

攻防世界 MISC

原创

dalssss 于 2019-10-04 01:51:15 发布 459 收藏

分类专栏: [MISC](#) 文章标签: [MISC](#) [攻防世界](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_42201260/article/details/102034300

版权



[MISC 专栏收录该内容](#)

4 篇文章 0 订阅

订阅专栏

功夫再高也怕菜刀

下载完附件, 发现是简单的流量分析。。

```
consent type: text/html; charset=UTF-8
-|./ 2017-12-08 11:42:11 0 0777
../ 2017-12-08 11:39:10 4096 0777
1.php 2017-12-08 11:33:16 33 0666
6666.jpg 2017-12-08 11:42:11 102226 0666
flag.txt 2017-12-08 11:35:29 17 0666
hello.zip 2017-12-08 09:32:36 224 0666
|<-
```

https://blog.csdn.net/qq_42201260

发现里面有个几个文件, 去kali下用foremost把它分离下。。

```
root@kali:/file# foremost 19ae303d520c4790b0401569b354e6a2.pcapng
Processing: 19ae303d520c4790b0401569b354e6a2.pcapng
|foundat=flag.txtC0000cS000J00Ea0v0
00&e$K002%0$00,0=0J001p00p46PK?
*|
root@kali:/file# ls
12.png 19ae303d520c4790b0401569b354e6a2.pcapng output
```

里面的压缩把进去是要密码的, 找到上传的数据包, 把jpg文件格式开头 (FFD8) 和结尾 (FFD9) 的图片数据copy出来, 导入到010ed中, 另存为jpg格式就出现压缩包的密码了。

Cephalopod

是到流量题。。放到wireshark里看看，发现了图片名为flag.png的图片

分组字节流	宽度	区分大小写	字符串	flag
61	2017-08-23 15:18:34.269458	10.0.2.10	10.0.2.7	Ceph 75 ACK
62	2017-08-23 15:18:34.308113	10.0.2.7	10.0.2.10	TCP 66 54924 → 6812 [ACK] Seq=1255 Ack=1583 Win=32768 Len=0 TSval=21190 TSecr=268656
63	2017-08-23 15:18:38.380203	10.0.2.7	10.0.2.10	Ceph 75 KEEPALIVE2
64	2017-08-23 15:18:38.380843	10.0.2.10	10.0.2.7	TCP 66 6812 → 54924 [ACK] Seq=1583 Ack=1264 Win=35456 Len=0 TSval=269684 TSecr=22208
65	2017-08-23 15:18:38.381257	10.0.2.10	10.0.2.7	Ceph 75 KEEPALIVE2
66	2017-08-23 15:18:38.381277	10.0.2.7	10.0.2.10	TCP 66 54924 → 6812 [ACK] Seq=1264 Ack=1592 Win=32768 Len=0 TSval=22208 TSecr=269684
70	2017-08-23 15:18:43.372323	10.0.2.7	10.0.2.10	Ceph 169 Client Session
71	2017-08-23 15:18:43.373461	10.0.2.10	10.0.2.7	Ceph 178 ACK Client Session
72	2017-08-23 15:18:43.373492	10.0.2.7	10.0.2.10	TCP 66 54924 → 6812 [ACK] Seq=1367 Ack=1704 Win=32768 Len=0 TSval=23456 TSecr=270932
73	2017-08-23 15:18:43.869082	10.0.2.7	10.0.2.10	Ceph 75 ACK
74	2017-08-23 15:18:43.618204	10.0.2.10	10.0.2.7	TCP 66 6812 → 54924 [ACK] Seq=1704 Ack=1376 Win=35456 Len=0 TSval=270994 TSecr=23508
75	2017-08-23 15:18:44.869235	10.0.2.7	10.0.2.10	Ceph 315 Client Request
76	2017-08-23 15:18:44.869781	10.0.2.10	10.0.2.7	TCP 66 6812 → 54924 [ACK] Seq=1704 Ack=1625 Win=36480 Len=0 TSval=271307 TSecr=23830
77	2017-08-23 15:18:44.877804	10.0.2.10	10.0.2.7	Ceph 769 ACK Client Reply
78	2017-08-23 15:18:44.877822	10.0.2.7	10.0.2.10	TCP 66 54924 → 6812 [ACK] Seq=1625 Ack=2407 Win=34176 Len=0 TSval=23832 TSecr=271308
99	2017-08-23 15:18:44.894464	10.0.2.7	10.0.2.10	Ceph 354 ACK Client Capabilities
100	2017-08-23 15:18:44.897685	10.0.2.10	10.0.2.7	Ceph 366 ACK Client Capabilities
102	2017-08-23 15:18:44.936086	10.0.2.7	10.0.2.10	TCP 66 54924 → 6812 [ACK] Seq=1913 Ack=2707 Win=35584 Len=0 TSval=23847 TSecr=271314
104	2017-08-23 15:18:45.108091	10.0.2.7	10.0.2.10	Ceph 75 ACK
106	2017-08-23 15:18:45.1145961	10.0.2.10	10.0.2.7	TCP 66 6812 → 54924 [ACK] Seq=2707 Ack=1922 Win=37632 Len=0 TSval=271376 TSecr=23890
107	2017-08-23 15:18:47.114642	10.0.2.7	10.0.2.10	Ceph 263 Client Request
108	2017-08-23 15:18:47.114848	10.0.2.10	10.0.2.7	TCP 66 6812 → 54924 [ACK] Seq=2707 Ack=2119 Win=38656 Len=0 TSval=271868 TSecr=24391
109	2017-08-23 15:18:47.115235	10.0.2.10	10.0.2.7	Ceph 366 ACK Client Capabilities
110	2017-08-23 15:18:47.115241	10.0.2.7	10.0.2.10	TCP 66 54924 → 6812 [ACK] Seq=2119 Ack=3007 Win=36992 Len=0 TSval=24391 TSecr=271868
111	2017-08-23 15:18:47.115366	10.0.2.10	10.0.2.7	Ceph 469 Client Reply
112	2017-08-23 15:18:47.115372	10.0.2.7	10.0.2.10	Ceph 75 ACK
113	2017-08-23 15:18:47.115514	10.0.2.7	10.0.2.10	TCP 272 ACK Client Request
114	2017-08-23 15:18:47.115689	10.0.2.10	10.0.2.7	TCP 66 6812 → 54924 [ACK] Seq=3410 Ack=2334 Win=39680 Len=0 TSval=271868 TSecr=24391
115	2017-08-23 15:18:47.115899	10.0.2.10	10.0.2.7	Ceph 1030 ACK Client Reply

Is Target: True
Trace, Size: 249, Data: 0100000000000000feffffffffffffffff00000000a00000...
Size: 249
Data: 0100000000000000feffffffffffffffff00000000a00000...
Extra, Size: 556, Data: 000000000000000000000000200000001010800000666c...
Size: 556
Data: 0000000000000000000000200000001010800000666c...

31a0 00 01 01 08 00 00 00 66 6c 61 67 2e 70 6e 67 00f lag.png.

丢到kali里用binwalk看看能不能分离，发现不行，在接着用foremost试试，还是不行，最后去看看下wp，发现是用这款“tcpxtract”。。

```
tcpxtract -f 40150e85ac1b4952f1c35c2d9103d8a40c7bee55.pcap Found file of type "png" in session
```

分离出两张图片，还是不能查看，但是放入winhex发现，它的头部少了89加上后保存，其中一张可以打开，出现了flag



hit-the-core

下载附件完发现是个.core文件(这他喵的是什么...)
百度了一下，发现是一个linux系统下程序崩溃生成的文件，里面有内存映射和存储调试信息的文件，主要是用来调试的。。

我丢到kali strings看了下，有点小收获，找到了一个疑似flag，但应该不是的字符串。。仔细一看发现“A”后面的大写字母都有规律的样子，我用脚本跑了一下，就出flag了

```
[ ]A[A]A A  
cvqAeqacltqazEigwiXobxrCrtuiTzahfFreqc{bnjrkWgk83kgd43j85ePgb_e_rwqr7fvbmHjkl03t  
ews_hmkogooyf0vbnk0ii87Drfgh_n kiwutfb0ghk9r0987k5tfb hjoiou087ptfcv}  
.*34"
```

```
string = "cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrkWgk83kgd43j85ePgb_e_rwqr7fvbmHjkl03tews_hmkogooyf0vbnk0ii8  
7Drfgh_n kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}"  
flag = ""  
for i in range(3, len(string),5):  
    flag += string[i]  
print(flag)
```