人们使用的记号，抽象意义上的一个符号。 '1', '中', 'a', '$', '¥', ……

字符的传输是通过转换成其ASCII码对应的二进制的。一个英文字符对应一个字节。

因此，在题中的00001010是8个 字符

4.重复一下，脚本目的是：二进制字符串->二进制文件
产生的思路如下：字符->ascii码->二进制文件

要拿到二进制文件，我们得通过字节写入，因此可以用拿到的二进制字符串来模拟字节的8位

5.脚本如下：

```
package textToBin;

```

```
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;

public class TextToBin {

 public static void main(String[] args) throws IOException {
  // TODO Auto-generated method stub

  FileReader reader = new FileReader(new File("E:\\a.txt"));
  DataOutputStream dos = new DataOutputStream(new FileOutputStream(new File("E:\\b.bin")));

  textToBin(reader,dos);
 }

 private static void textToBin(FileReader reader, DataOutputStream dos) throws IOException {
  // TODO Auto-generated method stub
  int a;
  int i=0;
  char charArray[] = new char[8];
  byte temp = 0;
  while((a=reader.read())!=-1){
   if(i<8){

    charArray[i]=(char)a;
    i++;
    continue;
   }
   //处理足8位的字节

   for(int j=0;j<8;j++){
    temp += (byte)((charArray[j]-48)*Math.pow(2, 7-j));
   }
   dos.write(temp);
   //read()多读了一个字符
   charArray[0]=(char)a;
   i=1;
   temp=0;
  }
  //处理如果正好读完的情况，最后一次的数组没有写入
  if(i==7){
   for(int j=0;j<8;j++){
    temp += (byte)((charArray[j]-48)*Math.pow(2, 7-j));
   }
   dos.write(temp);
  }
  //处理不足8位的字节
  temp=0;
  for(int k=i;k<8;k++){
   charArray[k]='0';
  }
  for(int j=0;j<8;j++){
   temp += (byte)((charArray[j]-48)*Math.pow(2, 7-j));
  }
  dos.write(temp);
 }
```

```
}
```

尤其注意这里是如何表示byte类型的，实际上是用一个10进制int型进行的类型转换

6.换个角度想，现在有一个字节型temp,它是如何存储的?
通过二进制。那这个二进制如何表示成字符呢？
通过ASCII码。

所以倒过来仍是对的。

7.这道题中还有一个问题，通过查看winHex，你咋知道它是压缩文件？
关于这点...猜吧

8. **存ASCII码用byte[]**

1题：

```
   <div class="panel-body"><script language='javascript'>var qrivy = eval;NanylmrgurXrl="7D6A792B606E723629
```

加密解密第一题

解题攻略：

这道题目思路来源于之前比较流行的某款网马生成器，变量NanylmrgurXrl是某段js代码根据算法函数Xrlzrgubq()加密后的结果。选手调用Xrlzrgubq()函数即可将NanylmrgurXrl解出来。考察选手的js代码的阅读和动手能力。

首先将script标签中的内容拷贝到txt里，然后将function里内容拷贝出来放到script之间，然后将最后的nccy2vf(LbhT0gvg);换成alert(LbhT0gvg);保存为html，点击即可看到解密后的代码，分析代码得知key在key360目录下，打开key360目录后查看源码，发现rot13加密后的字符串，解密即可。

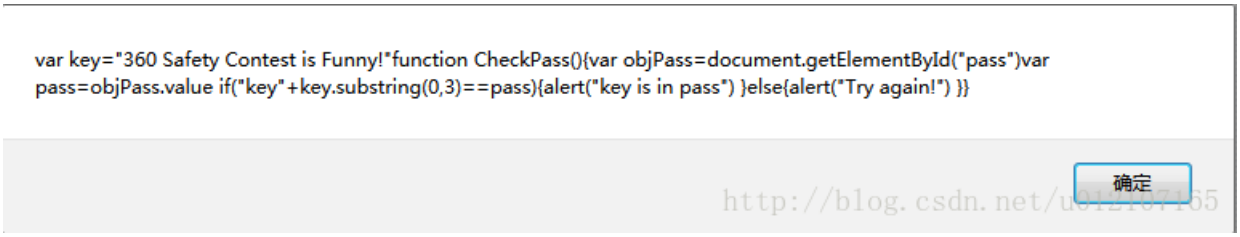
通过查看源代码，拿到这么一大串js脚本，key肯定和这么多乱七八糟的东西有关，分析一下这段js脚本：

```
<div class="panel-body">
<script language='javascript'>
var qrivy = eval;
NanylmrgurXrl="7D6A792B606E723629383D3B2B586A6D6E7F722B4864657F6E787F2B62782B4D7E6565722A296D7E65687F626465
ArrqrqSha="function Xrlzrgubq()"+
  "{qnauhnatcnv=Math.PI;cnefrVag=parseInt;sov='length';jebat0=cnefrVag(~((qnauhnatcnv&qnauhnatcnv)|(~qnauhn
  "rknz6znbm=lhn3afh<<lhn3afh;erf0hygVfabg=jebat0;LbhT0gvg='';jvxvqrp0qr=eval(unescape('%5'+'3%74%'+'72%69%
  "nccy2vf(LbhT0gvg);}"
  </script>
```

首先可以分段，一定注意分格式的时候 **字符串的链接**一定要用+，不能直接敲换行！！！！

```
<div class="panel-body">
<script language='javascript'>
var qrivy = eval;
NanylmrgurXrl="7D6A792B606E723629383D3B2B586A6D6E7F722B4864657F6E787F2B62782B4D7E6565722A296D7E65687F626465
ArrqrqSha="function Xrlzrgubq()"+
  "{qnauhnatcnv=Math.PI;cnefrVag=parseInt;sov='length';jebat0=cnefrVag(~((qnauhnatcnv&qnauhnatcnv)|(~qnauhn
  "rknz6znbm=lhn3afh<<lhn3afh;erf0hygVfabg=jebat0;LbhT0gvg='';jvxvqrp0qr=eval(unescape('%5'+'3%74%'+'72%69%
  "nccy2vf(LbhT0gvg);}";
function Xrlzrgubq()
  {qnauhnatcnv=Math.PI;cnefrVag=parseInt;sov='length';jebat0=cnefrVag(~((qnauhnatcnv&qnauhnatcnv)|(~qnauhna
  rknz6znbm=lhn3afh<<lhn3afh;erf0hygVfabg=jebat0;LbhT0gvg='';jvxvqrp0qr=eval(unescape('%5'+'3%74%'+'72%69%6
  alert(LbhT0gvg);};
Xrlzrgubq();
</script>
```

提取出function，修改最后为alert,调用Xrlzrgubq()函数。得到：



到360key下拿到

```
671rs4n35nsro3np2p3rrsoonn2612q2
```

根据js中提示，这是rot13，解密得key
671ef4a35afeb3ac2c3eefbbaa2612d2

1.字符串换行一定别忘+;.字符串换行一定别忘+;.字符串换行一定别忘+;

2.ROT13:
一种简易的置换暗码,该算法并没有提供真正的密码学上的保全，故它不应该被套用在需要保全的用途上。它常常被当作弱加密示例的典型

ROT13

套用ROT13到一段文字上仅仅只需要检查字元 字母顺序并取代它在13位之后 的对应字母，有需要超过时则重新绕回26 英文字母开头即可[2]。A换成N、B换成O、依此类推到M换成Z，然后序列反转：N换成A、O换成B、最后Z换成M。只有这些出现在英文字母里头的字元受影响；数字、符号、空白字元以及所有其他字元都不变。因为只有在英文字母表里头只有26个，并且26=2×13，ROT13函数是它自己的逆反：
对任何字元x： **ROT13(ROT13(x))=ROT26(x)=x**。
换句话说，两个连续的ROT13应用函式会回复原始文字（在数学上，这有时称之为 对合（involution）；在 密码学上，这叫做 对等加密（reciprocalcipher））。
转换可以利用 查找表完成，如下例所示：

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzabcdefghijklm

例如，下面的英文笑话，精华句为ROT13所隐匿：

How can you tell an extrovert from an introvert at NSA?Va gur ryringbef, gur rkgebireg ybbxf ng gur BGURE thl'f fubrf.

透过ROT13表格转换整片文字，该笑话的解答揭露如下：

Ubj pna lbh gryy na rkgebireg sebz na
vagebireg ng AFN?In the elevators,
the extrovert looks at the OTHER guy's shoes.

第二次ROT13函数将转回原始文字。

3.调用Js函数的方法：

```
<form>
<input type="button" onclick="Xrlzrgubq()";>
</form>
```

或直接在脚本中写：

```
Xrlzrgubq();
```

6.提示：下载图片获得通关密码

解题攻略：

将图片pass.gif下载后，后缀名改为rar，解压得到pass.txt 破解NTLM密文，得到通关密钥。

解压得pass.txt文件：
将 AAD3B435B51404EEAAD3B435B51404EE:DBDAAAC4D524F0DF9B34CCC255D061B5 解密后，与 e61e06202691107480213a6e369097d2 合并后作为通关密码.

通过http://www.hashkiller.co.uk/ntlm-decrypter.aspx破解NTLM密文

转：http://m.blog.csdn.net/blog/ask_man/41282331

密文类型	格式举例
md5	e10adc3949ba59abbe56e057f20f883e 49ba59abbe56e057 标准md5，32位或16位
md5(md5($pass))	b80c9c5f86de74f0090fc1a88b27ef34 第一次加密后，结果转换成小写，对结果再加密一次
md5(md5(md5($pass)))	e57941ff9000aedb44eb2fa13f6e3e3c 第一次加密后，结果转换成小写，对结果再加密一次,结果转换成小写，对结果再加密一次
MD5(MD5($pass))	bb7ff6177ee612ef9dc6acd3a9ea7ea9 第一次加密后，结果转换成大写，对结果再加密一次

MD5(MD5(MD5($pass)))	36d627bd562e83ab995fb1fdf59c95d9 第一次加密后，结果转换成大写，对结果再加密一次,结果转换成大写，对结果再加密一次
sha1	f03e8a370aa8dc80f63a6d67401a692ae72fa530 密文长度必须为40位
md4	c0a27f801162b8b862cd5f5a1a66e85a 32位
mysql	29596332026fd206 老MYSQL数据库用的，16位，且第1位和第7位必须为0-8
mysql5	b34c662f720236babfc1b3f75203a80e1009844a 新版本MySql数据库用的
md5($pass.$salt)	9393dc56f0c683b7bba9b3751d0f6a46:OTD6v4c8l3Zid2AL 在密码后附加一个字符串再加密。
md5($salt.$pass)	5610604c157ef1d0fb33911542e5b06f:zg 5610604c157ef1d0fb33911542e5b06f zg 在密码前附加一个字符串再加密。
md5(md5($pass).$salt); VB;DZ	30e23a848506770eca92faed1bd9f3ec:gM5 30e23a848506770eca92faed1bd9f3ec gM5 cd1a0b2de38cc1d7d796b1d2ba6a954f:dc2bce ad5f538296c0e05c26b85451fef9ea95:To!@35B%QS@)]U.DTy%fDm;SLwW58w 用于dz,vB等论坛程序，discuz的salt长度是6位，vBulletin的salt长度是3位或30位。
md5(md5($salt).md5($pass)) IPB	ac8dfc54ba110487b86ad6514328fd49:m@kZ} salt长度5位
sha1($salt.$pass)	9cea8c041ce88e0b2066343d819113005b80421c:2391 9cea8c041ce88e0b2066343d819113005b80421c 2391 用于SMF
Md5(Phpbb3)	H912345678Mw/BjmincvnSS94/STawW/ Linux
Md5(Wordpress)	PB12345678/c7bOMfLdQB9B/ypks8iB/ Linux
Md5(Unix)	$1$12345678$kbapHduhihjieYlUP66Xt/ Linux
Des(Unix)	af.kPXROLU9uY Linux
ntlm	71dd0709187df68befd20973fc23f973 Windows

Domain Cached Credentials	1aefd85a507965a6f1719e951b81d0f7 Windows
sha256	8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
sha256($pass.$salt)	1ec82d9b57403e53fafcf0ad8a86db196d135ef7513443a985385d7c20bdbfbd:abcdabcd
sha256($salt.$pass)	a6a4ccd14c6b21c63b8a0d38cfb7ead3e5032c58fdea7cd8a4da901db9462088:abcdabcd $sha256$abcdabcd$a6a4ccd14c6b21c63b8a0d38cfb7ead3e5032c58fdea7cd8a4da901db9462088

判断字符串加密类型是关键

16题：
提示：这是一道古典算法题，不过我们稍微改了改~，下面是两组明文密文对照，请尝试解开最终密文，提交即可获得通关密钥.

提示信息：

明文：I LIKE THIS GAME 私钥：THIS IS CTF 密文：FZAPCEFAZEPFK

明文：THE MORE YOU EAT THE MORE YOU FAT 私钥：THIS IS CTF 密文：QVWRMCCQVSTTZSTDFWQUSVUSAN

通关信息：

明文：?

私钥：ADLAB CTF

最终密文：BSVBUJCKCVWCTPMLL

明文：

加密解密第四题

解题攻略：

1.根据文字提示与图片搜索，得到古典算法 维尼吉亚密码

2.维尼吉亚密码可以理解为一个二维数组的映射，我们把维尼吉亚密码表想象为一个矩阵A[x,y]。

3.根据测试数据，我们可以得出我们修改后的映射关系为 A[x,y],y=3i+1，i为字符的位置

4.根据导出的映射关系结合通关信息推导出明文。

解得：XIAODONGANDREWZJL