Too Afraid to Drive: Systematic Discovery of Semantic DoS Vulnerability in Autonomous Driving Planning under Physical-World Attacks

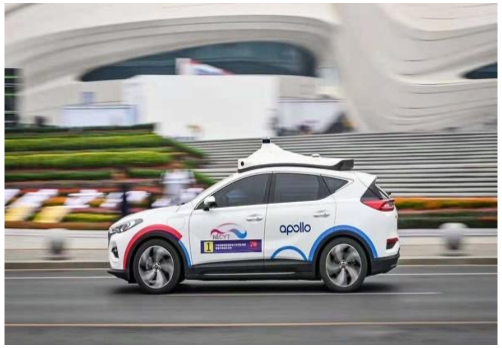
Ziwen Wan, Junjie Shen, Jalen Chuang, Xin Xia,
Joshua Garcia, Jiaqi Ma, and Qi Alfred Chen

Presenter: SangminWoo@Syssec

UCLA
AS2Guard Autonomous & Smart Systems Guard Research Group
UCI
SYSSEC

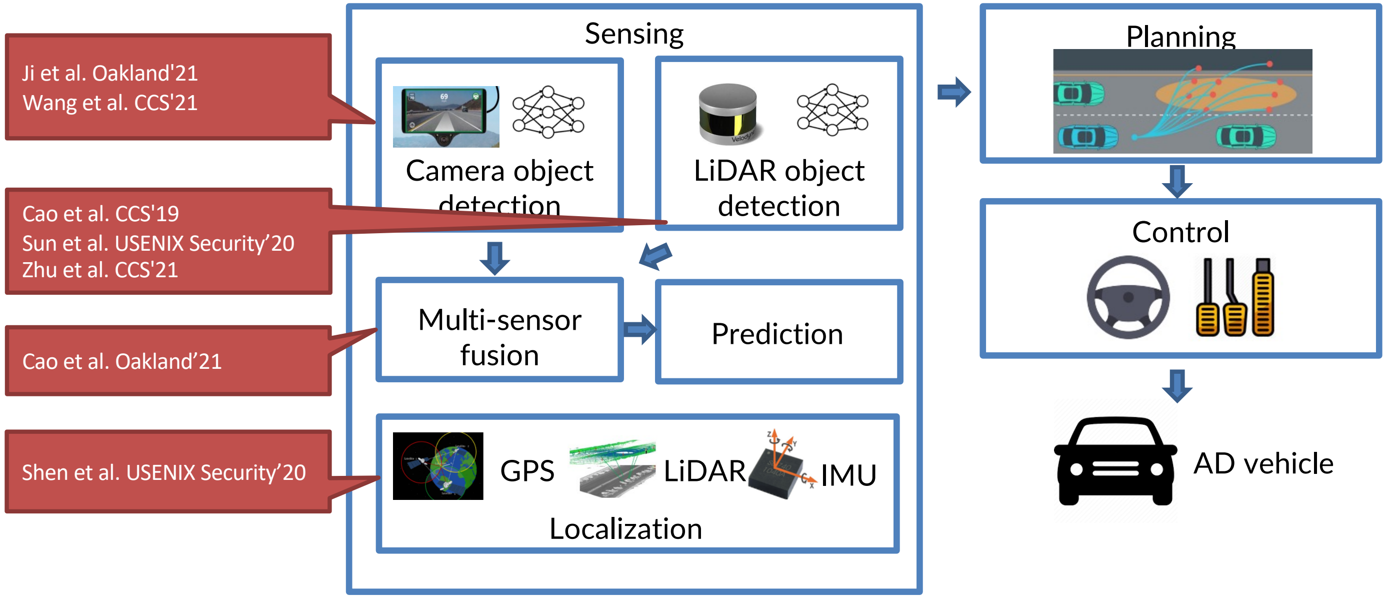
Introduction

❖ **High-level** autonomous driving vehicles are already providing services **without safety drivers**.



Introduction

❖ We have witnessed security problems in high-level AD systems.

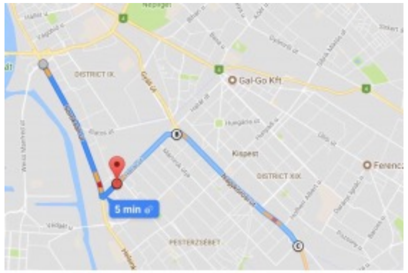


Question: Could planning (critical driving decision-making) also be vulnerable and thus exploitable to external attackers?

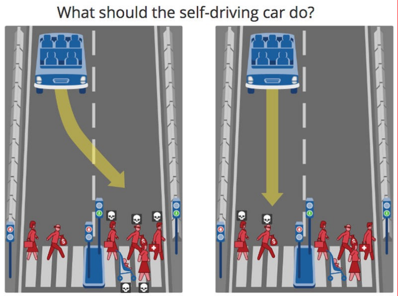
SYSSEC KAIST

Background

Planning

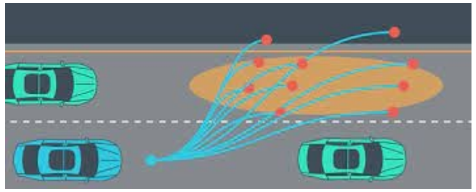


Route Planning



Behavioral Planning

Our focus



Local Planning

Example

As a human driver, how should you react to this scenario?

- ❖ Ignore them?
- ❖ Slow down?

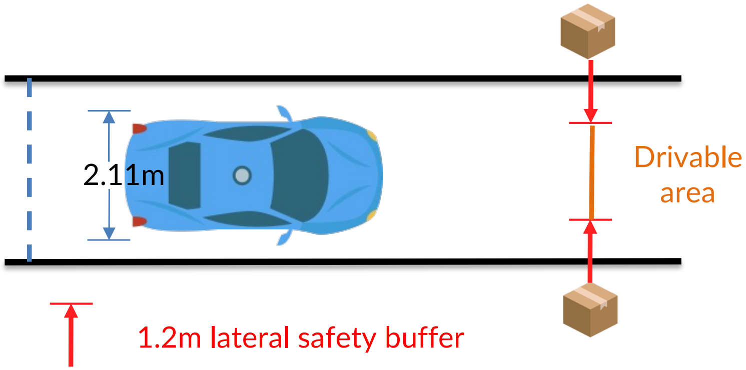
Example

Contribution

- ❖ Formulate the problem with a domain-specific vulnerability definition and a practical threat model
- ❖ Design PlanFuzz, a dynamic testing approach to systematically discover vulnerabilities
- ❖ Evaluate PlanFuzz on 3 different planning implementations
- ❖ Case studies

SYSSEC KAIST

DoS Vulnerability of Behavioral Planning



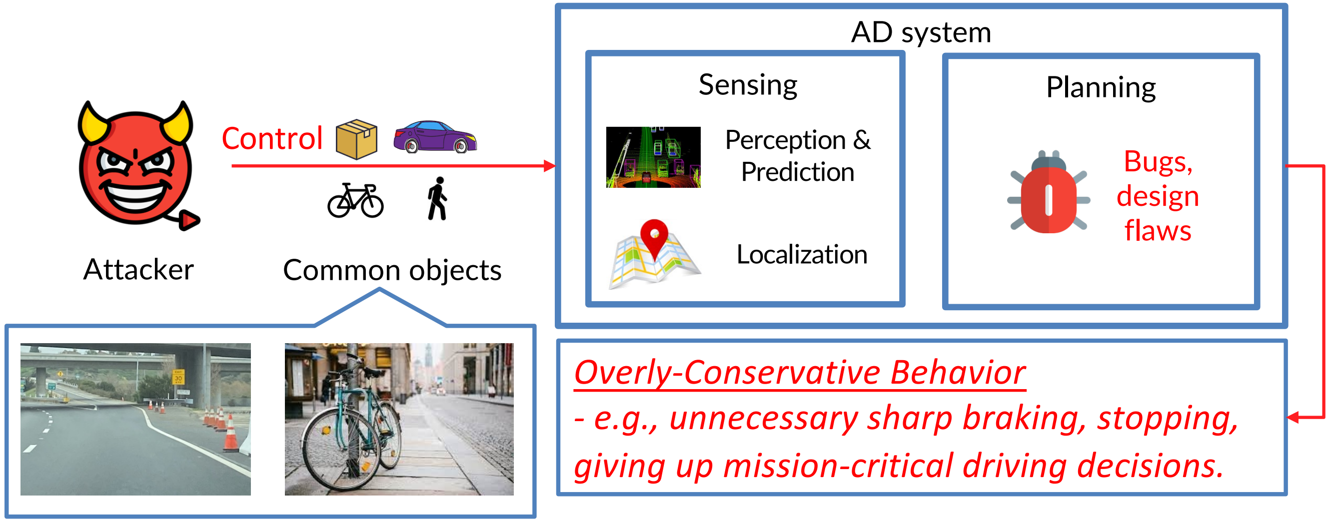
Drivable area (minimal value is (3.5 - 2*1.2)) < car width (2.11m)

<u>The AD vehicle thinks there is not enough space</u>

DoS Vulnerability of BP (Behavioral Planning):
Weakness in BP that disrupts decision-making, causing overly cautious actions and leading to mission failure or degraded performance.

Threat Model

- Attack vector: **attacker-controllable common** roadside objects
  - e.g., dumped cardboard boxes, parked bikes on the road side



Solution: Simulation-based Testing

❖ Real world testing is…

- Expensive
- Dangerous
- Time consuming

Simulation-based testing can address above issues!!

Question: How can we generate vulnerable scenario effectively?

Answer: Use guided fuzzing technique!

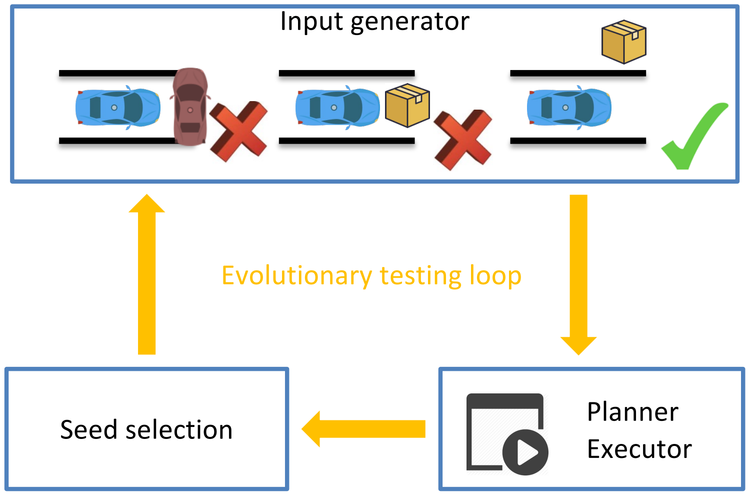
SYSSEC KAIST

Design Challenges

<u>Challenge 2:</u> How to generate inputs that satisfy domain constraints?

<u>Challenge 3:</u> How to design feedback to efficiently guide the testing ?

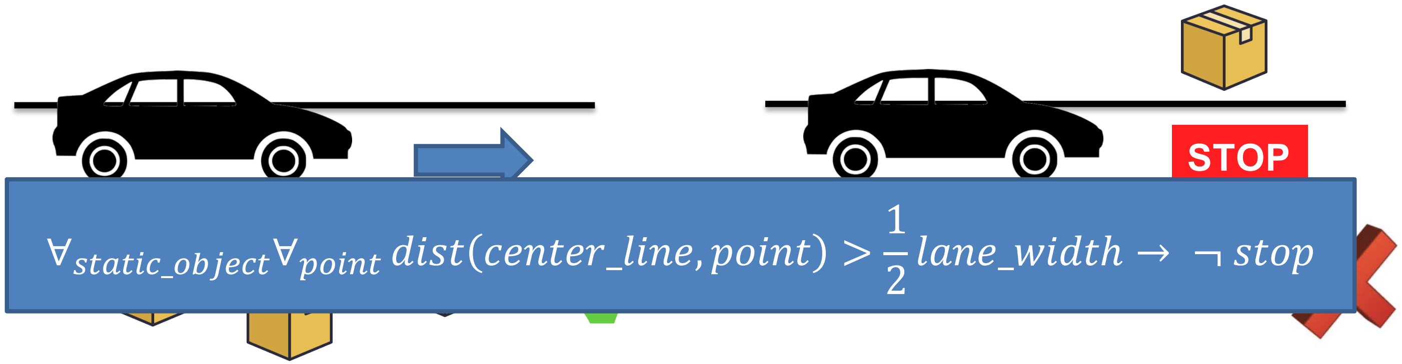
<u>Challenge 1:</u> How to judge a driving decision is *overly-conservative*?



SYSSEC KAIST

Solution: Planning Invariant (PI)

- To address challenge 1 (lack of testing oracles for semantic DoS vuln), we design planning invariant
  - Planning Invariants (PI) = **planning scenario** + **desired planning behavior** + **attacker-controllable changes**



$$\forall_{static\_object} \forall_{point}\, dist(center\_line, point) > \frac{1}{2} lane\_width \rightarrow \neg\, stop$$

SYSSEC KAIST

Solution: Planning Invariant (PI)

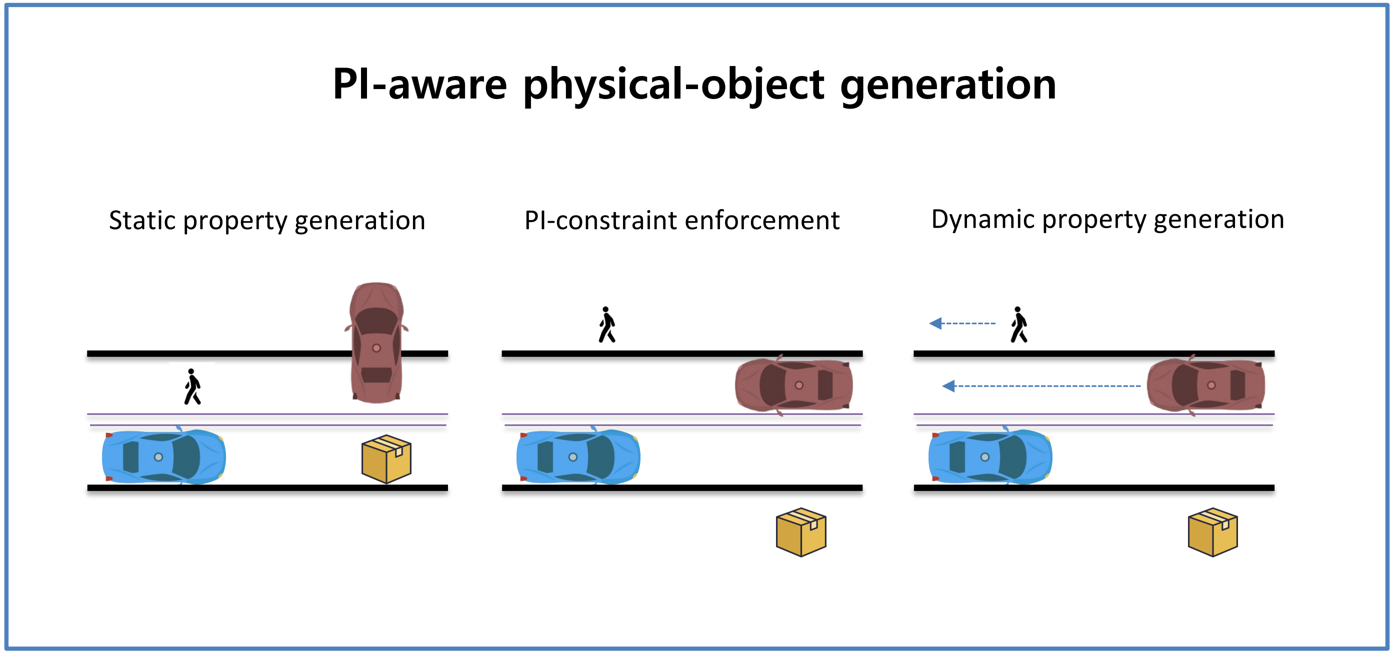
- **Systematically** define PIs under 8 diverse scenarios with **temporal logic** to constraint static objects, and **moving** pedestrian/vehicles

Table IV: Summary of Planning Invariants (PI) identified and used in the paper.

PI Index	Planning Scenario	Object Type	Constraints on Physical Objects	Desired Planning Behavior
PI1	Lane following (single-lane road)	Static obstacles	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on	Keep cruising in the current lane
		Vehicles	**PI-C2.** Follow the AD vehicle **PI-C3.** Drive on reverse lane	
		Pedestrians	**PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	
PI2	Lane following (multiple-lane road)	Static obstacles	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on	Keep cruising in the current lane
		Vehicles	**PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes	
		Pedestrians	**PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	
PI3	Lane changing	Static obstacles	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on	Finish changing to the targeted lane
		Vehicles	**PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes	
		Pedestrians	**PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	
PI4	Lane borrow (due to a blocking obstacle)	Static obstacles	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on **SP-PI-C1.** On-lane and in front of the blocking obstacle	Finish borrowing the reverse lane and pass blocking vehicle
		Vehicles	**PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes **SP-PI-C2.** On-lane and park in front of the blocking obstacle	
		Pedestrians	**PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	
PI5	Intersection w/ stop sign	Static obstacles	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on and the intersection the AD vehicle is going to pass	Pass intersection w/ stop sign following the traffic rule
		Vehicles	**PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes	
		Pedestrians	**PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	
PI6	Intersection w/ traffic signal	Static obstacles	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on and the intersection the AD vehicle is going to pass	Pass intersection w/ traffic signal following the traffic rule
		Vehicles	**PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes	
		Pedestrians	**PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	
PI7	Bare intersection	Static obstacles	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on and the intersection the AD vehicle is going to pass	Pass the bare intersection
		Vehicles	**PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes	
		Pedestrians	**PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	
PI8	Parking	Static obstacles	**SP-PI-C3.** Placed on other parking spots	Park into an empty targeted parking spot
		Vehicles	**SP-PI-C4.** Parked on other parking spots	
		Pedestrians	**SP-PI-C5.** Walking pedestrians moving away from AD vehicle	

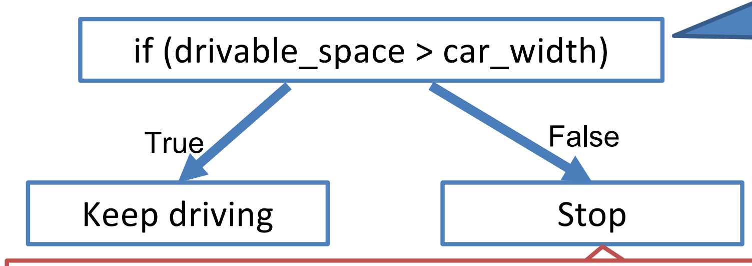
Solution: PI-Aware Object Generation

Input generation:
- Satisfy domain-specific constraints
- Maintain diversity and inheritance during mutation



Solution: BP Vulnerability Distance

- To address challenge 3 (lack of efficient guidance)
  - We propose **BP vulnerability distance,** which is a **gray-box** guidance.



Tiny fraction of Apollo lane changing control flow graph

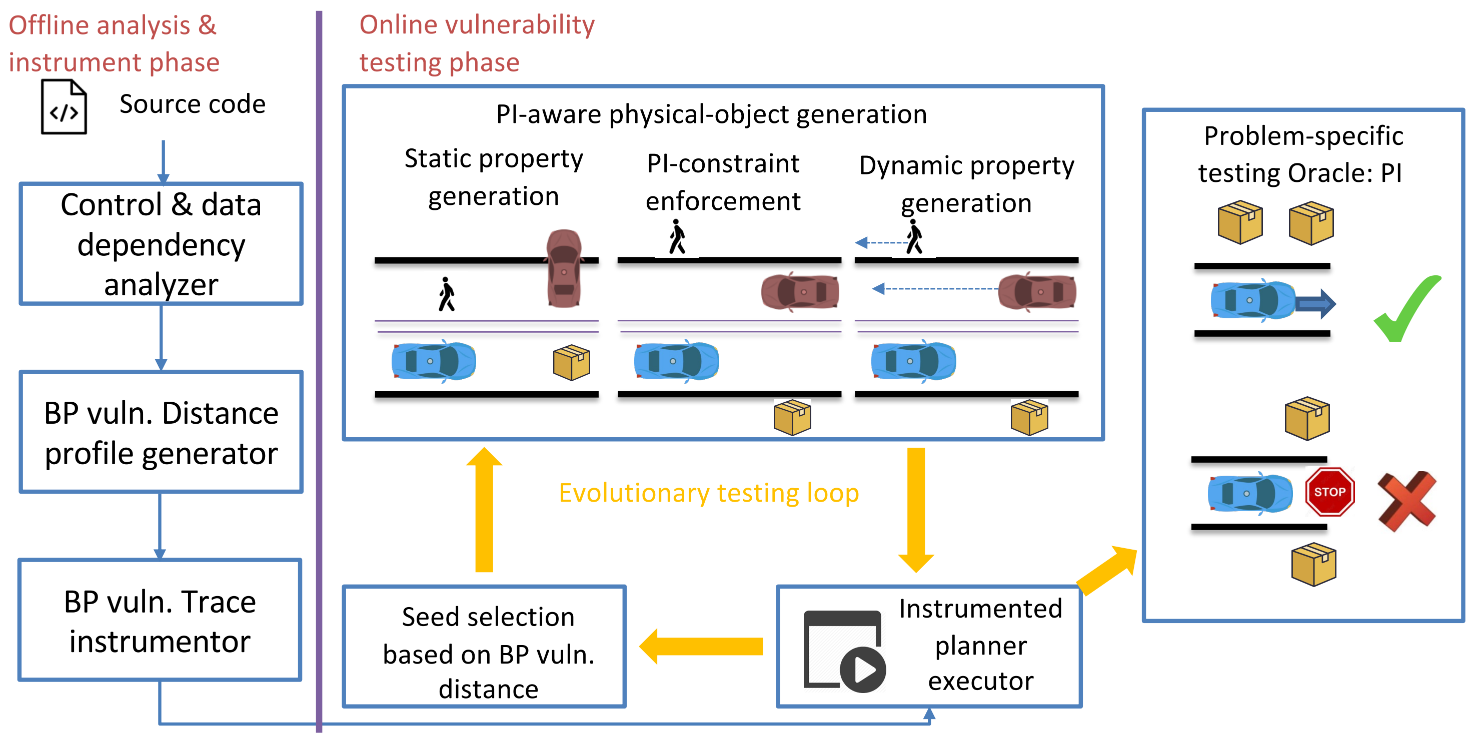
Key idea: Use the distance between operands in decision-related predicates to guide driving decision changes

Offline static analysis:
- Extract control/data dependency
- Generate BP vuln. distance profile for instrumentation

Online dynamic analysis:
- Calculate BP vuln. dist. at runtime

SYSSEC KAIST

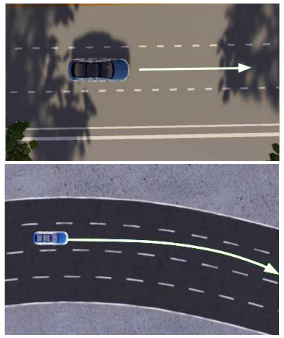
PlanFuzz

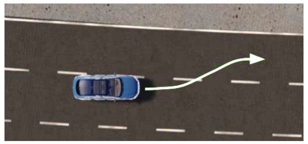

Offline analysis &
instrument phase
Source code
Control & data dependency analyzer
BP vuln. Distance profile generator
BP vuln. Trace instrumentor
Online vulnerability
testing phase
PI-aware physical-object generation
Static property generation
PI-constraint enforcement
Dynamic property generation
Evolutionary testing loop
Seed selection based on BP vuln. distance
Instrumented planner executor
Problem-specific testing Oracle: PI
STOP

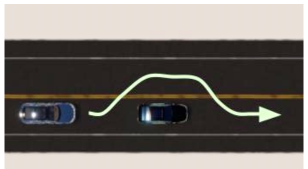
Evaluation

❖ **9 previously unknown** semantic DoS vulnerabilities from **3 BP implementations** of Baidu Apollo and Autoware.AI (full-stack open-source AD software)

- Causes: 1 due to implementation bug, 8 due to overly-conservative planning parameters (e.g., safety buffer, angle threshold) & overly-conservative estimation of surrounding object intentions (e.g., from pedestrians, parked bicycles)


Lane following


Lane changing


Lane borrowing


Intersection passing

Evaluation

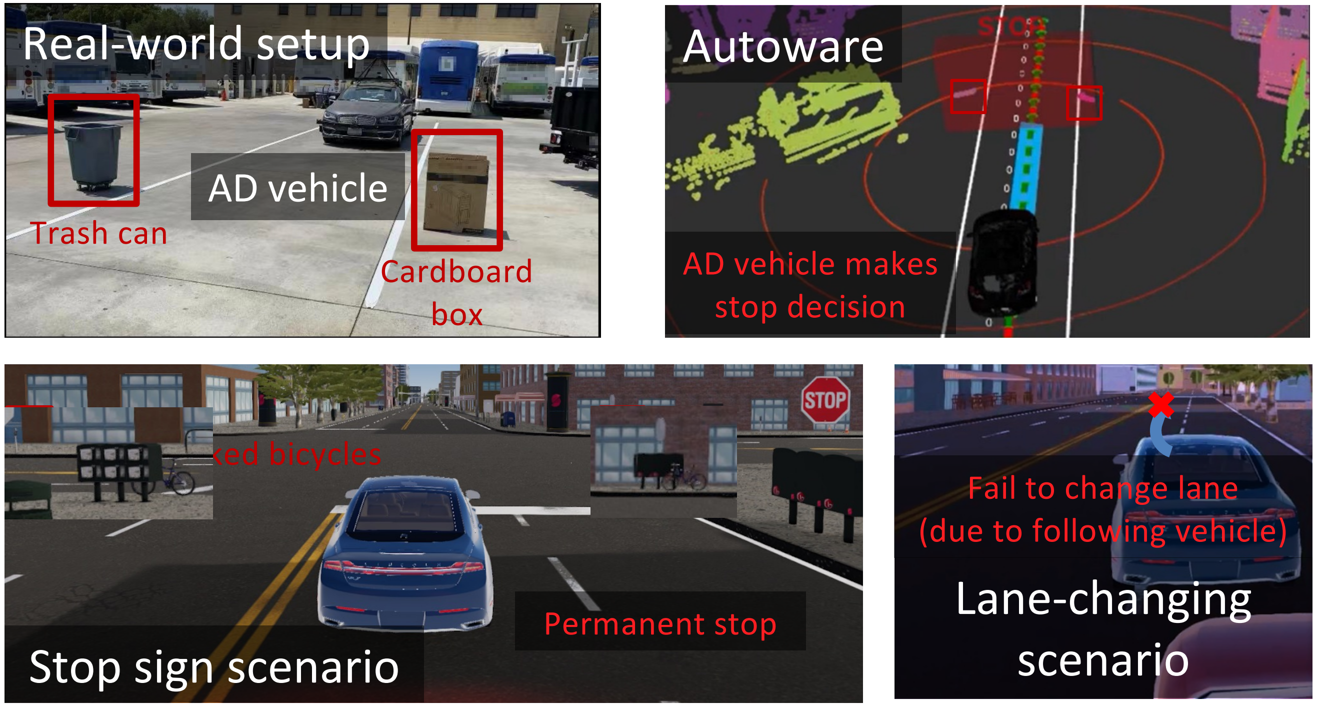
Scenario	Driving Behavior	Map	Vehicle	Duration (# of Planing Decisions)
Lane Follow (Single lane road)	Follow a 1-lane straight narrow road (2.7m lane width)	Single-lane road	Apollo: Lincoln	15.0s (133)
			Autoware: Lexus	25.4s (2394)
	Follow a 1-lane straight medium road (3.0m lane width)	Single-lane road	Apollo: Lincoln	14.3s (121)
			Autoware: Lexus	23.8s (2241)
	Follow a 1-lane straight wide road (3.5m lane width)	Single-lane road	Apollo: Lincoln	18.6s (157)
			Autoware: Lexus	24.6s (2037)
	Follow a 1-lane left-curved road	CubeTown	Apollo: Lincoln	21.3s (209)
			Autoware: Lexus	18.3s (1749)
	Follow a 1-lane right-curved road	CubeTown	Apollo: Lincoln	17.6s (172)
			Autoware: Lexus	21.3s (1978)
Lane Follow (Multiple lane road)	Follow a 2-lane straight road	San Francisco	Apollo: Lincoln	18.7s (177)
			Autoware: Lexus	15.4s (1379)
	Follow a 3-lane straight road	Modern City	Apollo: Lincoln	14.3s (121)
			Autoware: Lexus	21.3s (1840)
	Follow a 4-lane left-curved road	San Francisco	Apollo: Lincoln	18.7s (181)
			Autoware: Lexus	19.8s (1679)
	Follow a 4-lane right-curved road	San Francisco	Apollo: Lincoln	21.5s (208)
			Autoware: Lexus	25.9s (2379)
	Follow a 4-lane straight road	San Francisco	Apollo: Lincoln	13.4s (129)
			Autoware: Lexus	19.5s (1437)
Lane Change	Right change on a straight road	San Francisco	Apollo: Lincoln	21.2s (203)
	Left change on a straight road	San Francisco	Apollo: Lincoln	15.7s (138)
	Left change on a left-curved road	San Francisco	Apollo: Lincoln	13.4s (130)
	Right change on a left-curved road	San Francisco	Apollo: Lincoln	18.7s (172)
	Left change on a right-curved road	San Francisco	Apollo: Lincoln	16.4s (159)
Lane Borrow	Borrow lane on a straight narrow road (2.7m lane width)	Single-lane road	Apollo: Lincoln	25.9s (238)
	Borrow lane on a straight medium road (3.0m lane width)	Single-lane road	Apollo: Lincoln	28.7s (279)
	Borrow lane on a straight wide road (3.5m lane width)	Single-lane road	Apollo: Lincoln	30.5s (317)
	Borrow lane on a left-curved road	CubeTown	Apollo: Lincoln	27.3s (262)
	Borrow lane on a right-curved road	CubeTown	Apollo: Lincoln	33.2s (329)
Traffic Signal Intersection	Turn left at a 4-way intersection	San Francisco	Apollo: Lincoln	47.1s (453)
	Turn right at a 4-way intersection	San Francisco	Apollo: Lincoln	36.8s (329)
	Go straight at a 4-way intersection	San Francisco	Apollo: Lincoln	27.9s (288)
	Turn right at a 3-way intersection	San Francisco	Apollo: Lincoln	26.4s (233)
	Go straight at a 3-way intersection	San Francisco	Apollo: Lincoln	31.9s (308)
Stop sign Intersection	Turn left at a 4-way intersection	Shalun	Apollo: Lincoln	32.3s (334)
	Turn right at a 4-way intersection	Shalun	Apollo: Lincoln	27.9s (255)
	Go straight at a 4-way intersection	Shalun	Apollo: Lincoln	23.8s (220)
	Turn right at a 3-way intersection	Shalun	Apollo: Lincoln	33.2s (329)
	Go straight at a 3-way intersection	Shalun	Apollo: Lincoln	29.7s (283)
Bare Intersection Intersection	Turn left at a 4-way intersection	GoMentum Station	Apollo: Lincoln	37.9s (361)
	Turn right at a 4-way intersection	GoMentum Station	Apollo: Lincoln	42.3s (391)
	Go straight at a 4-way intersection	GoMentum Station	Apollo: Lincoln	30.1s (287)
	Turn right at a 3-way intersection	GoMentum Station	Apollo: Lincoln	29.2s (288)
	Go straight at a 3-way intersection	GoMentum Station	Apollo: Lincoln	38.5s (379)
Parking	Park to a front parking spot	GoMentum Station	Apollo: Lincoln	23.4s (228)
	Park to a left close parking spot	GoMentum Station	Apollo: Lincoln	30.5s (309)
	Park to a right close parking spot	GoMentum Station	Apollo: Lincoln	27.6s (263)
	Park to a left far parking spot	GoMentum Station	Apollo: Lincoln	24.3s (231)
	Park to a right far parking spot	GoMentum Station	Apollo: Lincoln	17.9s (163)

❖ **Diverse** driving scenarios

- **28,789** BP decision snapshots from **40** driving traces & **8** different scenario types

SYSSEC KAIST

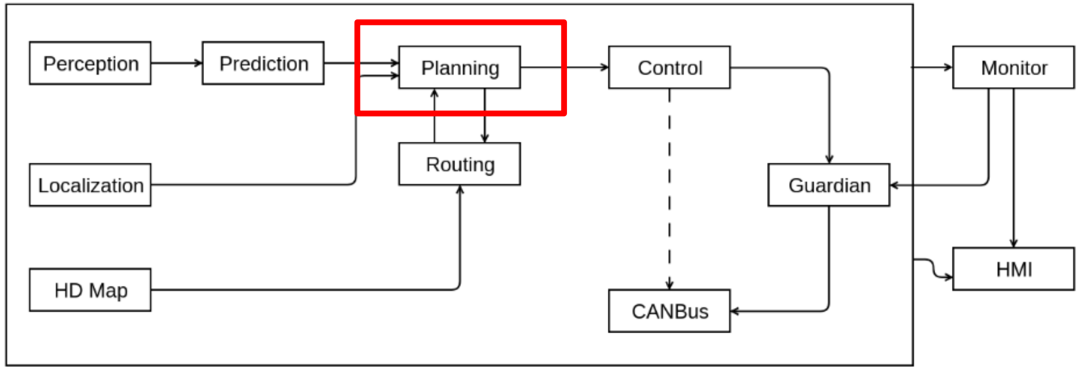
Case Study



SYSSEC KAIST

Limitations and Future Work

- **Testing Method: E2E vs Module Testing**
  - Result from module testing ≠ real-world vulnerability



- **Input Generation**
  - Driving scenarios with 40 driving traces
  - Uncovered scenario still exists.. (etc. Emergency scenarios in Baidu Apollo)

SYSSEC KAIST

Related Work – Testing Framework for ADS

ISSRE' 20
Av-fuzzer: Finding safety violations in autonomous driving systems

CVPR' 21
AdvSim: Generating Safety-Critical Scenarios for Self-Driving Vehicles

This paper

ASE' 22
LawBreaker: An approach for specifying traffic laws and fuzzing autonomous vehicles

CCS' 22
Drivefuzz: Discovering autonomous driving bugs through driving quality-guided fuzzing

SYSSEC KAIST

Conclusion

- **First** to perform AD planning-specific semantic vulnerability discovery with **a domain-specific vulnerability definition** and **a practical threat model**
- Design *PlanFuzz*, a **novel dynamic testing** approach that addresses various problem-specific design challenges
- Evaluate *PlanFuzz* on **two** practical open-source **full-stack** AD systems and discover **9** previously-unknown DoS vulnerabilities
- Perform exploitation case studies of **diverse driving scenarios** with simulation and driving traces collected from **a real AD vehicle**

Good Questions

- ❖ How can this approach to locating semantic DoS vulnerabilities be extended to aerial or marine autonomous systems or multi-agent AD?
- ❖ Wouldn't some of these attacks happen without anyone intending to (a real cardboard box on the side of the road), and in fact could happen rather frequently? Doesn't this paper hit the reputation of the AD systems by showing big flaws in their system?
- ❖ This paper highlights the challenge of overly conservative decisions in autonomous driving systems, leading to semantic DoS attacks. However, it doesn't fully explore how vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communication could be leveraged to mitigate these vulnerabilities. How could future research focus on using real-time communication networks between vehicles and traffic systems to provide additional context for decision-making, ensuring that an autonomous vehicle's behavior is aligned with its surroundings?
- ❖ Would the approach in this paper still be effective if the autonomous driving system were proprietary and the safety buffer algorithm were considerably more complex?

Best Questions

- ❖ **Donghyun Kim:** The paper focuses on how AD systems can be too careful. But is it possible that the opposite could happen? Could an attacker trick the car into thinking the road is clear, making the car drive too aggressively or even cause an accident? What protections are in place for this kind of problem?

- ❖ **Younghyo Kang:** Vulnerabilities can arise at various stages in the production and standardization of products due to reasons such as incorrect design, standard vulnerabilities, insufficient test case definitions, incorrect understanding, implementation vulnerabilities, and incorrect implementation. In the case of the vulnerability caused by overly conservative settings discussed in the paper, which stage would it belong to? I personally see it as an issue stemming from the absence of established standards (e.g., the range of safety margin settings). If this is the case, wouldn't it be more appropriate to attribute the problem not to a specific program but to the lack of established procedures in the process itself?

Best Questions

❖ **Sihun Yang:** What are the challenges in making PlanFuzz scalable to detect vulnerabilities across a variety of AD systems? How can PlanFuzz be extended or generalized to accommodate a variety of AD systems?