360 CTF Writeup

原创

shadowblade123 于 2014-05-30 23:43:54 发布 4406 收藏
分类专栏： CTF 文章标签： 安全

0x00

小记这次360全国安全大赛，抱着试试看的心，捧回了西北区第一。真是，以后还要加油啊，不然太菜了被虐就不好玩了。

0x01

比赛两天，第一天是CTF类似的夺旗赛。王总的逆向就不说了，主要把我酱油的一些题说说，总结共享一下思路。

1，网络协议 10

模拟iphone6访问网页。

说实在一开始没想好是利用http哪一个头域，后来想了想是user-agent，然后就在各种实验这个还未出世产品可能的头域部分。老实说。。。360这道题太坑。。。完全不按规律出牌下面我截了一部分原先的user-agent头域作为参考

(3B92C18B-D9DE-4CB7-A02A-22FD2AF17C8F)
browser: Safari 7 operating system: iOS 7 primarily used on: tablet

Mozilla/5.0 (iPad; CPU OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Version/7.0 Mobile/11A465 Safari/9537.53
browser: Safari 7 operating system: iOS 7 primarily used on: tablet

Mozilla/5.0 (iPhone; CPU iPhone OS 6_1_4 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10B350 Safari/8536.25
browser: Safari 6 operating system: iOS 7 primarily used on: mobile

Mozilla/5.0 (iPad; CPU OS 5_1_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9B206 Safari/7534.48.3
browser: Safari 5 operating system: iOS 5 primarily used on: tablet

Mozilla/5.0 (iPhone; CPU iPhone OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Mobile/11A465 Twitter for iPhone
browser: Safari 7 operating system: iOS 7 primarily used on: mobile

Mozilla/5.0 (iPhone; CPU iPhone OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Version/7.0 Mobile/11A465 Safari/9537.53
browser: Safari 7 operating system: iOS 7 primarily used on: mobile

Mozilla/5.0 (iPhone; CPU iPhone OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Mobile/11A4449d Twitter for iPhone
browser: Safari 7 operating system: iOS 7 primarily used on: mobile

Mozilla/5.0 (iPad; CPU OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Mobile/11A465
browser: Safari 7 operating system: iOS 7 primarily used on: tablet

Mozilla/5.0 (iPad; CPU OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Mobile/11A465 Twitter for iPhone
browser: Safari 7 operating system: iOS 7 primarily used on: mobile

Mozilla/5.0 (iPhone; CPU iPhone OS 6_1_3 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10B329 Safari/8536.25
browser: Safari 6 operating system: iOS 6 primarily used on: mobile

可以看到前面mozilla 的版本都一样， safari浏览器的版本和ios几的版本决定os 后面的数字还有后面组件的序列号53x以及version号，依次类推，可能的ios8头域如下

Mozilla/5.0 (iPhone; CPU OS 8_0 like Mac OS X)AppleWebKit/538.51.1 (KHTML, like Gecko) Version/8.0 Mobile/11A465 Safari/9538.53

但是真没想到题目居然只是在iPhone后面加了一个6。。。。6。。。。做出来真是人品。

取证 10

这个就是一个数据包分析题，获取提交的文件名。因为题没了只能大致赘述，大概就是http200响应注意一下后面包中的内容就好了

朋友，密码是 8 位

你自己猜哦？

取证20 pngdecode

这题挺神奇的，当时倒是没做出来，也是思路错了，题目给了一个doc文档，然后打开后发现

以为是对这张图片处理，结果怎么看二进制也不知所以，后来知道一个知识，就是所有的doc文档或者docx文档都可以以zip或者rar打开。。真是孤陋寡闻。。。或者直接winhex这个doc文件都可以，然后发现其实这个doc隐藏了另外一张图片打开一看就知道密码。。。

加密解密 40

这题和上题差不多 不过这次给了一堆01010的二进制数，直接送去转码，发现又是一个zip文件，winhex打开发现有一个base64encode，解开就是密钥

加密解密 80

这题比想象的简单，一开始送去工具破解 发现不行

后来 观察了一下 发现 明文与密文字符数相同，又联系古典算法，猜测八成是逐个加密，然后列了一下明文，私钥还有密文的字母对应数，发现公式是

（明文字母序 + 私钥字母序 + 3 * 当前序号 %26 =密文序

举个例子，

比如第一个明文 第一个字母i是第九个字母，私钥T是20，密文F是6，那么关系就是20+9+3(第一个字母)=32，循环以后正好对应F也就是6，私钥是循环使用的，然后就开心的写程序跑就好了

代码很简单，就是反推

注意一点就是其中有一个字母反推正好是0结果程序返回的是@，楞了半天，后来只要替换成z就好了。

```cpp
#include <iostream>
#include <string>
#include <fstream>
using namespace std;

int main()
{
    ofstream cout("2.txt");
    string key = "ADLABCTF";
    string text = "BSVBUJCKCVWCTPMLL";
    string aim;
    for(int i = 0;i<text.length();i++)
    {
        aim += ((text[i] - 'A' +1 +26*10)-(key[i%key.length()]-'A'+1)-3*(i+1))%26+'A'-1;
    }
    cout<<aim<<endl;
    return 0;
}
```

在王总的光荣带领下 逆向除了160都做完了。。。实在威武霸气。。。然后我们就不明所以的拿到了第一名。。。王总威武

0x02

第二天是实战攻防，四台服务器，只做了两台，其中一台linux比较简单就能拿到rootshell，网页中提示了ssh端口222和密码2014-05-24.。。。图片里面隐藏了一个pcap包，打开是802.11wpa四次握手的通信包，解密也可以得到密码20140524.。。关键是后期相互之间阻止别人登录费了点时间，这里我用到了一个python脚本，一开始用的subprocess这个库不能用，估计是linux版本和python版本的问题。后来改成了os这个基础库就ok了，命令也比较简单，用的是pkill –kill –t pts/x

这里也现学习了一个知识，就是使用&和nothup

Unix/Linux下一般想让某个程序在后台运行，很多都是使用 & 在程序结尾来让程序自动运行。比如我们要运行mysql在后台：/usr/local/mysql/bin/mysqld_safe --user=mysql &

但是我们很多程序并不象mysqld一样可以做成守护进程，可能我们的程序只是普通程序而已，一般这种程序即使使用 & 结尾，如果终端关闭，那么程序也会被关闭。为了能够后台运行，我们需要使用nohup这个命令，比如我们有个start.sh需要在后台运行，并且希望在后台能够一直运行；

那么就使用nohup：

nohup /root/start.sh &

然后脚本就可以开心的运行，还可以继续深入的扫描内网。

附脚本：

```
import os,sys
import subprocess as subp
import re
import shlex
import time

while(True):
        pipe = os.popen("pkill -kill -t pts/1")
```

这里默认自己是0号客户