Université Gustave Eiffel

PROJECT REPORT:

Multi-Agent Systems - VANET Cybersecurity simulation

Tran Le Tuan NGUYEN, Ian COLLET, Mahdi ZARGAYOUNA[1,*]

[1] Université Gustave Eiffel, 77420 Champs-sur-Marne, France

*Supervisor. Email: mahdi.zargayouna@univ-eiffel.fr

Students: {tran-le-tuan.nguyen, ian.collet}@univ-eiffel.fr

Abstract ***- This report presents the design and implementation of a multi-agent system that simulates the propagation of a cybersecurity attack on connected vehicles in a Vehicular Ad Hoc Network (VANET). The system models the movement of vehicles on a grid and examines their states, including not infected, infected, repaired, and broken down. The objective of the simulation is to explore various movement scenarios and percentages of initially infected vehicles. To accomplish this, the system incorporates system parameters, an environment representation, movement scenarios, and vehicle behaviours. Additionally, the report highlights the achieved features, including a graphical user interface (GUI) for animation and a plot illustrating the evolution of infected, not infected, repaired, and broken-down vehicles***

I. INTRODUCTION

The increasing reliance on connected vehicles in modern transportation systems has opened new avenues for potential cybersecurity threats [1]. Addressing these threats requires understanding the dynamics of a cybersecurity attack in a Vehicular Ad Hoc Network (VANET) and evaluating countermeasures [2]. In this context, this report presents a simulation-based approach to study the propagation of a cybersecurity attack among connected vehicles.

The primary goal of the simulation is to design and implement a multi-agent system that accurately represents the behaviour of vehicles in response to a cybersecurity attack. The system considers various vehicle states, such as not infected, infected, repaired, and broken down, and examines the probabilities of infection, repair, and breakdown. By defining system parameters, representing the environment, incorporating movement scenarios, and defining vehicle behaviours, the simulation enables the exploration of different attack scenarios and their impact on the network.

One crucial aspect of the simulation is the representation of the environment and movement scenarios. The system can simulate random movements of vehicles or incorporate central attractors, such as malls, schools, workplaces, and other key locations,

to observe variations in the attack propagation. Furthermore, different percentages of initially infected vehicles can be simulated to analyze the effects of varying attack intensities.

To provide a comprehensive analysis, the simulation incorporates a graphical user interface (GUI) that enables the visualization of the movement of agents in real-time. The GUI allows for the animation of vehicle movements, providing a visual representation of the attack propagation process. Additionally, a plot is generated to illustrate the evolution of the different vehicle states over time, including the number of infected, not infected, repaired, and broken down vehicles. This feature facilitates a clear understanding of the simulation results and aids in identifying trends and patterns.

II. SYSTEM DESIGN

This section will discuss the environment representation of the simulation attack, the vehicle's states and behaviours.

A. ENVIRONMENT REPRESENTATION

The environment representation in the VANET cybersecurity simulation is crucial for simulating the movement and interaction of vehicles within the network. The environment is structured using classes such as Environment, Network, and Cell, which collectively create a realistic environment for the simulation.

The main class responsible for managing the environment is the Environment class. Upon initialization, it creates and configures the network and vehicles based on the provided parameters. The network is represented by the Network class, which consists of individual cells representing different locations in the simulated environment. Each cell is an instance of the Cell class, which contains information about its position, type, neighboring cells, the vehicle present in the cell, and its attractivity probability.

The Network class maintains a list of cells and provides methods to add cells to the network and connect cells as neighbors. The add_cell method allows for the addition of individual cells to the network, while the connect_cells method establishes neighbor relationships between cells. These connections enable vehicles located in adjacent cells to interact and influence each other's states during the simulation.

The Cell class represents a specific location within the network and encapsulates various attributes. Each cell is defined by its x and y coordinates, type, and attractivity probability. The attractivity probability represents the likelihood of a vehicle in a neighboring cell being influenced by the current cell. Additionally, the Cell class maintains a list of neighboring cells, allowing for efficient communication and interaction between vehicles located in adjacent cells.

The Environment class also creates the vehicles based on the provided configuration. Each vehicle is represented by the Vehicle class, which contains information such as the vehicle's ID, position, and initial state (not infected, infected, repaired, or broken down). The vehicles are stored in a list within the Environment class, allowing for easy access and tracking of their states throughout the simulation.

To facilitate the analysis and visualization of the simulation results, the Environment class includes a tick_stats dictionary. This dictionary captures statistics for each simulation tick, specifically tracking the number of vehicles in different states (infected, not infected, repaired, and broken down). These statistics serve as valuable insights into the propagation of the cybersecurity attack and allow for the observation of trends and patterns over time.

By representing the environment using classes such as Environment, Network, and Cell, the simulation accurately models the movement, interaction, and state evolution of vehicles in a VANET. This environment representation forms the foundation for studying the propagation of a cybersecurity attack and evaluating various scenarios and countermeasures within the simulation.

B. VEHICLE STATES AND BEHAVIOURS

In this simulation, vehicles operate on a grid and can exhibit different states: Not infected, Infected, Repaired, or Broken down. Each vehicle is represented by a unique vehicle ID and has a position on the grid.

The behavior of the vehicles is governed by certain probabilities. When a not infected vehicle interacts with an infected vehicle, it has a chance of being infected based on the probability ρinf. An infected vehicle, on the other hand, can be repaired with a probability of ρrep or broken down with a probability of ρbreak.

The code provided represents the implementation of the vehicle class and its behaviors. Each vehicle object has attributes such as its ID, position, and state. The state can be accessed through the get_state() method.

The simulation considers the interactions between vehicles within the network. For each vehicle, the code checks its state and neighboring cells in the grid. If the vehicle is not broken down, it iterates through neighboring vehicles and examines their states. If the current vehicle is not infected or repaired and encounters an infected vehicle, it has a chance of becoming infected based on the given probability ρinf. This infection probability is evaluated using the random.random() function.

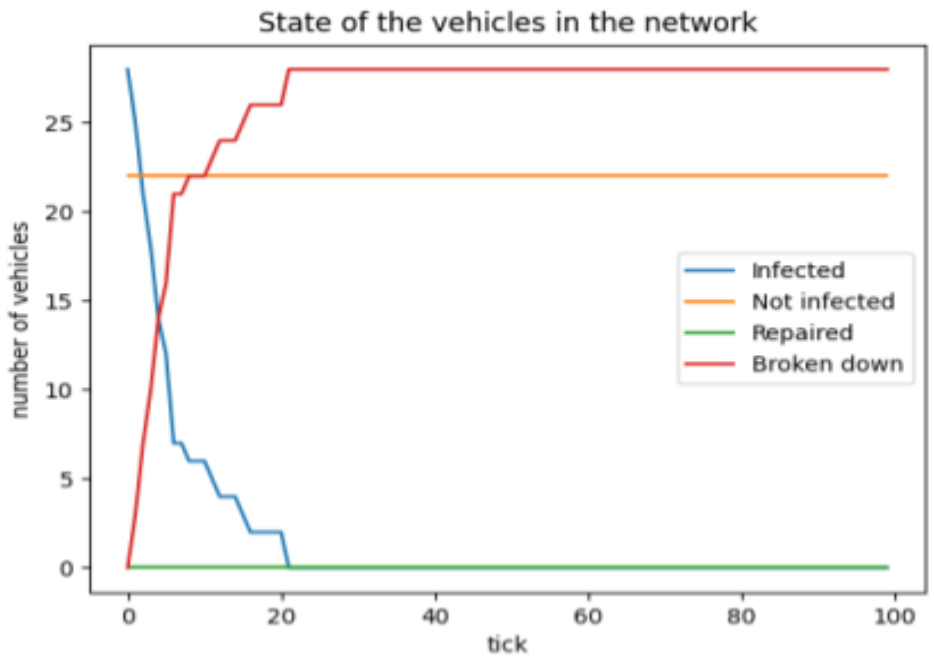
If a vehicle is already infected, the code generates a list of possible outcomes for the vehicle's state change. The probabilities of repair (ρrep), breakdown (ρbreak), and remaining infected states are taken into account to create this list. The choice of the next state is made randomly from this list using random.choice().

III. SIMULATION RESULTS

In this section, we define the system parameters with a configuration table as follows:

Parameter	**Description**
network_size	Size of the simulated network (100 units in the x-axis and 100 units in the y-axis)

total_ticks	Duration of the simulation (100 discrete time steps)
probabilities	Probabilities associated with events in the simulation:
	- p_inf: Probability of a vehicle being infected (0.2)
	- p_rep: Probability of a vehicle being repaired (0.2)
	- p_break: Probability of a vehicle breaking down (0.8)
seed	Random seed used for generating pseudo-random numbers (10)
vehicles	Initial conditions for the vehicles in the simulation:
	- id: Unique identifier for each vehicle
	- position: (x, y) coordinates of the vehicle within the network
	- state: Initial state of the vehicle (randomly assigned as "Not infected" or "Infected")



The evolution of vehicles' states

We run the simulation with 100 steps and for every step, we record the states of each vehicle in the network. Initially, there are 50 vehicles in the grid. The evolution of vehicle states (ranging from Infected, Not infected, Repaired, and Broken down) can be illustrated in the above Figure.

From the given Figure, we can see that the number of infected vehicles reduces significantly through time steps, while the number of broken-down vehicles increases dramatically. This can be understood that most infected vehicles now have broken down. As the system is configured with low repaired rate, the line of this rate shows no sign of increasing in the plot. The number of not infected vehicles was set high initially and almost stayed the same until the end of the period.

IV. CONCLUSION

conclusion, this project successfully simulated a cybersecurity attack on connected vehicles in a Vehicular Ad Hoc Network (VANET). Through the simulation, we gained insights into the dynamics of the attack and its impact on the network.

The results demonstrated the effectiveness of attack containment measures, as evidenced by the significant decrease in the number of infected vehicles over time. This highlights the importance of implementing robust security measures to mitigate the spread of such attacks in VANETs.

However, the simulation also revealed vulnerabilities in the network's resilience, as indicated by the notable increase in the number of broken-down vehicles. This emphasizes the need for further research and the development of efficient repair mechanisms to minimize vehicle downtime and ensure the network's functionality.

The project's graphical user interface (GUI) and the visual representation of vehicle states over time provided valuable insights and enhanced understanding of the attack's progression. Such visualization tools can aid in analyzing and responding to cybersecurity threats effectively.

This simulation project serves as a valuable learning experience for students interested in VANET cybersecurity. It underscores the importance of preventive measures, efficient repairs, and continuous research to enhance the security and resilience of connected vehicle networks.

Future research could explore different scenarios and parameters, such as varying percentages of initially infected vehicles, to gain a deeper understanding of attack dynamics and to evaluate the effectiveness of countermeasures. This would contribute to the development of comprehensive strategies to protect VANETs from cybersecurity threats.

REFERENCES

[1] Kim, K., Kim, J. S., Jeong, S., Park, J. H., & Kim, H. K. (2021). Cybersecurity for autonomous vehicles: Review of attacks and defense. Computers & Security, 103, 102150.
[2] Arif, Muhammad, et al. "A survey on security attacks in VANETs: Communication, applications and challenges." Vehicular Communications 19 (2019): 100179.