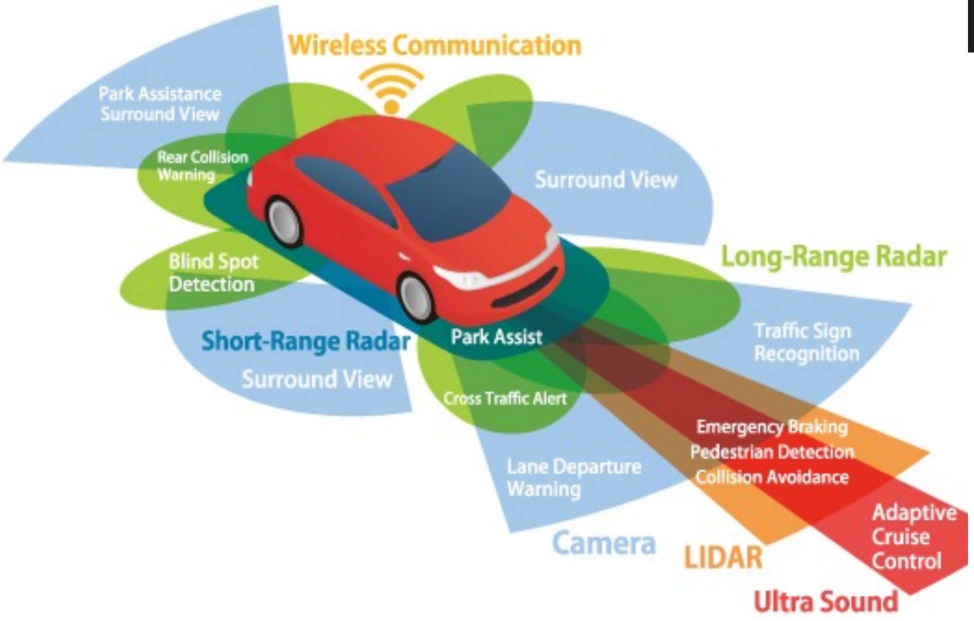
Discovering Adversarial Driving Maneuvers against Autonomous Vehicles

Ruoyu Song, M. Ozgur Ozmen, Hyungsub Kim, Raymond Muller, Z. Berkay Celik and Antonio Bianchi

Purdue University

USENIX Security 2023

Background




Bloomberg

Newsletter | Hyperdrive

This $220 Billion Market Opens Up a Path for Driverless Cars

Our Goal

- Identify adversarial maneuvers that cause an Autonomous Vehicle to deviate from its <u>mission</u> while maintaining attacker's <u>low liability</u>

Missions

- Identify adversarial maneuvers that cause an Autonomous Vehicle to deviate from its **mission** while maintaining attacker's low liability
- We extract 7 missions that **different levels** of AVs should comply from NHTSA (National Highway Traffic Safety Administration)'s documentation
- Two metrics and categories
  - Distance and Time to Collision (TTC)
- Formalize them to Linear Temporal Logic (LTL) formula
  - E.g., $\square Time\_To\_Collision(Victim_{Car}, FrontCar) > reaction\_time$

Low Liability

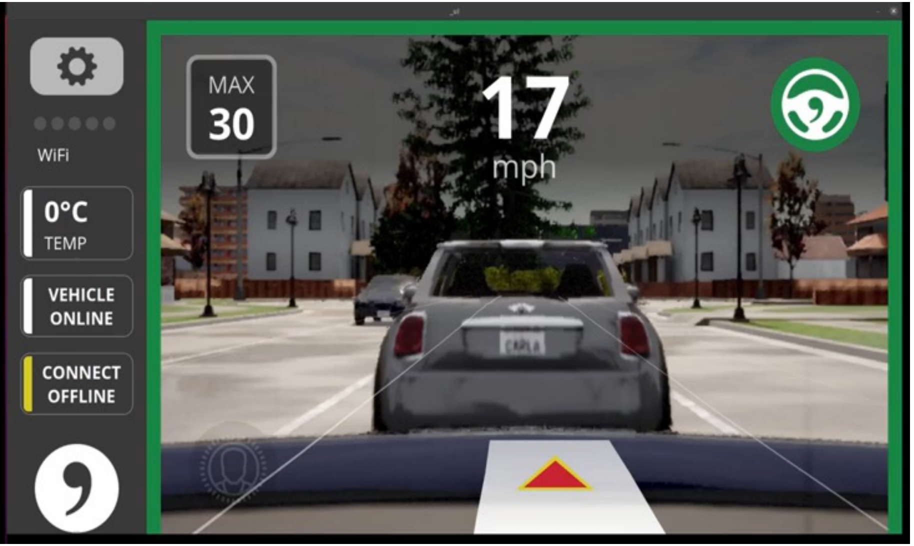
- Identify adversarial maneuvers that cause an Autonomous Vehicle to deviate from its mission while maintaining attacker's **low liability**
  - Does **not** crash with **any** object in the traffic
  - Does **not** violate traffic rules
- In total, we represent the ***low liability*** with 7 LTL formulas
  - The attacker should not make excessive maneuvers
  - E.g., $\square(throttle < \tau_a \wedge brake < \tau_b \wedge |steer| < \tau_c)$

Threat Model

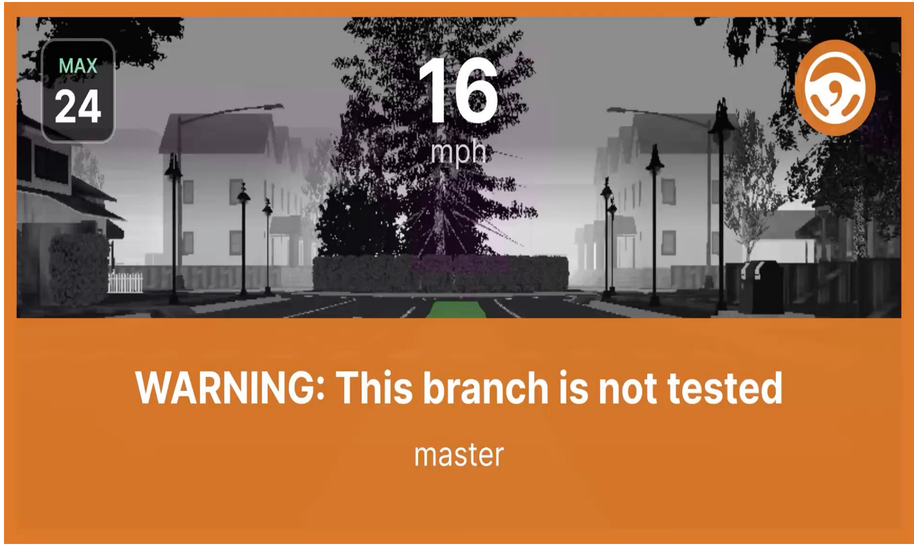
- Attackers drive their own car near a victim vehicle
- Attackers have the knowledge of victim car’s control software and physical state

Motivating Example

Without Low Liability



With Low Liability

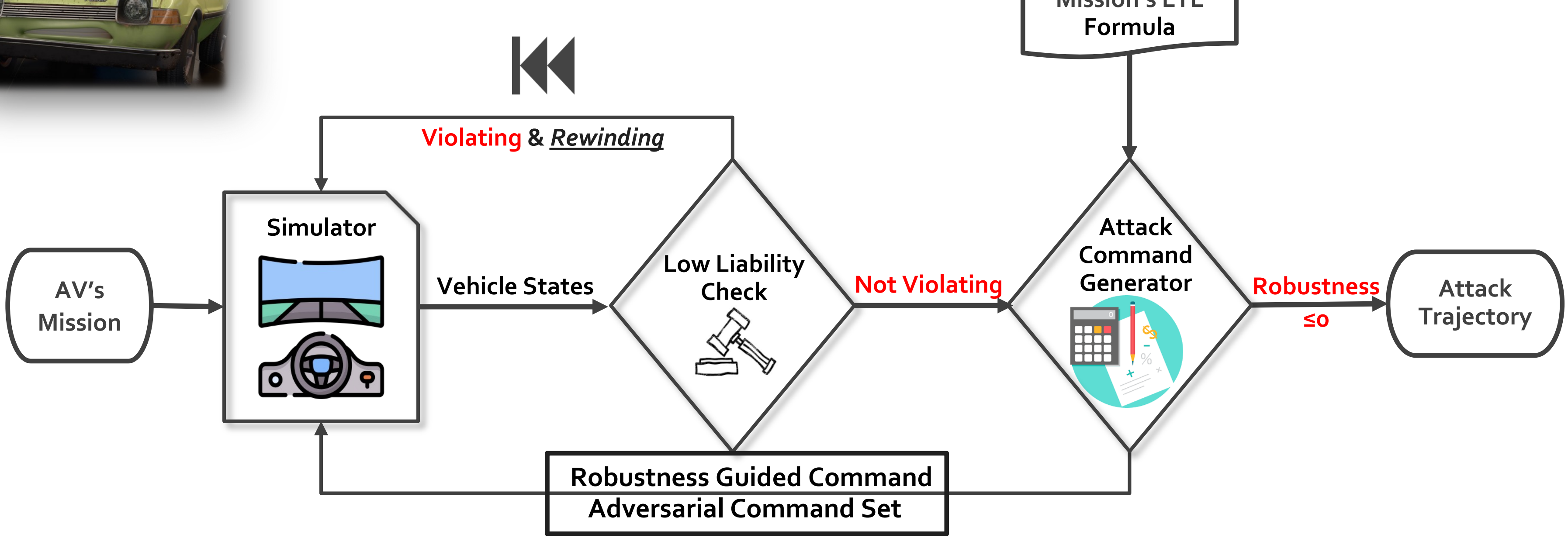


Car Mission: Lane-Centering

Approach

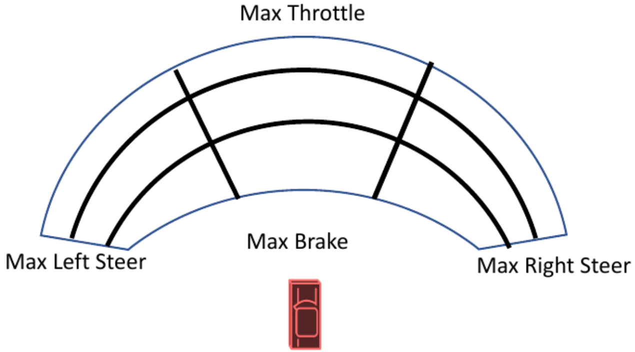
- We use a fuzzing approach to find adversarial maneuvers
- Our fuzzer is using Robustness as a guiding heuristics
  - Robustness defines how well the victim vehicle’s physical states (velocity and location) satisfy its safety missions
- When robustness is less than or equal to zero, the AV violates a mission

Acero Overview



Adversarial Command Generator

- Acero conducts a grid search at each round
  - In the initial round, Acero generates a set of adversarial commands without guidance



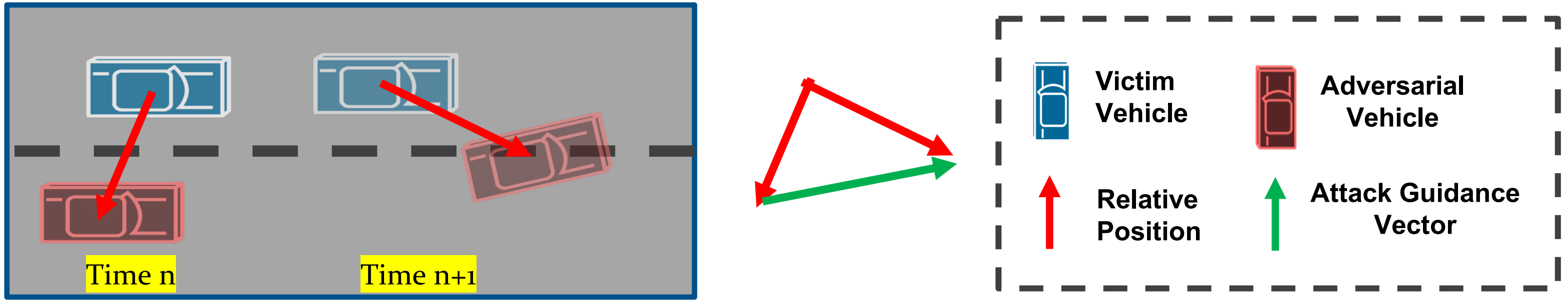
- Robustness Calculation
  - Robustness defines how well the victim vehicle's physical states (velocity and location) satisfy its safety missions

$$TTC\_Robustness = TTC(Victim, object) - reaction\_time$$

$$Dist\_Robustness = Dist(Victim, area)$$

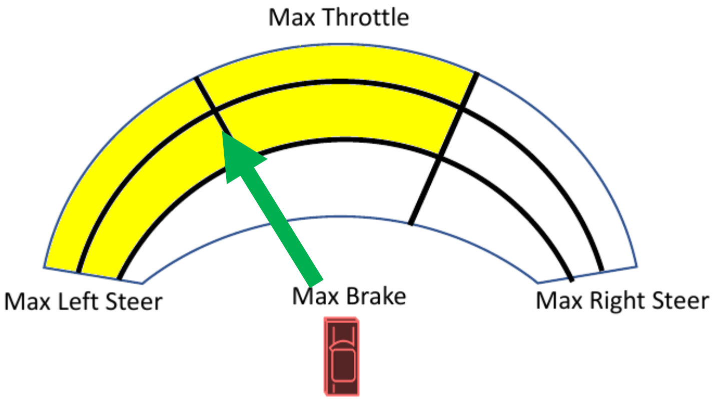
Attack Guidance Vector

- Attack Guidance Vector
  - We subtract the relative position between the victim car and attacker car at time n from their relative position at time $n+1$
  - $\boxed{relative_{position}(vv, av, n+1) - relative_{position}(vv, av, n)}$



Adversarial Command Generator

- After the initial round, Acero conducts a grid search guided by the attack guidance vector
  - Grid search **with guidance**
    - E.g., left steer and throttle grids
  - Terminates when the robustness reaches zero



Evaluation

- Evaluation Setup
  - Simulator: CARLA
  - AD Software:
    - openpilot
    - Autoware

THE AUTOWARE FOUNDATION

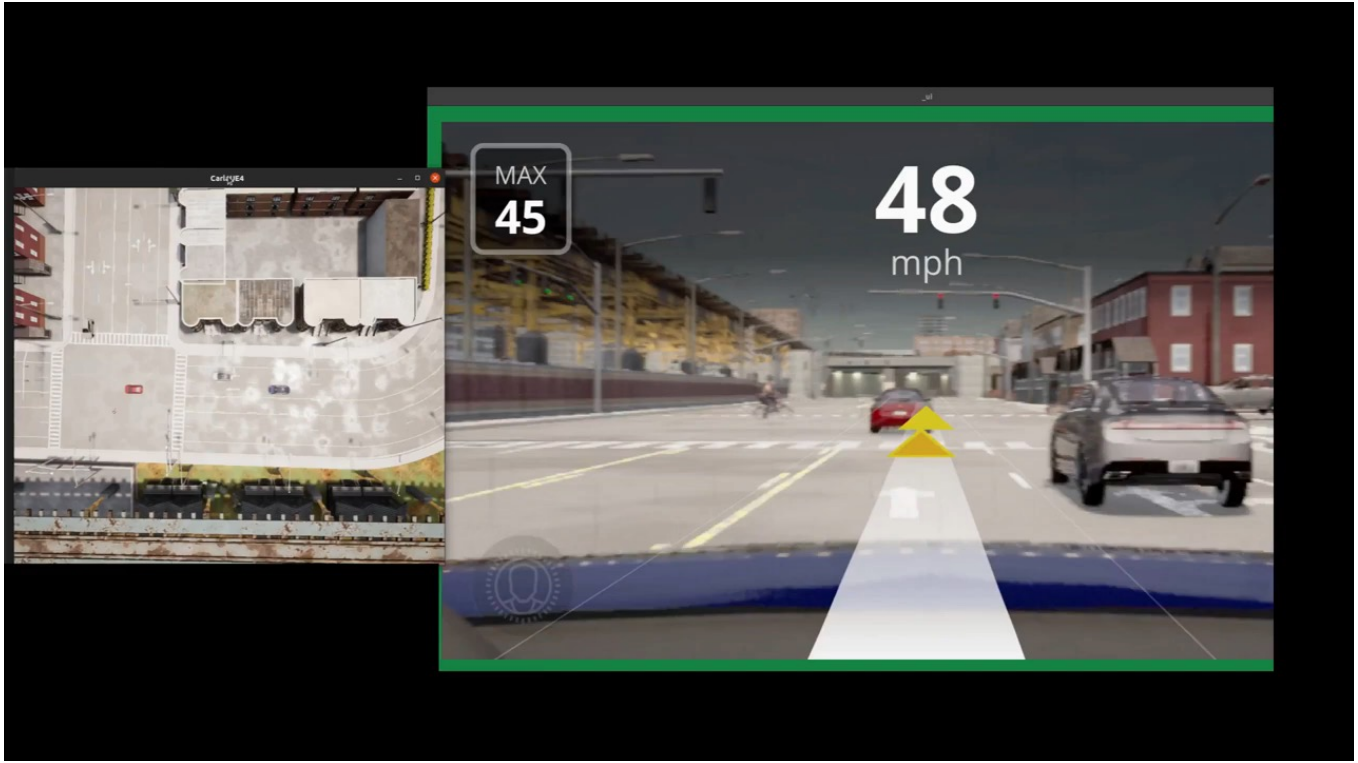
comma.ai

Evaluation

- **341** successful attacks out of **7000** attack attempts
  - **219** on openpilot and **122** on Autoware
  - **13** clusters on openpilot and **15** clusters on Autoware
  - Root Causes: vision blocking, configuration error, and planner error

Case Study 1-Autoware (Fails to React to a Stopped Vehicle)

Case Study 2-Openpilot (Fails to Follow the Front Vehicle)



Conclusions

- Acero is a trajectory generation system that generates low liability trajectories to cause the AV to fail its missions
  - Mission Identification and Formalization
  - Robustness-guided Adversarial Command Generation
  - Enforcing Physical Constraints on the Adversarial Vehicle
- We extensively evaluated Acero with two AV software (openpilot and Autoware) and identified hundreds of adversarial maneuvers that puts the victim vehicle and other agents in danger

Thank you! Questions?

song464@purdue.edu