ångstromCTF 2020

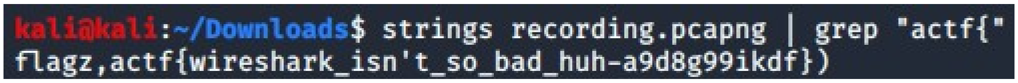
Misc : WS1

Description : Find my password from this recording (:

Attachment : recording.pcapng

Solution :

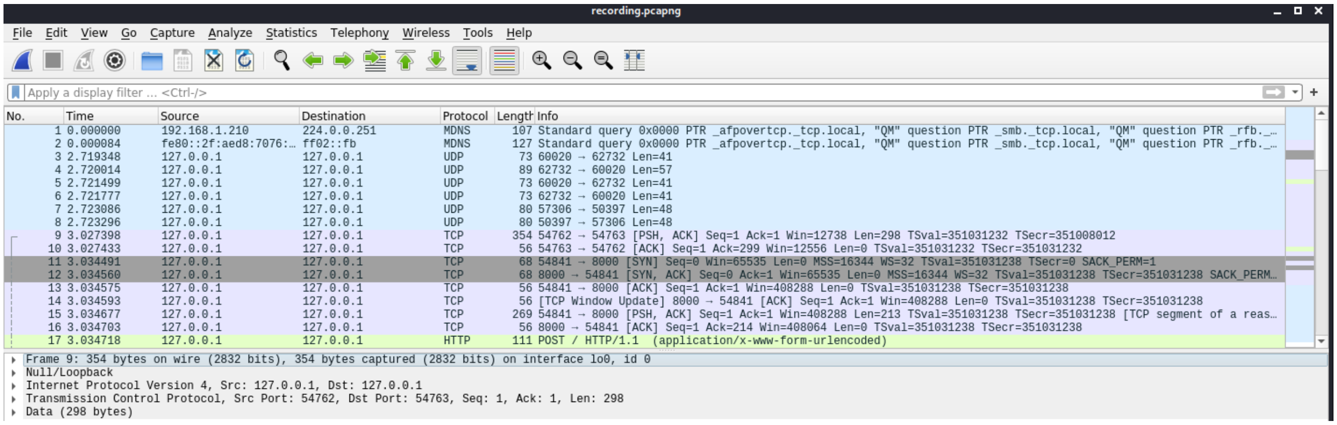
We can solve this challenge by only using “**strings**” and “**grep**” command for retrieve the flag.



```
kali@kali:~/Downloads$ strings recording.pcapng | grep "actf{"
flagz,actf{wireshark_isn't_so_bad_huh-a9d8g99ikdf})
```

But once we find the flag, we see that the intended way is using “**Wireshark**”. Start wireshark and load the file.

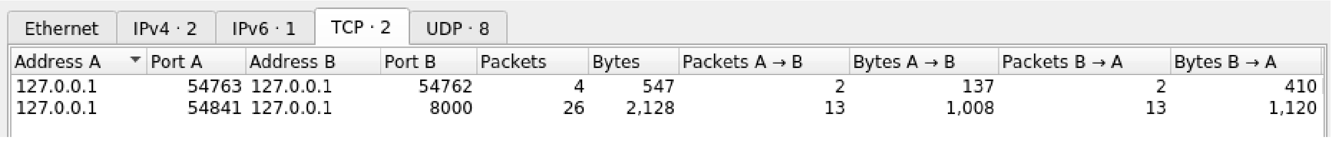
You will find something like that.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.210	224.0.0.251	MDNS	107	Standard query 0x0000 PTR _afpovertcp._tcp.local, "QM" question PTR _smb._tcp.local, "QM" question PTR _rfb._…
2	0.000084	fe80::2f:aed8:7076:…	ff02::fb	MDNS	127	Standard query 0x0000 PTR _afpovertcp._tcp.local, "QM" question PTR _smb._tcp.local, "QM" question PTR _rfb._…
3	2.719348	127.0.0.1	127.0.0.1	UDP	73	60020 → 62732 Len=41
4	2.720014	127.0.0.1	127.0.0.1	UDP	89	62732 → 60020 Len=57
5	2.721499	127.0.0.1	127.0.0.1	UDP	73	60020 → 62732 Len=41
6	2.721777	127.0.0.1	127.0.0.1	UDP	73	62732 → 60020 Len=41
7	2.723086	127.0.0.1	127.0.0.1	UDP	80	57306 → 50397 Len=48
8	2.723296	127.0.0.1	127.0.0.1	UDP	80	50397 → 57306 Len=48
9	3.027398	127.0.0.1	127.0.0.1	TCP	354	54762 → 54763 [PSH, ACK] Seq=1 Ack=1 Win=12738 Len=298 TSval=351031232 TSecr=351008012
10	3.027433	127.0.0.1	127.0.0.1	TCP	56	54763 → 54762 [ACK] Seq=1 Ack=299 Win=12556 Len=0 TSval=351031232 TSecr=351031232
11	3.034491	127.0.0.1	127.0.0.1	TCP	68	54841 → 8000 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=32 TSval=351031238 TSecr=0 SACK_PERM=1
12	3.034560	127.0.0.1	127.0.0.1	TCP	68	8000 → 54841 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=32 TSval=351031238 TSecr=351031238 SACK_PERM…
13	3.034575	127.0.0.1	127.0.0.1	TCP	56	54841 → 8000 [ACK] Seq=1 Ack=1 Win=408288 Len=0 TSval=351031238 TSecr=351031238
14	3.034593	127.0.0.1	127.0.0.1	TCP	56	[TCP Window Update] 8000 → 54841 [ACK] Seq=1 Ack=1 Win=408288 Len=0 TSval=351031238 TSecr=351031238
15	3.034677	127.0.0.1	127.0.0.1	TCP	269	54841 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=408288 Len=213 TSval=351031238 TSecr=351031238 [TCP segment of a reas…
16	3.034703	127.0.0.1	127.0.0.1	TCP	56	8000 → 54841 [ACK] Seq=1 Ack=214 Win=408064 Len=0 TSval=351031238 TSecr=351031238
17	3.034718	127.0.0.1	127.0.0.1	HTTP	111	POST / HTTP/1.1 (application/x-www-form-urlencoded)

```
▸ Frame 9: 354 bytes on wire (2832 bits), 354 bytes captured (2832 bits) on interface lo0, id 0
▸ Null/Loopback
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 54762, Dst Port: 54763, Seq: 1, Ack: 1, Len: 298
▸ Data (298 bytes)
```

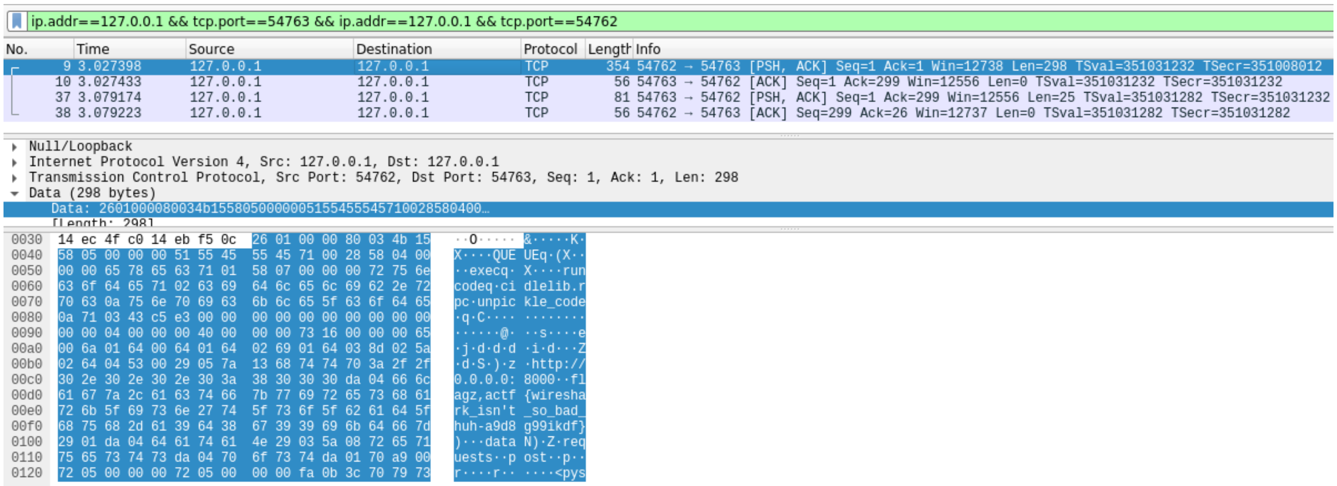
Inside “**Statistics > Conversations**”, you can find two TCP packet.



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
127.0.0.1	54763	127.0.0.1	54762	4	547	2	137	2	410
127.0.0.1	54841	127.0.0.1	8000	26	2,128	13	1,008	13	1,120

The one with the better timeline communication contain the flag, its the one which start from port **A (54763)** to port **B (54762)**. Right click on it and choose “**Apply as filter > Selected > A<→B**”.

The packet with the better length contain the flag.

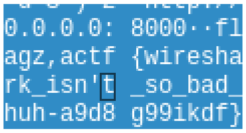


```
ip.addr==127.0.0.1 && tcp.port==54763 && ip.addr==127.0.0.1 && tcp.port==54762
```

```
No.  Time      Source     Destination Protocol Length Info
   9 3.027398  127.0.0.1  127.0.0.1   TCP      354 54762 → 54763 [PSH, ACK] Seq=1 Ack=1 Win=12738 Len=298 TSval=351031232 TSecr=351008012
  10 3.027433  127.0.0.1  127.0.0.1   TCP       56 54763 → 54762 [ACK] Seq=1 Ack=299 Win=12556 Len=0 TSval=351031232 TSecr=351031232
  37 3.079174  127.0.0.1  127.0.0.1   TCP       81 54763 → 54762 [PSH, ACK] Seq=1 Ack=299 Win=12556 Len=25 TSval=351031282 TSecr=351031232
  38 3.079223  127.0.0.1  127.0.0.1   TCP       56 54762 → 54763 [ACK] Seq=299 Ack=26 Win=12737 Len=0 TSval=351031282 TSecr=351031282
```

```
▸ Null/Loopback
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 54762, Dst Port: 54763, Seq: 1, Ack: 1, Len: 298
▾ Data (298 bytes)
    Data: 2601000080034b1558050000005155455545710028580400…
    [Length: 298]
```

```
0030  14 ec 4f c0 14 eb f5 0c  26 01 00 00 80 03 4b 15   ··O····· &·····K·
0040  58 05 00 00 00 51 55 45  55 45 71 00 28 58 04 00   X····QUE UEq·(X··
0050  00 00 65 78 65 63 71 01  58 07 00 00 00 72 75 6e   ··execq· X····run
0060  63 6f 64 65 71 02 63 69  64 6c 65 6c 69 62 2e 72   codeq·ci dlelib.r
0070  70 63 0a 75 6e 70 69 63  6b 6c 65 5f 63 6f 64 65   pc·unpic kle_code
0080  0a 71 03 43 c5 e3 00 00  00 00 00 00 00 00 00 00   ·q·C···· ········
0090  00 00 04 00 00 00 40 00  00 00 73 16 00 00 00 65   ······@· ··s····e
00a0  00 6a 01 64 00 64 01 64  02 69 01 64 03 8d 02 5a   ·j·d·d·d ·i·d···Z
00b0  02 64 04 53 00 29 05 7a  13 68 74 74 70 3a 2f 2f   ·d·S·)·z ·http://
00c0  30 2e 30 2e 30 2e 30 3a  38 30 30 30 da 04 66 6c   0.0.0.0: 8000··fl
00d0  61 67 7a 2c 61 63 74 66  7b 77 69 72 65 73 68 61   agz,actf {wiresha
00e0  72 6b 5f 69 73 6e 27 74  5f 73 6f 5f 62 61 64 5f   rk_isn't _so_bad_
00f0  68 75 68 2d 61 39 64 38  67 39 39 69 6b 64 66 7d   huh-a9d8 g99ikdf}
0100  29 01 da 04 64 61 74 61  4e 29 03 5a 08 72 65 71   )···data N)·Z·req
0110  75 65 73 74 73 da 04 70  6f 73 74 da 01 70 a9 00   uests··p ost··p··
0120  72 05 00 00 00 72 05 00  00 00 fa 0b 3c 70 79 73   r····r·· ····<pys
```



```
0.0.0.0: 8000··fl
agz,actf {wiresha
rk_isn't _so_bad_
huh-a9d8 g99ikdf}
```

Flag : actf{wirehsark_isn’t_so_bad_huh_a9d8g99ikdf}