

Meeting Notes

We had sponsor meetings on: 1/24, 1/29, 2/5, 2/12, 2/19, 2/26, 3/5, 3/12

1/24

- Only using open source code
- Building models to understand behavior of malware
- Machine learning is two-part process, Learning then Inferencing
- Takes at least 3 people to teach and inference the model
- Building a platform for interactions
- Vertices are Principals (people)
- Edges are communications
- All interactions are limited to data scientists being one of the nodes
- Start studying jupyter notebook

SUMO

- Simulation of Urban Mobility
- Build network of roads and traffic

Veins

- Communication
- Uses SUMO as simulator

DSRC

- Emergency warning system for vehicles
- Cooperative Adaptive Cruise Control

Project: Extending Veins and SUMO

Veins and SUMO in C/C++

Extend SUMO:

- Develop various models to be attached to vehicle classes to generate required data.

Extend Veins:

- Add support for generating different DSRC (Dedicated Short-Range Communications) applications' messages and filling them by the data available through SUMO modules developed in SUMO

DSRC:

https://www.its.dot.gov/itspac/october2012/PDF/data_availability.pdf

SUMO sim

I'd recommend to capture below to-do lists if they are not captured currently:

- 1- Learn on how to extend Sumo and create new Device to be added as a module to a vehicle class or how to create a new Vehicle class type (Rogue)
- 2- Develop a very simple Device class with simple parameters to generate data and add it to a Vehicle class. Run a simulation including this module.
- 3- Learn different types of DSRC WAVE messages and their properties (message attributes). Design models (if needed) on how to abstract and design them in the DSRC device class.

Please document your experience as you successfully progress on both Sumo and Veins as it comes handy to share among all of us so that we can piggyback on eachother's learnings

1/29

- Enhance simulator, IOT transportation
- Real time data from network
- Simulate net models
- Inject threads to test models
- Data collection verification
- Transportation use case
- Model all the way to traffic patterns
- Safety priority
- Inject packets to trigger safety packets
- Contribute to open source

2/5

- What's the difference between smart car node class vs regular car node class?
- More specific definition of our end goal? Threat detection, DSRC threat data sharing among car nodes, merging data with subteam A
- Provide style interface to automate and facilitate jupiter communications
- Graphic bug potential fix, debug small too many artifacts
- Load VM fix running out of VRAM, use debugger or create a ticket

2/12

- TraCi would be the interface that allow you to play with a static configuration files.
- Static configuration is that static map of roads with traffic lights and fixed number of cars.
- Once you have those XML files, then you can import the configuration into TraCi and interact with it by the python interface (python script).

- A script that could possibly generate all the configuration files with given parameters.
- We need some sort of similar script similar to randomTrips.py to generate config files for us given the parameters.
- Figure if SUMO already has set of tools to be used for this purpose or where is it lacking
- Veins provides a middle-ware for coupling SUMO with OMNET++
- Veins, as part of the infrastructure, provides a C++ client library for the TraCI API.

2/19

TraCI protocol: A side script for communicating with the cars

- Create a behaved and an unbehaved car class
- Make the unbehaved car class configurable with individual intersections and stop signs (dont stop at certain intersections, stop at others, speed higher than the designated edge speed limit, etc.)
- Additionally - cars merge before they turn, create a situation in which a rogue car does not merge left before a lefthand turn, and turns left while in the righthand lane for example.
- Figure out how to event signal; detect event subscriptions and don't listen to the event handler while in rogue car class.
- Take out stopsign handlers
- TraCI sets flags and runtime
- Cars create events for other cars to consume, rogue cars can transmit out wrong events, multiple events, different vehicle ID's, can pretend it is 3 cars.
- The simulation logs should capture this info, and we should be able to indicate a wrong event on the simulation log even though other cars in the simulation will not know it is a wrong event.
- Be aware of protocol writes, write packets, build packets, so that they can be transmitted within veins.
- Fill DSRC protocol messages
- SUMO just generates the sim, the protocol messages are within the transmissions between Veins and SUMO.

2/26

- The models that are not in SUMO, figure out simplest way to showcase parameters(simple physics)
- 100 cars in specific map, scale up 1000, select 10 to be rogue.
- (Sits on top of SUMO)
- You can use Traci
- Randomly/dynamically assign rogue characteristics to a car
- Followback Traci script/variables and see how it access the sumo source code backend
- Implement traci functions that call rogue

3/5

- Confirm next meeting before finals week or before spring break
- Rogue has different characteristics, can it occur at different parts of the roadway,
- Can Rogue only have few specific rogue characters but not all?
- The pros and cons (code complexity) rogue be configured
- Easier to have one big rogue class with different characteristic sets or multiple rogue classes
- Traci script in python and change the behavior not source code
- DSRC messages in packets
- Insert fields/parameters based on message ID of the DSRC packet the encapsulation is different
- Encoders SUMO messages to DSRC packets
- BSM needs and adding that (figure out how to add it) and then figure out accurately to model the heading

3/12

- Backend should work even if there are visualization errors.
- Grab IDs to connections/4way intersections/artifacts to use in TraCi script
- Do not have to deal with real time issues
- Do not focus on the pixels/coordinates as it is unreliable
- Do not focus on dynamically controlling the simulation due the difficulty, as a replacement use time; ie: first 5 mins rogue car
- Work on combining DSRC and Traci script
- DSRC data generation and rogue car scenario make match
- DSRC formatting; possibly use wireshark and show network packet format, payload. http will have dsrc/ip wave
- Packetize the data, to match data on wire using their code pcaps
- Proprietary packets based on XML data
- JSON
- Scenario
- Compress the time scenario using time stamps
- Not real time ie run simulation for 5 hours and compress into a 5 minute video.
- Talk to Project guppi to remotely show simulation using:
- VLC x11 server