

- **Identify a Problem:** Vehicle Systems Security susceptible to hacking

- **Explore Existing Solutions:**
 - <https://www.federalregister.gov/documents/2022/09/09/2022-19507/cybersecurity-best-practices-for-the-safety-of-modern-vehicles#:~:text=ISO%2FSAE%2021434%20is%20a,of%20Modern%20Vehicles.%E2%80%9D%20Many%20comments>
 - “On January 12, 2021, NHTSA released its draft *Cybersecurity Best Practices for the Safety of Modern Vehicles* guidance”
 - Not many specifics of how to make vehicles more secure
 - The document, as described by many, was vague and didn’t allow for very much serviceability
 - Not user friendly for mechanics working on computer systems
 - Many complained about how there was little information on how to actually make the computer systems more secure; more of a “do this” rather than a “here’s how”

- **Justify Your Choice:** This problem very much interests me because first off, I absolutely love cars, always have, and second off, I love programming and solving issues - so two of some of my favorite things to do are mixed into one project!

Step 2. Design a Solution:

- **Propose a Software Solution:**
 - This software solution includes input validation, sanitation and authentication that is also extended to those who are maintaining vehicles, as we want this to be a program that can be edited by those who have access.
 - This is different than current solutions now, since these current programs are not verifying users properly, allowing anyone to change code. Current security systems do not rely on cybersecurity, especially older car models. We would want a system that can be extended to older cars as well; anything with a CPU can be affected.

- **Pseudocode:**

- The pseudocode for this program would realistically be much more extensive, considering you must also consider the actual writing of the entire CPU contents (which is thousands of lines long and has to do with the intricacies of the vehicle communication with the computer).

This would theoretically be in front of the command code for the vehicle.

1) Two step input validation:

- a) We want a function such as “getData” that would verify if the user inputs are correct and able to be processed as such.
- b) If user inputs “unknown data” (such as trying to insert code that does not match the language or the data type that a certain program is asking for), we want the program to catch this and put the user in a loop, until user enters “correct data.”
- c) If statement - if the user enters incorrect data more than three times, they will be locked out for a certain period of time. Perhaps something that can only be unlocked by a dealership or someone who has access.

2) Sanitation:

- a) We want a function that strips data and re-enters it how it is supposed to be formatted.
- b) If the unauthorized user does make it through the input validation loop, we want their data to be filtered through the program itself to match the style and contents of the program.
- c) For example, if someone is trying to modify a Tesla program by adding a function that says “driveAutonomously ()”, then the program will sift through its contents, compare if there are any matches for that function and “decide” that this is an illegal modification.
- d) We only want authorized users to be modifying this data - users who already know what functions are used in the program and are not trying to modify the drivability of the vehicle.

3) Authentication:

- a) Arguably the largest component. This part will allow only authorized users to be able to access the code and make changes.
- b) To make this as safe as possible, there will be various points within the code where you must access different passcodes to access various information.
- c) For example, if someone that owns the car wanted to add a modification to their car that changes the tune (how loud the car is, etc.), this is not a big deal and will not risk lives. This would be a separate access code than, say, for example, the code that deals with RPM's to acceleration ratio (which could affect how fast the car drives and/or could also blow up engines if not correct).

```

*untitled*

# Psuedocode for Final Lab

#First Part:

User opens file, upon opening, program launches into:
getData function ()
    if user enters "correct data"
        Proceed.
    else
        while loop that asks for information three more times
            "Please enter code edits in correct style, language, and data type:"
            if getData function () = False after three times of incorrect input:
                Break
                "Incorrect input. Program locked."

Close file - program is now inaccessible until authorized user unlocks it
Vehicle is disabled (this seems annoying, but is a safety measure).

#Second Part (after user enters "correct data"):

Strip code using .strip() method
Take out all characters deemed unnecessary by program (replace method)
Have program compare entered code to in-program code (likely in a file)
If statement:
    If there are no matches for words or functions listed in the entered code...
    that compares to the computer code, then:
        "Incorrect input. Program locked. Please see authorized user to continue."
    At this point, only authorized users (like a dealership) could modify code

#Third Part
input('Please enter which function you would like to access.')
    User inputs: 'EGR_Valve_Data()'
input('Please enter access code 1057a.')
if car owner has access to the data, it will be listed in car manual. If not:
    'Incorrect input. Please locate dealership.'

```

- **User Interaction:**

- In a perfect world, no user would need to interact with this program. Most people will not need to access data associated with their vehicle ECU. But, in the case that they do, it will be difficult. This program is not so much for the ease-of-use of users, but to make it only able to access by professionals who need to repair ECU issues. This would also solve the issue of cyber-attacks on vehicles, since they would have to go through a three-step process in order to break into someone's code. A lengthy and inefficient way to steal a car, or commit crimes.