# 76: Whitzard OJ逆向-Encrypt-Base encoding

原创

S1lenc3 于 2020-02-14 21:41:15 发布 137 收藏

分类专栏： CTF



本文链接：https://blog.csdn.net/qq_41858371/article/details/104320117





分析了代码，看了题目，猜出是base系列加密。

base64不对。

```
v20 = v17;
while ( j < n )
{
  v4 = v11;
  *((_BYTE *)s + v20) = byte_400EB0[v11[j]];
  *((_BYTE *)v26 + v20++) = byte_400EF0[v4[j++]];
}
*((_BYTE *)s + v20) = 0;
*v30 = v20 + 1;
if ( !strncmp((const char *)s, "D9", 2uLL)
  && !strncmp((const char *)s + 20, "Mp", 2uLL)
  && !strncmp((const char *)s + 18, "MR", 2uLL)
  && !strncmp((const char *)s + 2, "cS9N", 4uLL)
  && !strncmp((const char *)s + 6, "9iHjM", 5uLL)
  && !strncmp((const char *)s + 11, "LTdA8YS", 7uLL) )
{
  HIDWORD(v6) = puts("correct!");
}
```



分析了好半天，以为是base64换表加密，但还是搞不出来，想找writeup也找不到，然后随手搜了一下最终比较的字符串，没想到百度出来一模一样的字符串了。才知道这是base58，卧槽，base真牛逼。。。

然后直接base58解密就是flag。

base58的特征：

123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz

base58和base64的区别：

参考：https://xz.aliyun.com/t/2255

随后要搞一波加密，代码实现一遍，以后逆向的时候看见相似的逻辑就好办了。。。