RIDWAN HOSSAIN
100747897

2022/06/02

- I will continue to research possible resolutions to yesterday's ongoing issue. Hypothesis is that downloading the open source qt software ill allow me to easily install missing qt packages, so that is the step I am taking. The following sources may help:
    - https://forum.qt.io/topic/99649/install-serialbus-module/2
    - https://www.geopsy.org/wiki/index.php/Installing_Qt_binary_packages

- After installing the qt software, numerous additional libraries, such as the missing packages, were installed as well. Deleted the CANdevStudio folder and cloned the repository again, and after following the same steps in the build instructions on their GitHub, the cmake step worked successfully.

- The remaining portions of the CANdevStudio build took a significantly long time (~90 minutes).

- After installation, the CANdevStudio application is now runnable and the GUI is visible. The next step is to try to perform the 3 attacks Joelma had created on the Carla simulator using the CAN bus.

- Was able to send the attacks to the modified client Joelma made, now needed to be able to send them to the newest version of the client 'testbed.py'.

- Upon implementing the attacks into 'testbed.py', they were not working as intended. Whilst the effects of the attack are visible in the CAN interface, they are not affecting the vehicle in the simulator. It is not a case of reachability, as sample code in the statements ran correctly, therefore, it may be an issue of the reference to the vehicle. Looked into Carla documentation to look for possible resolutions.

- Discovered that the apply_control() method was not affecting the vehicle. Looked into other ways to control the vehicle in Carla.

- After **extensive** testing, found that something inside the **get_speed** function is causing the attacks to not be able to affect the vehicle. I will look further into that function and examine what specifically may be causing this issue.

- It was discovered that the lines "**c = world.player.get_control()**" and "**# p = world.player.get_physics_control()**" were the reasons the attacks could not control the vehicle. My assumption is that whenever the player's vehicle controls and physics are being accessed by these statements, which is all the time because the simulator is constantly acquiring the rpms, the statements corresponding to the attacks don't have access to the vehicle's physics and controls. As such, a global flag will be created which will be used to detect if an attack is incoming, and if so, temporarily disable the gauge cluster, allowing the attack statements to access vehicle physics and controls. This flag is called **attackFlag**.

- After some debugging and making some modifications to the attacks, I was successfully able to send those attacks to the Carla simulator through the CAN bus.

- The following attacks are sent in a looping fashion:
    - Turn the wheel completely to the left
    - Turn the wheel completely to the right
    - Apply full throttle

- With the modifications made to ‘testbed.py’, the attacks are now able to be sent to the Carla simulator through the CAN bus and successfully parsed on the client side.

- This concludes the integration of Joelma’s work on the CAN system into the new Carla bench. The next step is to review the previous capstone team’s work on the Jenkins based CI/CD framework and slowly begin to work on integrating that system into the new bench.