

# TPMS

*SS*

*3/16/2017*

## Odyssey

From <http://opengarages.org/handbook/ebook/>, TPMS data can be exploited in the following ways:

- Send an impossible condition to the engine control unit (ECU), causing a fault that could then be exploited
- Trick the ECU into overcorrecting for spoofed road conditions
- Put the TPMS receiver or the ECU into an unrecoverable state that might cause a driver to pull over to check for a reported flat or that might even shut down the vehicle
- Track a vehicle based on the TPMS unique IDs
- Spoof the TPMS signal to set off internal alarms

This project uses the code at <https://github.com/jboone/gr-tpms> to capture TPMS data. The author's talk, using an earlier version of the code, can be found here: <http://www.youtube.com/watch?v=bKqiq2Y43Wg>. Previous research on the topic can be found at [https://web.wpi.edu/Pubs/E-project/Available/E-project-091115-154458/unrestricted/MQP\\_piscitelli\\_arnold\\_2015.pdf](https://web.wpi.edu/Pubs/E-project/Available/E-project-091115-154458/unrestricted/MQP_piscitelli_arnold_2015.pdf), and security vulnerabilities discussed in more depth at [http://www.winlab.rutgers.edu/~gruteser/papers/xu\\_tpms10.pdf](http://www.winlab.rutgers.edu/~gruteser/papers/xu_tpms10.pdf).

Distribution of tire IDs

```
##
## 10111001101001000101100110010100 11000010111001000011100110111011
##                                     1                               9
## 11000011100100001000011110101100 11001011110111101100110100011100
##                                     2                               1
## 11010011010000111111011000011100 11010011010001001001001111011100
##                                     1                               6
## 11010011101110000011101010011100 11010011101110000011101011011100
##                                     5                               4
## 11010110110110111100110100101100 11011000101101010010100000111010
##                                     1                               9
## 11011100110100100010110011001010 11011100110100100100010100001010
##                                     113                             140
## 11011101010011001111100101101010 11011101010011010001010111101010
##                                     116                             37
## 11011101011100010010000111001100 11100110100100011001001101001011
##                                     2                               10
## 11100111110101101010111110001010 11101000000100110001110100001011
##                                     8                               2
## 11101000010000110000011000011011 11101011101011010101110000001100
##                                     2                               3
## 11110001011101100001100101001010 11110001011101100011010000001010
##                                     4                               4
## 11110110100010111110100001111010 11111011101111100000111010101010
##                                     6                               6
## 11111011101111100001011100011010
##                                     6
```

First 3 bytes, statistical distribution

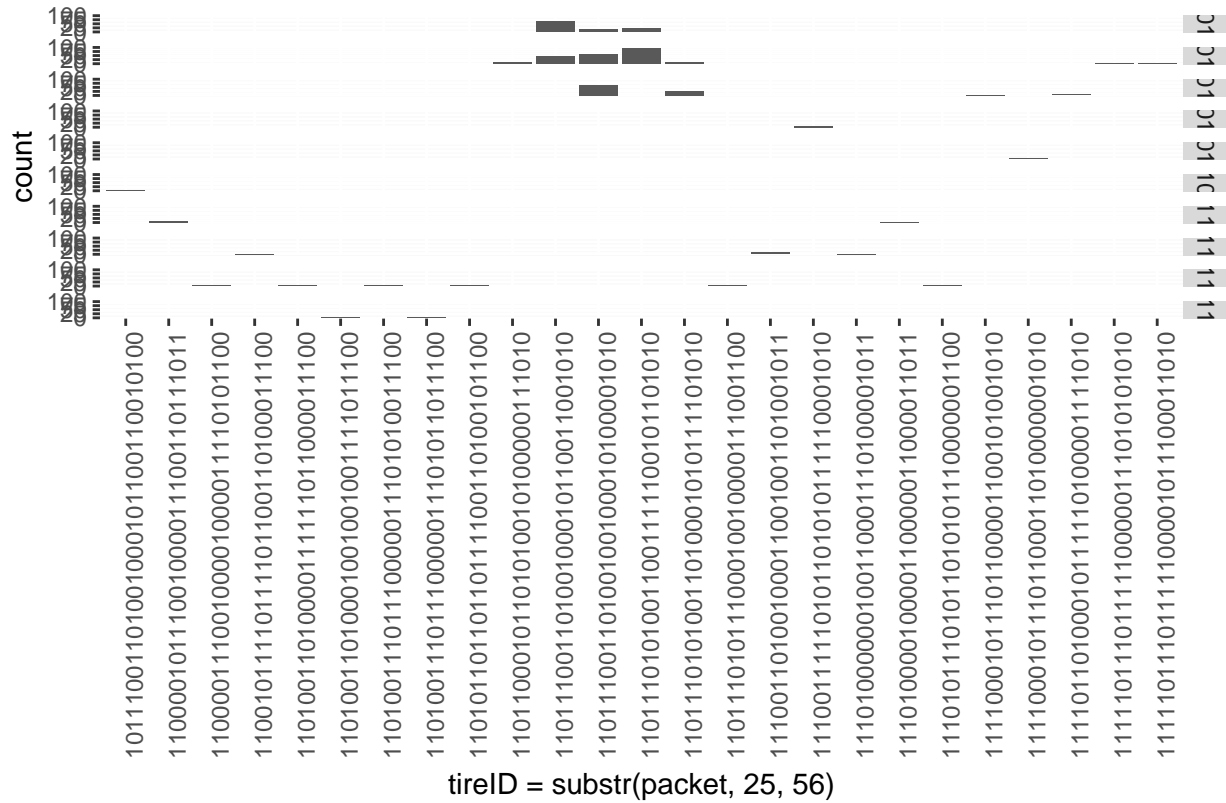
```
## Byte 1:
## byte
```

```

## 00001001 00001010 00001011 00001100 00001101 01010011 11111001 11111010
##      100      232      105      8      4      1      11      13
## 11111011 11111100
##      14      10
## Byte 2:
## byte
## 00000011 00000100 00010011 00010100 00100010 00100011 00100100 00110011
##      8      21      16      9      2      16      15      23
## 00110100 01000010 01000011 01000100 01010010 01010011 01010100 01100011
##      7      5      6      9      4      31      18      20
## 01100100 01100101 01110010 01110011 01110100 10000010 10000011 10000100
##      17      1      4      16      2      5      14      5
## 10010011 10010100 10100010 10100011 10100100 10110010 10110011 10110100
##      27      16      2      26      1      4      11      4
## 11000010 11000011 11000100 11010011 11100010 11100011 11100100 11110010
##      4      28      1      19      1      43      15      10
## 11110011 11110100
##      10      2
## Byte 3:
## byte
## 00001000 00001010 00001100 00011000 00011010 00011100 00011101 00101000
##      42      2      2      34      2      1      9      26
## 00101101 00111000 01001000 01011000 01011101 01101000 01111000 01111101
##      2      23      17      42      6      26      12      8
## 10001000 10001100 10011000 10011100 10101000 10101100 10101101 10110001
##      14      4      9      4      42      3      6      1
## 10111000 11001000 11001010 11011000 11101000 11101100 11111000
##      49      38      19      25      7      6      17

```

### Byte 1 Distribution By Tire ID



### Tire Distribution By Trip

