2018 强网杯 安恒杯 部分write up

转载

ayg14593 于 2018-03-27 20:49:00 发布 130 收藏
文章标签： php 操作系统 python
原文链接：http://www.cnblogs.com/P201521410044/p/8660055.html

- ## 蜘蛛侠啊

  安恒杯misc

- linux tshark identity 部分指令
- 附件下载 https://pan.baidu.com/s/1uo0GFufTjMqR8rtBzp5PlQ 密码 2cq5

```
0000  00 0c 29 23 98 99 00 50  56 c0 00 08 08 00 45 00   ..)#...P V.....E.
0010  00 66 60 b7 00 00 80 01  dc 0c c0 a8 be 01 c0 a8   .f`..... ........
0020  be 80 08 00 0f 51 0c 6e  01 00 24 24 53 54 41 52   .....Q.n ..$$STAR
0030  54 24 24 74 72 6a 42 46  33 6b 4a 6a 44 72 35 30   T$$trjBF 3kJjDr50
0040  49 65 39 53 41 47 65 38  39 51 72 79 4e 69 43 6d   Ie9SAGe8 9QryNiCm
0050  44 32 59 67 7a 49 6e 79  36 63 76 6e 67 43 7a 51   D2YgzIny 6cvngCzQ
0060  69 77 4b 4e 49 6d 71 33  6d 69 55 45 30 66 64 44   iwKNImq3 miUE0fdD
0070  4d 4e 6e 0a                                       MNn.
```

```
0000  00 50 56 c0 00 08 00 0c  29 23 98 99 08 00 45 00   .PV..... )#....E.
0010  00 66 46 94 00 00 40 01  36 30 c0 a8 be 80 c0 a8   .fF...@. 60......
0020  be 01 00 00 e1 c3 b8 a8  01 00 24 24 53 54 41 52   ........ ..$$STAR
0030  54 24 24 51 6b 33 6f 33  64 42 44 4e 66 53 75 43   T$$Qk3o3 dBDNfSuC
0040  62 30 62 65 73 66 51 71  78 67 51 4b 59 4b 61 52   b0besfQq xgQKYKaR
0050  69 38 47 42 41 55 45 44  55 32 4b 69 45 45 42 55   i8GBAUED U2KiEEBU
0060  51 46 44 37 30 6f 54 41  56 45 44 41 71 4b 69 67   QFD70oTA VEDAqKig
0070  6f 4a 69 0a                                       oJi.
```

首先打开附件pcap包 茫茫多的icmp包 稍微看一下似乎每个包内容都不太一样

但是 $$START$$ 似乎暗示了什么

看看最后一个ICMP包：

```
0000  00 50 56 c0 00 08 00 0c  29 23 98 99 08 00 45 00   .PV..... )#....E.
0010  00 3f d5 dd 00 00 40 01  a7 0d c0 a8 be 80 c0 a8   .?....@. ........
0020  be 01 00 00 ca ad 0c 6e  01 00 24 24 53 54 41 52   .......n ..$$STAR
0030  54 24 24 2d 2d 2d 2d 2d  45 4e 44 20 43 45 52 54   T$$----- END CERT
0040  49 46 49 43 41 54 45 2d  2d 2d 2d 2d 0a            IFICATE- ----.
```

基本可以判定这是传递了一个文件 因此我们的思路就是把这个文件提取出来

linux 的 tshark 命令可以实现这个功能 具体的参数可以度娘学习下

```
root@kali:~/ctf# tshark -r out.pcap -T fields -e data > 1.txt
```

```
2424535441525424242d2d2d2d2d424547494e20434552544946494341544452d2d2d2d2d0a
2424535441525424242d2d2d2d2d424547494e20434552544946494341544452d2d2d2d2d0a
2424535441525424242d2d2d2d2d424547494e20434552544946494341544452d2d2d2d2d0a
2424535441525424242d2d2d2d2d424547494e20434552544946494341544452d2d2d2d2d0a
2424535441525424245457344424251414141414149414b6d515445776c7343383457545546e414b3547445141494141414154a6d78685a79356e61575a6b7646645545307a62425a704f0a
2424535441525424245545734442425141414141414149414b6d515445776c7343383457545546e414b3547445141494141414154a6d78685a79356e61575a6b7646645545307a62425a704f0a
2424535441525424245545734442425141414141414149414b6d515445776c7343383457545546e414b3547445141494141414154a6d78685a79356e61575a6b7646645545307a62425a704f0a
2424535441525424245545734442425141414141414149414b6d515445776c7343383457545546e414b3547445141494141414154a6d78685a79356e61575a6b7646645545307a62425a704f0a
2424535441525424242516b336f336442444e6653754362306265736651717867514b594b6152693847424155454455324b6945454255514644373061544564544417146b69676f4a690a
2424535441525424242516b336f336442444e6653754362306265736651717867514b594b6152693847424155454455324b6945454255514644373061544564544417146b69676f4a690a
2424535441525424242516b336f336442444e6653754362306265736651717867514b594b6152693847424155454455324b6945454255514644373061544564544417146b69676f4a690a
2424535441525424242516b336f336442444e6653754362306265736651717867514b594b6152693847424155454455324b6945454255514644373061544564544417146b69676f4a690a
2424535441525424242657a6e666638376c575773755a7333467a467037356e6c6d37796e623174374779446834415767477944344456950484c536f71704b45697279596b6f69614d550a
2424535441525424242657a6e666638376c575773755a7333467a467037356e6c6d37796e623174374779446834415767477944344456950484c536f71704b45697279596b6f69614d550a
2424535441525424242657a6e666638376c575773755a7333467a467037356e6c6d37796e623174374779446834415767477944344456950484c536f71704b45697279596b6f69614d550a
2424535441525424242657a6e666638376c575773755a7333467a467037356e6c6d37796e623174374779446834415767477944344456950484c536f71704b45697279596b6f69614d550a
2424535441525424243549386456354848596c524d444e534e6a62544e6a5053737251307472557964485538364557784f2b56745a2b3973362b6a7136527a7351416768654d6662320a
2424535441525424243549386456354848596c524d444e534e6a62544e6a5053737251307472557964485538364557784f2b56745a2b3973362b6a7136527a7351416768654d6662320a
2424535441525424243549386456354848596c524d444e534e6a62544e6a5053737251307472557964485538364557784f2b56745a2b3973362b6a7136527a7351416768654d6662320a
2424535441525424243549386456354848596c524d444e534e6a62544e6a5053737251307472557964485538364557784f2b56745a2b3973362b6a7136527a7351416768654d6662320a
2424535441525424244d5a61574d57362b4d523675735147422f71366e593035486e6f74495349384b5434382f642f6e632b59784955676f784c4e6f6a314f47456c5a5775756261320a
2424535441525424244d5a61574d57362b4d523675735147422f71366e593035486e6f74495349384b5434382f642f6e632b59784955676f784c4e6f6a314f47456c5a5775756261320a
2424535441525424244d5a61574d57362b4d523675735147422f71366e593035486e6f74495349384b5434382f642f6e632b59784955676f784c4e6f6a314f47456c5a5775756261320a
2424535441525424247462477869365768 6f355757715a2f685352554e61326b4645776b5a724b61306d6f3243727132496b722b697569316132555a41326c39594956594f633045430a
2424535441525424247462477869365768 6f355757715a2f685352554e61326b4645776b5a724b61306d6f3243727132496b722b697569316132555a41326c39594956594f633045430a
2424535441525424246b792b7655326e68774e5138305849537838493573317a39426d7a6352346d684c364e495838695a63366d70497835426c63362b6465456839564 66e61314f690a
2424535441525424246b792b7655326e68774e5138305849537838493573317a39426d7a6352346d684c364e495838695a63366d70497835426c63362b6465456839564 66e61314f690a
2424535441525424246b792b7655326e68774e5138305849537838493573317a39426d7a6352346d684c364e495838695a63366d70497835426c63362b6465456839564 66e61314f690a
2424535441525424246b792b7655326e68774e5138305849537838493573317a39426d7a6352346d684c364e495838695a63366d70497835426c63362b6465456839564 66e61314f690a
2424535441525424242461394a53366a4f5443304f394c353779754568776487a43496438556c322b6b58366d6f3432766c357563523775743 54a6f61594542655a52496f68583735590a
2424535441525424242461394a53366a4f5443304f394c353779754568776487a43496438556c322b6b58366d6f3432766c357563523775743 54a6f61594542655a52496f68583735590a
2424535441525424242461394a53366a4f5443304f394c353779754568776487a43496438556c322b6b58366d6f3432766c357563523775743 54a6f61594542655a52496f68583735590a
2424535441525424242461394a53366a4f5443304f394c353779754568776487a43496438556c322b6b58366d6f3432766c357563523775743 54a6f61594542655a52496f68583735590a
2424535441525424242654a4663565a682f3679617a7137487461557650363737 4a72656e4654382b5764786665666e2f31646c6442536b564c7a54516b49434c76516d62526864546a0a
2424535441525424242654a4663565a682f3679617a7137487461557650363737 4a72656e4654382b5764786665666e2f31646c6442536b564c7a54516b49434c76516d62526864546a0a
```


将 hex 转为 ascii 码 发现内容有冗余 跑个脚本冷静一下

```
a=open('1.txt')
f=open('2.txt','w')
i=0
while i != 72156:
    b=a.readline()
    if i%4==0:
        f.write(f'{b}')
    i+=1
f.close()
```

跑完之后发现使用base64进行了加密 再来个脚本

```
import base64
lines = open('2.txt','rb').readlines()
flie = open('file','wb')
result = ""
for line in lines:
    flie.write(base64.b64decode(line.strip()))
```

文件最终是以zip格式生成的 解压后得到蜘蛛侠的特写gif

gif帧播放的很诡异 使用linux的 identity 命令查看一下

```
root@kali:~/图片# identify -format "%T" flag.gif
20505020505020502050202020502020202050502050205020502050505050502020505020202050
2050505020502020666root@kali:~/图片#
```

绝大多数帧播放时间都是20 或者50毫秒 将20看做0 50看做1

转化为二进制 再解码之后：

将这个字符串md5后即可提交flag

- **md5**

  强网杯web签到

- 题目环境 http://39.107.33.96:10000/

题目虽说是签到题 难度却丝毫不签到 而且从这道题可以学到不少哈希函数的姿势

- 第一部分

题目逻辑：

```
<!--
    if($_POST['param1']!=$_POST['param2'] && md5($_POST['param1'])==md5($_POST['param2'])){
            die("success!");
        }
-->
```

应对方案：php 0e开头的数字会当做科学计数法解析 因此只要传两组md5值开头为0e的即可

参考网站：https://www.cnblogs.com/Primzahl/p/6018158.html

- 第二部分

题目逻辑：

```
<!--
    if($_POST['param1']!==$_POST['param2'] && md5($_POST['param1'])===md5($_POST['param2'])){
            die("success!");
        }
-->
```

php为了针对0e错误 专门定义了=== 系列操作符 意思是两参数类型和值都相同才返回真 !== 是指两个参数的类型和值都相同时才返回假

因此这一问的意思是要构建两个参数 使得他们类型或者值不相同 但哈希值相同

应对方案：使用burpsuite 传两个值不相同的数组即可（php的md5函数如果参数不是字符串则返回一个null 而null===null）

- 第三部分

题目逻辑：

```
<!--
if((string)$_POST['param1']!==(string)$_POST['param2'] && md5($_POST['param1'])===md5($_POST['param2'])){
    die("success!);
}
-->
```

这下传数组大法使不了了 php这次将两个参数强行转成string

度娘了一个生成相同md5值的软件fastcoll

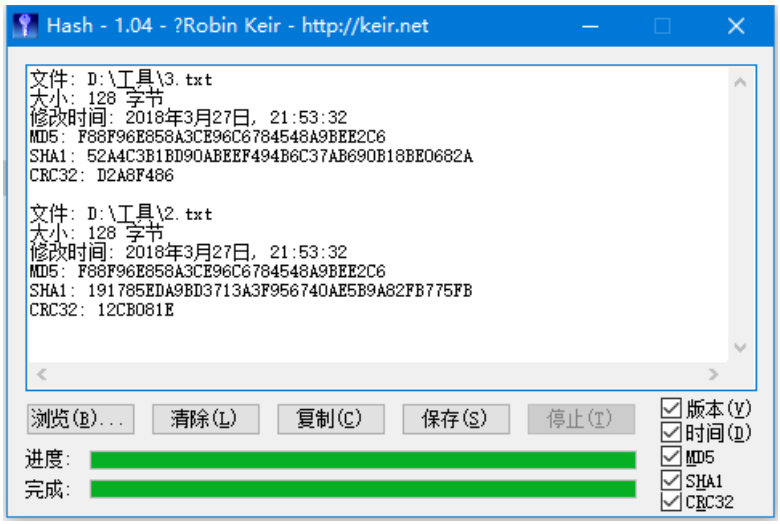
```
D:\工具>fastcoll_v1.0.0.5.exe -p 1.txt -o 2.txt 3.txt
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: '2.txt' and '3.txt'
Using prefixfile: '1.txt'
Using initial value: 0123456789abcdeffedcba9876543210

Generating first block: ...
Generating second block: S11.....................................................
Running time: 3.331 s

D:\工具>
```

检验下生成的两个文件



将两个文件url编码后上传到服务器即可得到flag

```python
#编码的python脚本
from urllib import parse
result = parse.quote(open('3.txt','rb').read())
print(result)
```

txt文件只哈希文件内容 不一样的内容哈希值基本不可能一样 fastcoll造出来的文件只能碰撞单一哈希函数（两文件其他的哈希值均不相同）
fastcoll产生的两个txt文件是加上文件头尾之后恰好哈希值相同，所以去除掉文件头尾（只哈希内容）哈希值就不会相同

转载于:https://www.cnblogs.com/P201521410044/p/8660055.html