

DDCTF 2019 writeup Wireshark

原创

barzar 于 2019-04-20 01:35:25 发布 1232 收藏 4

分类专栏: [ctf](#) 文章标签: [wireshark](#) [ddctf](#) [ctf](#) [数据包](#)

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/zhang644720213/article/details/89411275>

版权



[ctf](#) 专栏收录该内容

2 篇文章 0 订阅

订阅专栏

DDCTF 2019 Wireshark

写在最前

分析数据包

分析文件

解码

写在最前

总的来说这是一道挺常规的Wireshark分析题, 一般的wireshark题的信息总是隐藏在http的url, html或者图片里面, 所以一般拿到Wireshark的题直奔http就行啦 要是不在http我就不承认我说过这句话

[Wireshark数据包下载地址](#)

分析数据包

拿到数据包先搜索http

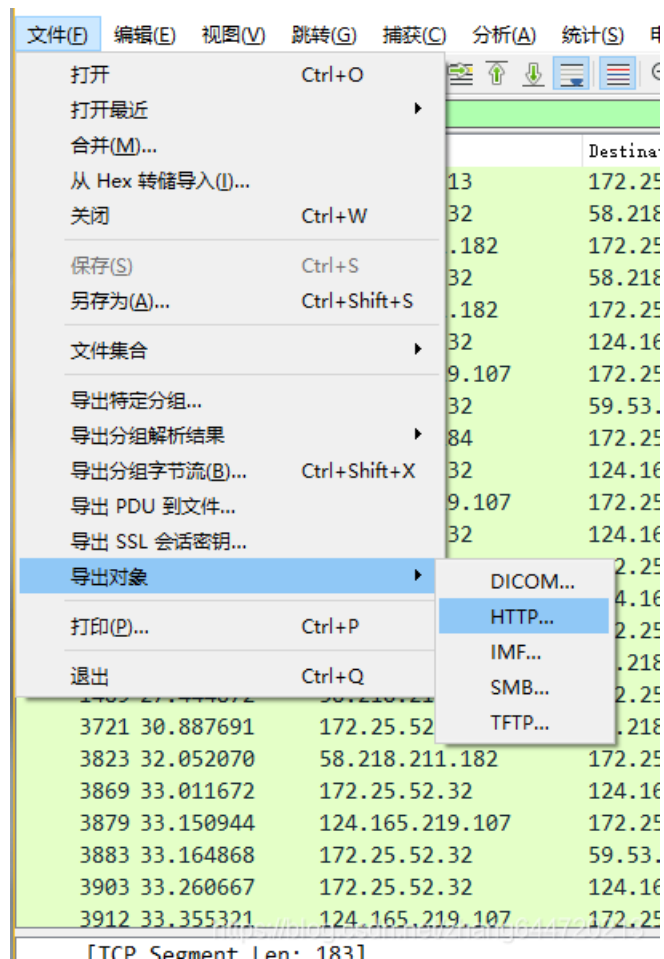
No.	Time	Source	Destination	Protocol	Length	Info
1051	19.279170	59.53.95.184	172.25.52.32	HTTP	329	HTTP/1.1 304 Not Modified
1056	19.293911	172.25.52.32	124.165.219.107	HTTP	901	POST /?c=User&a=getmessnum HTTP/1.1
1064	19.396442	124.165.219.107	172.25.52.32	HTTP	74	HTTP/1.1 200 OK (text/html)
1113	20.225362	172.25.52.32	124.165.219.107	HTTP	891	GET /upload HTTP/1.1
1119	20.392096	124.165.219.107	172.25.52.32	HTTP	652	HTTP/1.1 200 OK (text/html)
1240	22.061045	172.25.52.32	124.165.219.107	HTTP	891	POST /?c=User&a=getmessnum HTTP/1.1
1247	22.362860	124.165.219.107	172.25.52.32	HTTP	74	HTTP/1.1 200 OK (text/html)
1484	27.378232	172.25.52.32	58.218.211.182	HTTP	449	OPTIONS / HTTP/1.1
1489	27.444872	58.218.211.182	172.25.52.32	HTTP	455	HTTP/1.1 200 OK (text/json)
3721	30.887691	172.25.52.32	58.218.211.182	HTTP	1213	POST / HTTP/1.1
3823	32.052070	58.218.211.182	172.25.52.32	HTTP	657	HTTP/1.1 200 OK (json)
3869	33.011672	172.25.52.32	124.165.219.107	HTTP	891	GET /cf4a99fe55a59b82 HTTP/1.1
3879	33.150944	124.165.219.107	172.25.52.32	HTTP	1156	HTTP/1.1 200 OK (text/html)
3883	33.164868	172.25.52.32	59.53.95.184	HTTP	464	GET /674874/fd2f6f3479d1741es.png HTTP/1.1
3903	33.260667	172.25.52.32	124.165.219.107	HTTP	901	POST /?c=User&a=getmessnum HTTP/1.1
3912	33.355321	124.165.219.107	172.25.52.32	HTTP	74	HTTP/1.1 200 OK (text/html)
5227	34.540501	59.53.95.184	172.25.52.32	HTTP	1280	HTTP/1.1 200 OK (PNG)

通过分析发现http的数据包不是很多，并且还附带了几张png
看到这我的内心毫无波动甚至有点想笑

Protocol	Length	Info
P	455	HTTP/1.1 200 OK (text/json)
P	1226	POST / HTTP/1.1 (PNG)
P	656	HTTP/1.1 200 OK (json)
P	891	GET /ddc891b23147ba21 HTTP/1.1
P	1151	HTTP/1.1 200 OK (text/html)
P	545	GET /674874/7782abccd820677fs.png HTTP/1.1
P	529	HTTP/1.1 304 Not Modified
P	901	POST /?c=User&a=getmessnum HTTP/1.1
P	74	HTTP/1.1 200 OK (text/html)
P	891	GET /upload HTTP/1.1
P	652	HTTP/1.1 200 OK (text/html)
P	891	POST /?c=User&a=getmessnum HTTP/1.1
P	74	HTTP/1.1 200 OK (text/html)
P	449	OPTIONS / HTTP/1.1
P	455	HTTP/1.1 200 OK (text/json)
P	1213	POST / HTTP/1.1
P	657	HTTP/1.1 200 OK (json)
P	891	GET /cf4a99fe55a59b82 HTTP/1.1
P	1156	HTTP/1.1 200 OK (text/html)
P	464	GET /674874/fd2f6f3479d1741es.png HTTP/1.1
P	901	POST /?c=User&a=getmessnum HTTP/1.1
P	74	HTTP/1.1 200 OK (text/html)
P	1280	HTTP/1.1 200 OK (PNG)

28 bits) on interface 0

所以我们直接一键保存http的所有包再一个个分析



分析文件

通过查看提取出来的文件发现有那么几个可疑的文件，以及一张图片。我们把图片放一边，先查看一下那些文件，最后筛选出两个包含png的文件，以及一个html文件，提取一下html的url发现是个图片解密网站，而且需要密钥，于是猜想和文件中的两张png有关

图片解密网址

名称	修改日期	类型	大小
%3fc=User&a=getmessnum	2019/4/18 20:10	文件	1 KB
%3fc=User&a=getmessnum(1)	2019/4/18 20:10	文件	1 KB
%3fc=User&a=getmessnum(2)	2019/4/18 20:10	文件	1 KB
%5c	2019/4/18 20:10	文件	1 KB
%5c(1)	2019/4/18 20:10	文件	127 KB
%5c(2)	2019/4/18 20:10	文件	1 KB
%5c(3)	2019/4/18 20:10	文件	1 KB
%5c(4)	2019/4/18 20:10	文件	1,649 KB
%5c(5)	2019/4/18 20:10	文件	1 KB
cf4a99fe55a59b82	2019/4/18 20:10	文件	18 KB
ddc891b23147ba21	2019/4/18 20:10	文件	18 KB
fd2f6f3479d1741es.png	2019/4/18 20:10	PNG 文件	945 KB
img_add_info	2019/4/18 20:10	文件	25 KB
stat%3fappid=1018&ver=2.4.1.12&p...	2019/4/18 20:10	12&PEERID=005...	1 KB
upload	2019/4/18 20:10	文件	11 KB

https://blog.csdn.net/zhang644720213

```
<!DOCTYPE html><html lang="zh-cn"><head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>在线图片添加/解密隐藏信息(隐写术)工具 - 站长辅助工具 - 脚本之家在线工
</title>
<meta name="keywords" content="图片,图片信息,隐藏信息,图片隐写术,添加隐
息,解密隐藏信息,在线工具,图片工具,图片隐写术工具,图片添加隐藏信息工具,图片解
密信息工具" />
<meta name="description" content="这是一款可添加与解密图片隐藏信息的在
线工具,通过该工具,用户可以向图片中添加文字信息并重新生成图片,同样对于已经添
藏信息的图片也可以解密出之前添加的隐藏信息.提供给需要的朋友免费使用." />
<!-- Bootstrap -->
<link href="http://tools.jb51.net/static/skin/css/bootstrap.min.css" rel="stylesheet" />
<!-- Styles -->
<link href="http://tools.jb51.net/static/skin/css/theme.css" rel="stylesheet" />
<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
<script src="http://tools.jb51.net/static/skin/js/html5shiv.min.js"></script>
<script src="http://tools.jb51.net/static/skin/js/respond.min.js"></script>
<![endif]>
<!--[if IE]>
<script type="text/javascript">
window.jQuery || document.write("<script
src="http://tools.jb51.net/static/skin/js/jquery.min.js">"+<"+</script>");
</script>
<!-- <![endif]>
<!--[if IE]>
<script type="text/javascript">
```

我们用winhex打开那张唯一的png图片，与另外两个文件作比较，并且用winhex提取出两张png





<https://blog.csdn.net/zhang644720213>

解码

这里可以看到有一张钥匙一样的图片，那么可以确定解密密钥肯定在这张图片里面，本来打算拖进kali里面用binwalk分析一波，结果发现连图片都打不开，emmmm...m?

我脸上洋溢着灿烂的微笑，以我做了三四道图片隐写题的经验，这是png宽高爆破啊
直接拖进winhex，修改下高度就出来了

WinHex - [%5c(4).png]

文件(F) 编辑(E) 搜索(S) 导航(N) 查看(V) 工具(T) 专业工具(I) 选项(O) 窗口(W) 帮助(H) 19.7 x86 StrongWinHex

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI	ASCII
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG	IHDR
00000016	00	00	06	40	00	00	03	20	08	06	00	00	00	7B	C0	AE	@	{À@
00000032	5A	00	00	0C	14	69	43	43	50	49	43	43	20	50	72	6F	Z	iCCPICC Pro
00000048	66	69	6C	65	00	00	48	89	95	57	07	58	53	C9	16	9E	file	H%•W XSÉ Ž
00000064	5B	52	08	09	2D	10	01	29	A1	37	41	8A	74	E9	BD	08	[R	-) ;7AŠté:
00000080	48	07	1B	21	09	49	28	11	12	82	8A	1D	59	54	70	2D	H	! I(,Š YTp-
00000096	A8	58	B0	A2	AB	20	0A	AE	05	90	B5	62	57	16	C1	DE	`X°c«	® ubW AB

%5c(4).png
C:\Users\zkw6666\Desktop\新建
文件大小: 125 KB
128,088 字节
缺省编辑模式
状态: 原始的



key:57pmYyWt

<https://blog.csdn.net/zhang644720213>

然后直接去那个提取出来的网址里面解一下密

一、解密隐藏信息

1. 从电脑中选择一张带有隐藏信息的图片: %5c(1).png

2. 输入需要解开信息的密码 (如果没有密码可以不填):

图片中隐藏的信息为: flag+AHs-
44444354467B5145576F6B63704865556F32574F6642494E37706F6749577346303469526A747D+AH0-

<https://blog.csdn.net/zhang644720213>

得出来的一看就是一串16进制的ascii码 (因为最后一个为7D, 在ascii中为'}'字符, 所以很容易得出来)
直接base16一下就行了

Base16编码解码

44444354467B5145576F6B63704865556F32574F6642494E37706F6749577346303469526A747D

DDCTF{QEwokcpHeUo2WOfBIN7pogIWsf04iRjt}

<https://blog.csdn.net/zhang644720213>