USTC2017 writeup

原创

Pz_mstr 于 2017-10-25 10:56:06 发布 552 收藏

文章标签： USTC2017 CTF writeup

本文链接：https://blog.csdn.net/qq_35544379/article/details/78338712

0x00 前言

突然发现了这个比赛，作为校外同学参加了一下科大的第四届校内赛，题目质量挺不错的，发下wp做下记录（部分思路来自官方题解）

1.看不见的字

这道题挺有意思的，各种常规misc做法轮番轰炸，结果发现使用Adobe Reader的高级搜索功能就可以找到flag

2.科大学生家长的日常

入入入门级web题，查看源码可以发现注释中的flag

3.被入侵的云端

尴尬，云游戏（被入侵的云端下）会做，这道题反而做不出来。
看了下官方题解
解法一：利用计算机网络知识分析packet，可以从中找到ip和端口

```
root@kali:~/ctf/output/jpg/output/jpg# binwalk packet

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
42            0x2A            Zip archive data, at least v2.0 to extract, name:
index.html
19426         0x4BE2          Zip archive data, at least v1.0 to extract, name:
__MACOSX/
19481         0x4C19          Zip archive data, at least v2.0 to extract, name:
__MACOSX/._index.html
19740         0x4D1C          Zip archive data, at least v2.0 to extract, name:
200-offline-sprite.png
53189         0xCFC5          Zip archive data, at least v2.0 to extract, name:
__MACOSX/._200-offline-sprite.png
53424         0xD0B0          Zip archive data, at least v2.0 to extract, name:
100-offline-sprite.png
53497         0xD0F9          PNG image, 1233 x 68, 8-bit grayscale, non-interla
ced
56158         0xDB5E          Zip archive data, at least v2.0 to extract, name:
__MACOSX/._100-offline-sprite.png
56944         0xDE70          End of Zip archive
```


分离出来一个zip，里面就是我们的云游戏（手动滑稽），那么第一个Zip数据是从0x2A开始的，而前面的数据是做什么的呢？
因为是分析发送ip和端口，那么我们可以猜测前面的数据就是包头。
使用WinHex打开packet，观察0x2A前的数据，并不能发现什么
我们可以尝试使用wireshark随便抓一个包进行比对分析

```
    Source: 192.168.0.9
    Destination: 220.181.105.184
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▲ Transmission Control Protocol, Src Port: 3609, Dst Port: 80, Seq: 1, Ack: 1, Len:
    Source Port: 3609
    Destination Port: 80
    [Stream index: 3]
    [TCP Segment Len: 1]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 2    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
```

```
0000  e4 6f 13 cb 7e 26 08 62  66 de 02 68 08 00 45 00   .o..~&.b f..h..E.
0010  00 29 35 56 40 00 40 06  fe 59 c0 a8 00 09 dc b5   .)5V@.@. .Y......
0020  69 b8 0e 19 00 50 cf 72  8c 65 7c f9 8e 89 50 10   i....P.r .e|...P.
0030  01 00 31 f0 00 00 00                              ..1....
```


IP地址在这里

```
▲ Transmission Control Protocol, Src Port: 3609, Dst Port: 80, Seq: 1, Ack: 1, Len: 1
    Source Port: 3609
    Destination Port: 80
    [Stream index: 3]
    [TCP Segment Len: 1]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 2    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
```

```
0000  e4 6f 13 cb 7e 26 08 62  66 de 02 68 08 00 45 00   .o..~&.b f..h..E.
0010  00 29 35 56 40 00 40 06  fe 59 c0 a8 00 09 dc b5   .)5V@.@. .Y......
0020  69 b8 0e 19 00 50 cf 72  8c 65 7c f9 8e 89 50 10   i....P.r .e|...P.
0030  01 00 31 f0 00 00 00                              ..1....
```


端口号在这里
那么我们这时候再分析一下我们的packet文件

```
 0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F      ANSI ASCII
52 54 00 12 35 02 FE ED  FA CE BE EF 08 00 45 00  RT  5 þíúÎ¾ï  E
DE 78 09 1D 00 00 FF 11  00 00 0A 00 61 CC 0A 00  Þx    ÿ    aÌ
61 FF 1F BB 1F BB DE 64  00 00 50 4B 03 04 14 00  aÿ » »Þd  PK
08 00 08 00 03 29 4E 4B  00 00 00 00 00 00 00 00      )NK
00 00 00 00 0A 00 10 00  69 6E 64 65 78 2E 68 74        index.ht
6D 6C 55 58 0C 00 87 2C  E1 59 35 2B E1 59 F5 01  mlUX  ‡,áY5+áYõ
14 00 DD 7D 6B 73 1C C9  71 E0 F7 FD 15 CD 55 68    Ý}ks Éqà÷ý ÍUh
```


可以得到IP地址0a 00 61 cc 端口号 1fbb（大端序：数据的高位字节存放在地址的低端 低位字节存放在地址高端）
即10.0.97.204:8123

解法二 直接利用wireshark打开packet，因为很多时候数据都是以二进制形式进行传输的，因此再导入前我们需要对packet进行一些处理，将它转为16进制再导入
od -Ax -tx1 -v packet >> packet_h

4.真假flag

在一堆flag中找出只有数字和字母的flag，脚本跑

```
import re
x=open("123.txt").read()
print re.findall(r'flag{\w*}',x)
```

官方的脚本如下

```
list(filter(lambda s: bytes(s[1:-1], 'utf-8').isalnum(), \
requests.get("http://hack.lug.ustc.edu.cn/file/flag.txt").text.s
plit('flag')))
```

5.骚扰你的一位老学长

好吧这题没做出来....细心程度不够
还尝试发送了邮件给这个老学长，满心欢喜以为会get flag

官方题解

打开
http://www.chenweikeng.com/
发现底部有 jemdoc 的链接，打开后查看 jemdoc 官网的文档得知元信息位于
http://www.chenweikeng.com/index.jemdoc
打开在底部可以找到 flag.

6.云游戏

浏览代码发现有一些没啥用的奇怪运算

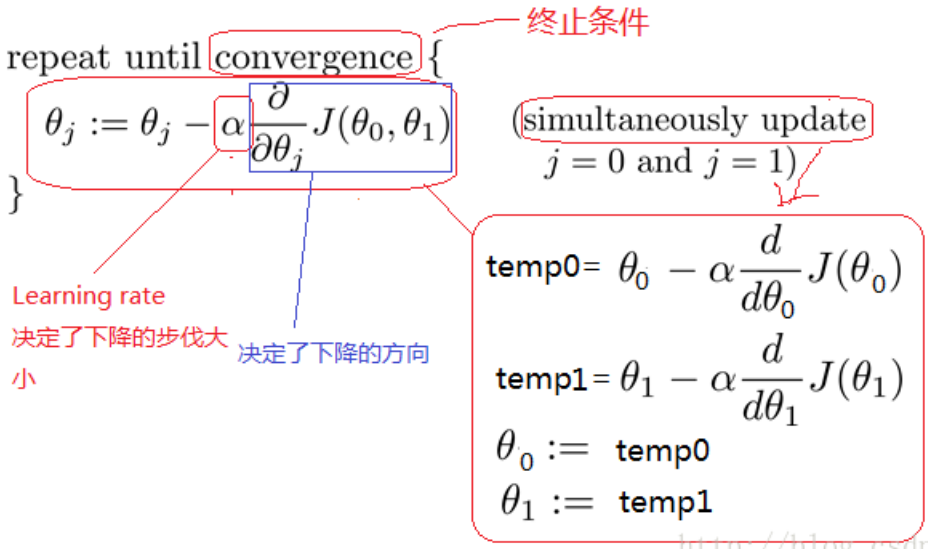
```
// try to gain reward, the constant part of the reward is 2.4657
676475
X -= 0.1* this.config.ACCELERATION * (4 * PARAM_A * X * X * X - 3
* PARAM_B * X * X);


//PARAM
ACCELERATION: 0.1
var PARAM_A = 0.943745363;
var PARAM_B = 2.895467371;
var X = 5.0;
```

作为一个高数差点挂科的家伙，我是不知道这个是梯度下降公式了，按照提示里的最优化算法，悄咪咪地搜索了一波才发现原来是梯度下降公式

方法：
(1)先确定向下一步的步伐大小，我们称为Learning rate；
(2)任意给定一个初始值：θ_0 θ_1；
(3)确定一个向下的方向，并向下走预先规定的步伐，并更新 θ_0 θ_1；
(4)当下降的高度小于某个定义的值，则停止下降；

算法：



所以梯度下降的公式也就是下面这个了

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta).$$

我们可以惊奇地发现，
X -= 0.1* this.config.ACCELERATION * (4 * PARAM_A * X * X * X - 3
* PARAM_B * X * X);
也就是公式中的J(θ)了
积分再加上提示中的常数得到
f(x) = PARAM_A * (x * *4) - *PARAM_B* (x ** 3) + 2.4657676475
之后注释掉小恐龙的游戏结束判断，得到稳定X值2.3010449782539832，代入计算即可得到
f（x） = -6.353530137073035
即flag{63535301}

7.被加密的实验报告

右键属性就可以看到flag了，小伙伴用解密器好像也能去掉密码，打开文档也可以得到flag

8.简单认证

抓包发现cookie是一个base64，解密为nobody
所以改成admin的base64结果即可

9.flag验证器

反汇编得到代码，进行算法分析可以简单出flag

10.黑客猜奇偶

前端对字符串进行了不能修改的限制，但可以通过抓包改，连续置空30次即可

11.熟悉的声音

电话拨号音

手机电话按键对应的双频				
	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

按这个进行翻译即可，发现使用#分割的ascii码值

16.永恒的东风

音频文件，用coolpro2软件打开就可以看到文件掺杂了额外的高频，简单出结果

剩下的题

剩下的题瞄了一眼
感觉都比较需要数学方面的分析，课程也比较紧，也就没有时间去尝试分析了