Auto2 writeup

In this task we were given a file(traffic dump) sent to a car simulator called ICSIM. the dump was captured using candump tool , then I changed some values in it.

In order to solve this task you need to know which arbitration ID you are looking for in the dump and consider the order of actions (locking doors > right indicator > accelerator > left indicator), each data associated with the right frame sent is a part of the flag.

you need to play a bit with ICSIM and cantools(cansniffer,cansend,canplay,candump) .

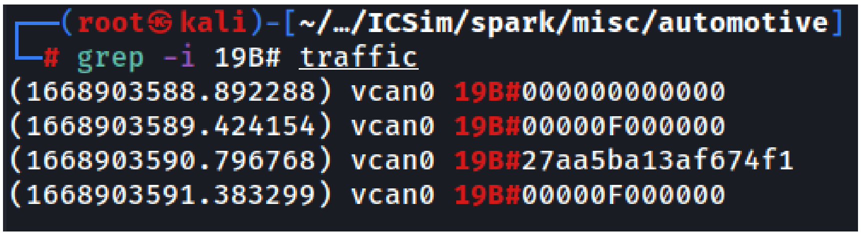
To determine an arbitration ID you can follow one of the two methods in this writeup this will help !!! : DFCTF — CAN Write-up. This is my write-up for the 'CAN'… | by Jawher Mastour | Nov, 2022 | Medium

You'll find that the accelerator ID is a bit tricky to determine, be patient and follow the procedure you'll get it !

Arbitration IDs:
188 : indicators
19B : locking/unlocking doors
244 : accelerator

Now we have the ID for the first action which is locking doors,let's look at it in the dump.

PART1 : data associated with locking doors signal :



```
┌──(root㉿kali)-[~/…/ICSim/spark/misc/automotive]
└─# grep -i 19B# traffic
(1668903588.892288) vcan0 19B#000000000000
(1668903589.424154) vcan0 19B#00000F000000
(1668903590.796768) vcan0 19B#27aa5ba13af674f1
(1668903591.383299) vcan0 19B#00000F000000
```

okay, here I sent 4 signals in total, but which one is part 1 of the flag ?

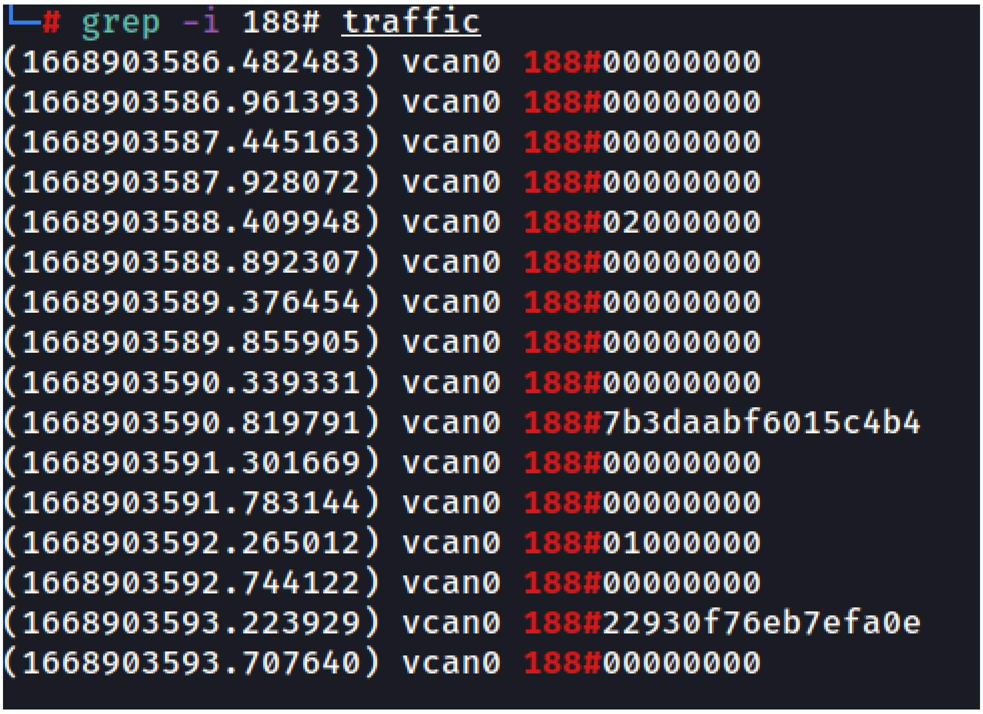
JUST LOOK AT THE WRONG SIGNAL ! , each action has a specific data .
19B#000000000000 to unlock all doors
19B#00000F000000 to lock all doors
19B#00000D000000 to unlock the right front door
and so on …

so the first part of the flag is ‘27aa5ba13af674f1‘ because its a wrong data .

PART 2 : data associated with turning-on the right indicator :

by the way, left and right indicator have the same ID but different values(keep that in mind)

no action : 188#00000000
left indicator : 188#01000000
right indicator : 188#02000000
anything else? maybe the flag .. let’s see



```
└─# grep -i 188# traffic
(1668903586.482483) vcan0 188#00000000
(1668903586.961393) vcan0 188#00000000
(1668903587.445163) vcan0 188#00000000
(1668903587.928072) vcan0 188#00000000
(1668903588.409948) vcan0 188#02000000
(1668903588.892307) vcan0 188#00000000
(1668903589.376454) vcan0 188#00000000
(1668903589.855905) vcan0 188#00000000
(1668903590.339331) vcan0 188#00000000
(1668903590.819791) vcan0 188#7b3daabf6015c4b4
(1668903591.301669) vcan0 188#00000000
(1668903591.783144) vcan0 188#00000000
(1668903592.265012) vcan0 188#01000000
(1668903592.744122) vcan0 188#00000000
(1668903593.223929) vcan0 188#22930f76eb7efa0e
(1668903593.707640) vcan0 188#00000000
```

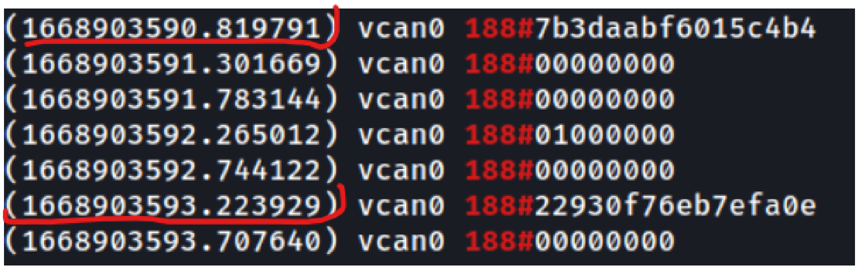
here we have two weird-looking values(remember to consider the order , so the right one is the first) ‘7b3daabf6015c4b4’
and the other weird signal is for the ***PART4 which is pressing the left indicators. ‘22930f76eb7efa0e’***

PART3 data associated with the acceleration:

you’ll get a lot of output here , where to look?
Its better to ask when to look(timestamp) and what are the acceptable values for accelerating(from - to)?

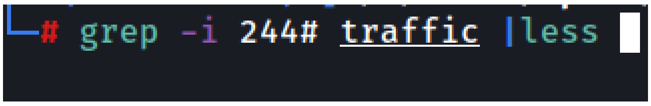
As i said , actions are ordered, so the value is between pressing the right indicator and the left indicator”

ID 188 :

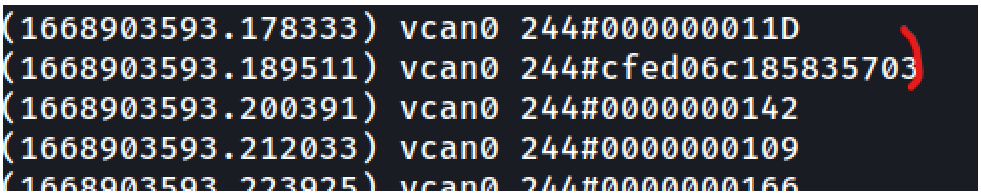


```
(1668903590.819791) vcan0 188#7b3daabf6015c4b4
(1668903591.301669) vcan0 188#00000000
(1668903591.783144) vcan0 188#00000000
(1668903592.265012) vcan0 188#01000000
(1668903592.744122) vcan0 188#00000000
(1668903593.223929) vcan0 188#22930f76eb7efa0e
(1668903593.707640) vcan0 188#00000000
```

ID 244 :



```
└─# grep -i 244# traffic |less
```



```
(1668903593.178333) vcan0 244#000000011D
(1668903593.189511) vcan0 244#cfed06c185835703
(1668903593.200391) vcan0 244#0000000142
(1668903593.212033) vcan0 244#0000000109
(1668903593.223925) vcan0 244#0000000166
```

‘cfed06c185835703’

<u>PART4: see part 2 ..</u>

and this is it !

sparkCTF{27aa5ba13af674f17b3daabf6015c4b4cfed06c185835703*22930f76eb7efa0e*}