Effects of Redundant LiDAR Sensors on Object Hiding Attacks in Autonomous Driving Systems

Matthew Lee
Darwin Deason Institute for Cybersecurity
Dallas, TX, USA
leemh@smu.edu

William Flinchbaugh
Darwin Deason Institute for Cybersecurity
Dallas, TX, USA
wflinchbaugh@smu.edu

Eric C. Larson
Darwin Deason Institute for Cybersecurity
Dallas, TX, USA
eclarson@smu.edu

Mitchell A. Thornton
Darwin Deason Institute for Cybersecurity
Dallas, TX, USA
mitch@smu.edu

***Abstract*—Autonomous vehicles (AVs) use complex suites of sensors to understand the surrounding environment and inform decision making systems which ensure the efficiency and safety of their operation. Light Detection and Ranging (LiDAR) sensors are an important part of the perception subsystems of many AVs, and are responsible for identifying obstacles to prevent collisions. This critical function makes LiDAR sensors a prime target for malicious attacks such as object hiding attacks in which an attacker uses a laser to spoof a LiDAR point cloud to cause an object to be "hidden" from the AV. However, despite many AVs today having multiple LiDAR sensors with overlapping fields of view, LiDAR spoofing attacks described in existing literature test only on single LiDAR systems. We hypothesize that sensor redundancy can effectively "fill-in" spoofed regions if one of the sensors is attacked, providing some inherent resilience to object hiding attacks. In this work, we evaluate the effectiveness of two object hiding attacks, the Object Removal Attack (ORA) and the Physical Removal Attack (PRA), on an AV digital twin with one, two, and three LiDAR configurations. We report up to an 83% reduction in attack success rate when using multi-LiDAR configurations for both ORA and PRA when hiding vehicles and up to a 100% and 75% reduction for ORA and PRA, respectively, when hiding pedestrians.**

***Index Terms*—autonomous vehicles, LiDAR, point cloud, spoofing, security**

I. Introduction

Consumer autonomous vehicles (AVs) or "self-driving cars" represent one of the main research and investment areas for the automotive industry [1], [2]. Many competing companies such as Waymo, Wayve, Cruise, and more [3]–[5] are developing autonomous systems, each with their own approaches to the physical and software design of their vehicle(s). While implementations of an AV as a whole may vary widely, one of the fundamental systems present in all AVs is a perception system [6].

The perception system enables the AV to "see" the world around it by fusing sensor data, detecting vehicles, pedestrians, stop signs, and many other objects. With this information, other decision making systems can ensure that the vehicle operates safely and efficiently. AVs use several different types of sensors to feed information to the perception system. Cameras, LiDAR, and radar are the most commonly used sensors, with many AVs using combinations each of these to get a complete picture of the surrounding environment [7], [8]. It is important that these sensors are robust and secure as they are one of the primary tools for avoiding obstacles and preventing collisions [9]–[11].

Because these sensors are so important to the safe operation of AVs, they can be the targets of malicious attacks designed to cause harm to passengers or other individuals or property near the AV [12]. In this work, we focus on attacks which target an AV's LiDAR sensors. LiDAR (Light Detection and Ranging) is a method of generating a 3D point clouds of an environment through many range measurements using laser pulses and is a common feature in AV sensor suites [13], [14].

One method commonly used to attack AV LiDAR system is a spoofing attack. In a LiDAR spoofing attack, a malicious actor injects a modified point cloud into the AV system to cause undesired behavior. Object detection and tracking algorithms ingest point clouds from the perception systems and are responsible for identifying objects and obstacles. These can be fooled by the spoofed point clouds, resulting in dangerous driving behavior and injury. Remote spoofing methods are able to inject these modified point clouds without direct physical access to the target vehicle by detecting outgoing laser pulses and firing precisely timed and aimed laser pulses in return. Using such methods, attackers can "add" objects into a point cloud, causing an AV to see phantom objects [15]–[17]. Attackers can also "hide" real objects through methods such as the Object Removal Attack (ORA) [18] and the Physical Removal Attack (PRA) [19], manipulating the point cloud such that the AV will fail to detect the object. Object hiding attacks are particularly dangerous as an AV with no knowledge of an upcoming obstacle could collide with it and cause serious physical harm.

While ORA and PRA are thoroughly evaluated in their respective studies, they are only evaluated on single LiDAR systems. Many AVs use multiple LiDAR systems, often with some degree of overlapping fields of view [20]–[22]. While ORA and PRA target one LiDAR sensor to move points away from their original positions in order to cause misdetections, having multiple LiDAR sensors would allow the others to "fill in" the gaps introduced by the attacks. This would afford the AV some degree of inherent resilience to these hiding attacks through the use of redundancy. Therefore, it is important to understand how vulnerable redundant systems are to such attacks so that AV manufacturers can build in effective

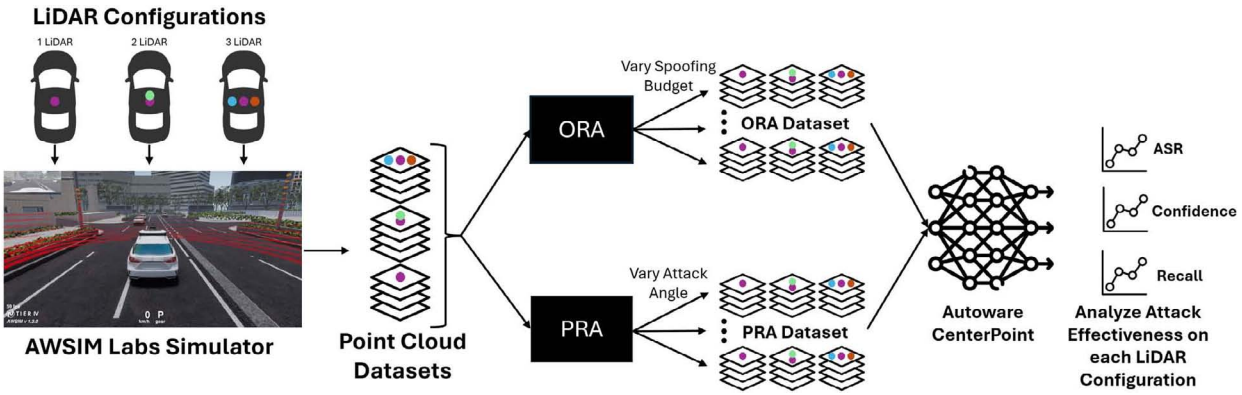

Fig. 1. Experiment overview; We capture point cloud datasets for each LiDAR configuration using Autoware/AWSIM Labs before applying ORA and PRA with varying degrees of intensity. The resulting attack-specific datasets are used as input for Autoware's CenterPoint object detector. The detection results are then analyzed to determine the effectiveness of the attack.

safeguards to prevent accidents.

In this work, we test this intuition through several experiments. We generate a dataset using digital twins with LiDAR configurations with one, two, and three LiDAR sensors and then implement ORA and PRA to fool the digital twins. We analyze the effectiveness of the attacks on these multi-LiDAR setups in a simulated environment using the Autoware autonomous driving stack and the AWSIM Labs simulator [23], [24]. When hiding vehicles, we report a $\sim$75% $\sim$83% reduction in attack success rate with two and three LiDAR sensors, even for ORA at maximum attack intensity (point budget of 400 points). For PRA, we also report a $\sim$65% and $\sim$83% reduction in attack success rate for two and three LiDAR configurations at maximum attack intensity (attack angle of 40°). When hiding pedestrians, we found that our baseline detection performance was much worse than with vehicles. We attribute this lack of performance to poor domain adaptation of our CenterPoint [25] object detection model to our simulated environment and 3D pedestrian model. Nevertheless, we report a $\sim$100% reduction in attack success rate of ORA and a $\sim$55% and $\sim$75% reduction in attack success rate of PRA for two and three LiDAR configurations, respectively. This indicates similar, although slightly diminished, resilience to attacks hiding pedestrians.

II. Background and Related Work

A. Light Detection and Ranging, LiDAR

Autonomous vehicles depend on a suite of sensors to ingest information about the surrounding environment which is used to make driving decisions. Combinations of cameras, LiDAR sensors, radars, and other sensors provide different modalities of information to an AV system [7], [8], [26]. LiDAR (Light Detection and Ranging) sensors are commonly used as a means to generate a 3D map of an environment. A LiDAR sensor emits (using a laser diode) a narrow laser beam in a direction and receives the beam's reflections (using a photodiode), recording the time delay between the initial emission and the resulting reflections. This delay is used to measure the distance from the sensor to an object in the direction of emission. Combining multiple measurements in different directions results in a point cloud, where each point represents a distance measurement. Many LiDAR sensors used in AVs are spinning LiDARs, consisting of stacks of emitters and receivers that cover different vertical angles, which spin as they take measurements. This creates rings of points which together form a 3D point cloud of the environment surrounding the sensor. These point clouds can be used as input to 3D object detection and tracking models which inform decision making within an AV [13], [14].

B. 3D Object detection

Once point clouds from the LiDAR sensors are collected and processed, they are passed to 3D object detectors which can operate on the point clouds alone or a fusion of mulitple sensor outputs [27]–[29]. In this work, we use the point cloud based detector CenterPoint [25] with a PointPillars [30] backbone. CenterPoint uses a two-stage process to detect 3D objects within the point cloud. In the first stage, the point cloud is fed through a backbone (PointPillars) to create a representation that is flattened and fed through a 2D convolutional nerual network (CNN). The CNN detection head estimates several object properties including the object's center location, orientation, velocity, and a 3D bounding box. The bounding box is used in the second stage to extract point features to refine the object's position and size, as well as a confidence score. While fooling 3D object detection models is ultimately the target of most perception attacks, the attacks we implement do not directly change or modify the model itself.

C. Perception Attacks

The importance of an AV's perception system makes it a target for malicious attacks, and, therefore, identifying possible avenues for these types of attacks is an evolving and active area of research. One method of attacking a perception system is through remote sensor spoofing. In a spoofing attack, an

attacker manipulates the environment detected by an AV's sensors in order to induce some desired behavior. We consider the attack to be remote if the attacker does not directly access the target vehicles system and instead manipulates external stimuli in order to execute the attack. Remote spoofing attacks have been demonstrated on cameras [31]–[33] and radar [34], [35], however in this work we focus on remote spoofing attacks on LiDAR sensors. Recent studies in LiDAR spoofing are based on an attacker sending fake echoes to a LiDAR sensor to cause the sensor to detect points at the attacker's desired location [15]–[19], [36], [37]. This approach is used for both object insertion and object hiding. With object insertion, the goal of the attack is to *insert* a fake object into an AV's perception system. In [17], Cao et al. use adversarial machine learning methods to generate point cloud patterns that will be recognized by the target vehicle as an obstacle. This method of object insertion is also demonstrated in [36], [37]. Conversely, an object hiding attack seeks to *hide* and object from an AV's perception system. Hau et al. [18] and Cao et al. [19] adapt the same point cloud spoofing methods used in insertion attacks to instead hide selected objects. They report high attack success rates when using a single LiDAR sensor. Hau et al. primarily present the impacts of ORA in terms of recall. Recall is the proportion of true positives that the model correctly detects vs. total number of positives in the dataset. We further discuss recall in Section V. Hau et al. report up to a $\sim$65% decrease in recall with PointRCNN [38] on vehicles and pedestrians when attacked and up to a $\sim$70% decrease with Point-GNN [39], which demonstrates how ORA can affect the performance of 3D object detectors and lead to misdetections. Cao et al. measure the effectiveness in terms of attack success rate (ASR), which is the proportion of objects which are detected by the object detector when not attacked and not detected when attacked. They achieve $>$95% ASR with PRA on both pedestrians and vehicles using Apollo 5.0 [40]. These results are calculated with an intersection over union (IoU) threshold of 0.7 for vehicles and 0.5 for pedestrians. These thresholds define how much overlap a predicted and ground truth bounding box must have to be considered a valid prediction. IoU in object detection is further discussed in Section V. Evaluating these attacks on multi-LiDAR systems is the main focus of this work.

D. Attack Countermeasures

As remote spoofing attacks have been developed, countermeasures to them have also appeared. Some such as [37], [41]–[44] analyze point clouds to identify spoofing attacks. These approaches search for temporal inconsistencies or physically impossible point placements which would indicate that the point cloud was spoofed. There is an inherent assumption with these methods that the AV has a single LiDAR sensor, or that the spoofed region is only within the field of view of a single LiDAR sensor. If multiple LiDAR sensors "cover" the spoofing region, then the resulting point clouds from each sensor can be compared directly to find inconsistencies. In [44] this is done with sequentially collected point clouds from a single sensor. Liu et al. [45] utilize a stereo camera system to perform cross-sensor validation, ensuring that objects detected by the LiDAR sensor or cameras are also detected by their counterpart. This approach leverages the presence of multiple types of sensors to detect inconsistencies. This multi-sensor approach and the single sensor assumptions of the previously mentioned countermeasures motivate and serve as the foundation for this work. Most AVs in development today have multiple LiDAR sensors with overlapping fields of view [20]–[22]. In autonomous driving stacks such as Autoware [23] and Baidu Apollo [40], point clouds from each LiDAR sensor are concatenated before being used for object detection. If the LiDARs' fields of view overlap, any regions where points are displaced due to object hiding attacks will be partially restored, allowing the objects to be detected.

E. Autonomous Driving Software

While many of the prominent AV developers create their own proprietary autonomous driving (AD) stacks, there are also open source solutions such as Autoware from the Autoware Foundation [23] and Baidu's Apollo [40]. These stacks can be downloaded and run by an individual and can be tested in 3D simulators such as AWSIM [24], CARLA [46], and LGSVL [47], among others [48], [49]. These simulators provide a virtual environment in which AD stacks can be tested. Most attacks and countermeasures listed in this section are evaluated in such simulators. We use Autoware and AWSIM Labs in our experiments.

III. Methodology

In this work, we implement two external LiDAR spoofing attacks, the Object Removal Attack (ORA) of Hau et al. [18] and the Physical Removal Attack (PRA) of Cao et al. [19] and evaluate their effectiveness against an AV with multiple LiDAR sensors with overlapping fields of view. These attacks use the same spoofing mechanism to carry out *remote object hiding attacks*. Object hiding attacks cause the AV to fail to detect real obstacles, unlike *insertion attacks* which cause the AV to detect objects that are not real (see Fig. 2). It is common for AVs to use multiple LiDAR sensors to provide the most complete coverage of the surrounding environment. In AD stacks such as Autoware [23], point clouds generated by these sensors are concatenated into a single point cloud which is then passed to the perception module to be used by the object detection/tracking modules. Our motivating intuition is that multiple LiDAR sensors with overlapping fields of view can mitigate the effects of these removal attacks, as they can at least partially restore or "fill in" the hidden or perturbed sections of the point cloud when the attacked and clean point clouds are concatenated. As a result, there is some inherent resilience to remote spoofing attacks, which would require a more complex attack to overcome. In the remainder of this section, we describe the primary mechanism that enables ORA and PRA, as well as the methodology behind each attack.

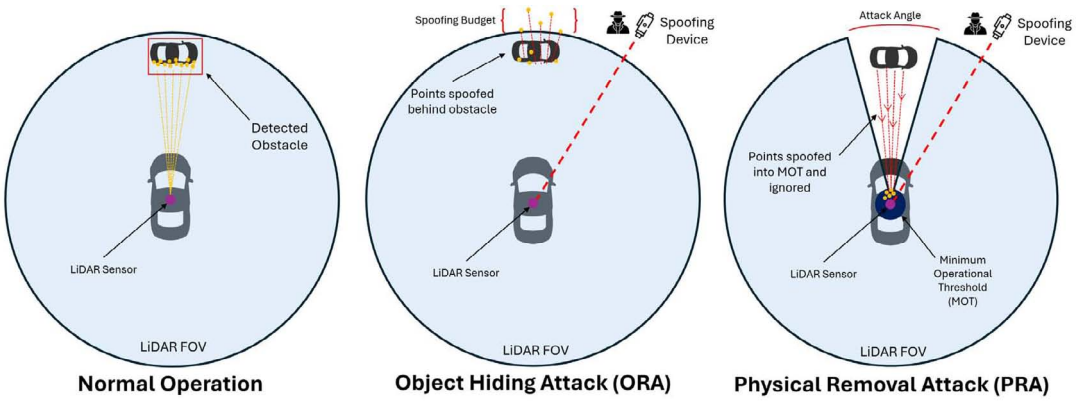

Fig. 2. Overview of attack methodology for ORA (center) and PRA (right). In our testing, we adjust the angle and the point budget to evaluate effectiveness across hardware capability of an attacker.

A. Remote Object Hiding Attacks

Both ORA and PRA are remote object hiding attacks, which we consider to be attacks that do not require physical access to any of the hardware components on the target system. Attacks such as [15]–[19] use a receiver similar to those on the LiDAR sensor to detect when the target vehicle's LiDAR sensor fires a pulse. Upon detection, the attacker fires a pulse of its own aimed at the target LiDAR after a calculated delay, which will induce the LiDAR to "detect" a point in the direction of the original pulse, but at a distance desired by the attacker. This requires precise timing and aiming of the attacker's laser pulse. Spinning LiDAR sensors often have a receiving window for a short duration after each pulse, during which received pulses are considered valid and recorded as points in the point cloud. Thus, the attacker must be able to time their laser pulse to fall within one of these windows. This is achievable through researching the specifications of the target hardware to find the receiving window and incorporating that window into the delay calculation once a pulse is received by the attacker's receiver. There is also an assumption that the attacker possesses an aiming system which includes an object tracker which can aim the attacker's laser pulses at the LiDAR sensor of a moving vehicle [50].

Previous studies utilizing this method have used different hardware setups and implementation techniques, resulting in varying constraints on the *maximum horizontal spoofing angle* and *spoofing budget* of the attacker. We define the maximum horizontal spoofing angle as the angle defining the sector of the spinning LiDAR's 360° horizontal field of view in which points can be spoofed by an attacker at a given location. We define the spoofing budget as the maximum number of points that an attacker can replace within one full rotation of the LiDAR sensor. With hardware and processing methods constantly improving, we relax these constraints in some instances as it is feasible that an attacker's capabilities will expand over time. Moreover, by assuming an attack with wider constraints, we create a challenging environment for countermeasures to maintain robust attack resilience.

Both attacks are evaluated in their original experiments using the KITTI dataset [51] which is a widely used dataset for AV applications. Since we are evaluating these attacks on AV systems with multiple LiDARs and KITTI was captured with a single LiDAR sensor, we create our own dataset with two different multi-LiDAR configurations. More details on the dataset are discussed in Section IV. We now discuss the methodology behind each of the two attacks and our implementations of these attacks.

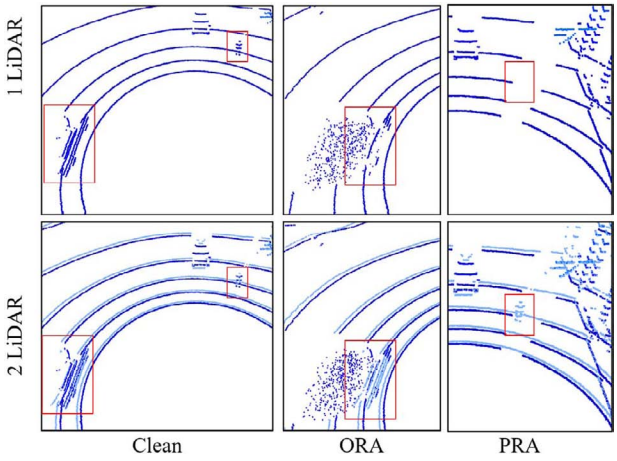

Fig. 3. Examples of the point perturbation of ORA (center) and the point removal of PRA (right) on a clean point cloud (left) in a 1 LiDAR (top) and 2 LiDAR (bottom) configuration

B. Object Removal Attack, ORA

The intuition behind the Object Removal Attack is that LiDAR points which would fall within an object's bounding box can be spoofed to appear to be outside of the bounding box, causing the distortion of the 3D features of the object and ultimately a misdetection by a 3D object detection algorithm.

Hau et al. [18] are the first, to the best of our knowledge, to use the spoofing methods described above for an object hiding attack rather than an insertion attack where the goal is to insert fake objects. In an imagined attack scenario, an attacker would set up a spoofing device (consisting of a receiver, delay component, and emitter) at a location with line of sight to the target LiDAR sensor. The attacker would then select a region or object to be hidden and determine its bounding box relative to the target sensor. The points within this bounding box would be candidates for spoofing. Some objects may contain more than the maximum spoofing budget within the region of their bounding box that falls within the maximum spoofing angle. This often occurs with large objects and objects close to the sensor. To address this limitation, Hau et al. use a random point selection strategy (*ORA-Random*), to choose which points to spoof. These selected points are then spoofed to appear a random distance behind the original point. This perturbation of the points within the bounding box causes a misdetection of the object. One key assumption of this method is that the spinning LiDAR is using *strongest return mode*. Light pulses emitted from the sensor may encounter objects that partially occlude or scatter the beam, resulting in multiple returns of different intensity which can allow the sensor to record multiple pulses from a single emission [52]. In strongest return mode, the sensor only records the strongest return pulse for each emission, resulting in a single point. This mode is commonly used in AV applications and is the assumed operation mode for ORA to be effective, so we also make that assumption [18]. As a result, the attacker in our assumed attack scenario can only "replace" a point along a legitimate laser pulse path. Figure 3 (center) demonstrates the perturbation effects of ORA. Displacing points from their original positions within an object's bounding box can disrupt the 3D features of the object enough to cause a misdetection.

In our implementation of this method, we follow the pseudocode provided in [18] with one modification. We relax the original work's maximum horizontal spoofing angle constraint of 10° and instead consider any points within the bounding box of the target object to be valid spoofing candidates. This results in a more complete perturbation of the point cloud than if only the points within the bounding box and within the 10° maximum horizontal spoofing angle could be spoofed. We also slightly relax the spoofing budget from 200 to up to 400 points. With more recently developed hardware setups such as the one in [19] of 45° and up to ∼3600 points, an attacker could plausibly move all the points in the bounding box even for a large vehicle. However, the original experiments demonstrate a decrease in performance upon perturbing only a subset of the points in the bounding box. Thus, in keeping with the motivation of the original experiments, we chose to double the original spoofing budget and remove the horizontal spoofing angle constraint to simulate an enhanced attacker capability without fundamentally changing the attack. The original experiments do not specify a range from which the random distances to spoof the point is selected. Therefore, we chose to randomly select a distance between one and three meters, as this keeps the displacement on the same scale as a vehicle with a footprint of about 4x2 meters and a pedestrian with a footprint of about 0.5x0.5 meters. Our modified ORA-Random implementation is described in Algorithm 1.

Algorithm 1 Modified ORA-Random
Note that ← denotes insertion to a list.

Input: raw_pc[]
$target\_bbox$ = (loc, dim, $orient$)
[$disp_{min}$,$disp_{max}$] // random displacement range
$budget$ // spoofing budget
Output: $spoofed\_pc$[]
$object\_pts$[] ← PointInBBox($raw\_pc, target\_bbox$)
$spoofed\_pc$[] ← [p for p ∈ raw_pc if p ∉ $object\_pts$]
$random\_pts$[] ← RandSample($object\_pts, budget$)
for all p in $object\_pts$ **do**
  if pt in $random\_pts$ **then**
    $random\_disp$ ← Random($disp_{min}, disp_{max}$)
    $spoofed\_pc$ ← SpoofPoint(p, $random\_disp$)
  else
    $spoofed\_pc$ ← p
  end if
end for
return $spoofed\_pc$

C. Physical Removal Attack, PRA

The Physical Removal Attack of Cao et al. [19] aims to completely remove a portion of a detected point cloud to induce misdetection. This is accomplished by spoofing points to be much closer to the target sensor, which causes them to be filtered or cropped from the sensor's output point cloud. The *spoofing region* defines the area into which the attacker must spoof points in order to remove them from the LiDAR sensor's output point cloud. By spoofing all points within a given attack angle θ into the spoofing region, an attacker can effectively remove an entire sector of the point cloud, preventing the AV from detecting any object within the spoofed sector. One of the mechanisms enabling this is the *Minimum Operational Threshold* (MOT). The MOT is the distance from the LiDAR sensor below which points are typically ignored or filtered. The MOT for a given sensor varies depending on the hardware and firmware of the device [52], [53]. A detected pulse that would place the point below this threshold is usually considered to be the result of an error or random event that is not relevant to the task. Thus, sensors either automatically filter these points out or have an option to do so in firmware.

An AV may also have a self cropping step in its processing pipeline which can have a similar filtering effect. Spinning LiDARs are often placed in locations on a vehicle where the vehicle itself occupies a portion of the sensor's field of view. This results in detected points which are reflections from the vehicle itself and not from any relevant external objects. To prevent the 3D object detection and tracking models from constantly detecting an obstacle on itself, AV pipelines such as Autoware will remove points which fall within its own

bounding box. Together, the regions defined by the MOT and the self bounding box form the LiDAR's spoofing region. In an attack scenario, we assume an attacker has knowledge of this spoofing region from the specifications of the target vehicle V and its LiDAR sensor model. With this knowledge, an attacker would choose an object to be hidden H. The attacker would then determine the spoofing direction ϕ which extends from the V to the center of H. Finally, the attacker would spoof all points emitted within a chosen attack angle θ around ϕ to be within this spoofing region, thus removing a portion of the target's point cloud. With no points to process, a 3D object detection and tracking system would fail to detect objects in the region which could lead to a collision. Figure 3 (right) gives an example of the point removal of PRA. Removing points within the region completely hides the pedestrian from the point cloud.

In our implementation of PRA, we used the same maximum horizontal spoofing angle constraint of 45° and point budget of ∼3600 points as in the original work. Since PRA involves spoofing all of the points within a given attack angle, the maximum horizontal spoofing angle is dependent on the spoofing budget. Spinning LiDARs in operation will have a vertical and horizontal resolution which defines how many points will be captured within a given horizontal angle. Thus, the maximum horizontal spoofing angle will be determined by the spoofing budget. Thus the constraint of 45° is based on the specifications Velodyne VLP-16 which captures ∼3600 points over 45° [52]. We did not explicitly constrain the spoofing budget, although the maximum horizontal angle we use is 40°. Our mean points removed for a 40° attack was ∼3100 points. We did not consider the specific MOT or self bounding box dimensions of our virtual LiDAR sensor, as our attack was implemented on captured data and we assume the attacker's ability to successfully spoof points into the spoofing region. Our implementation is further described in Algorithm 2

Algorithm 2 Physical Removal Attack, PRA
Note that ← denotes insertion to a list.

Input: $point\_cloud$[] // raw point cloud from sensor
Pos_V = $[x_v, y_v, z_v]$ // V
Pos_H = $[x_h, y_h, z_h]$ // H
θ // attack angle
Output: $spoofed\_point\_cloud$[]
ϕ_{spoof} = HeadingToPoint(Pos_V, Pos_H)
for all point in $point\_cloud$ **do**
  ϕ_{point} = HeadingToPoint(Pos_V, point)
  if $\phi_{point} > (\phi_{spoof} + \theta/2) \& \phi_{point} < (\phi_{spoof} - \theta/2)$ **then**
    $spoofed\_point\_cloud \leftarrow$ point
  end if
end for
return $spoofed\_point\_cloud$

IV. Experiments

To test the effects of these attacks on an AV with multiple LiDAR sensors, we create a custom LiDAR dataset using multiple LiDAR configurations in a simulated environment using the Autoware autonomous driving stack and the AWSIM Labs scene simulator [23], [24]. We then apply ORA and PRA attacks on the raw captured point clouds and replay the scenario using the "attacked" point clouds as input to simulate the attack. We analyze the object tracking results from the replayed scenario to evaluate the effectiveness of the attacks on the system. This pipeline is described in Figure 1. The following section will describe our experimental setup, procedure, and datasets.

A. Sensor Configurations

Our evaluation dataset consists of three LiDAR sensor placement configurations with one, two, and three LiDAR sensors. These configurations described below are chosen to provide multiple degrees of redundancy rather than to represent commonly used or optimal configurations in practice. In the single sensor configuration, the LiDAR sensor is placed in the center of the roof of the vehicle with a 360° FOV. The two and three sensor configurations allow us to investigate redundancy in multiple ways. The double sensor configuration has the same LiDAR sensor as in the single configuration, but we place a redundant LiDAR sensor stacked vertically, ∼70mm between the laser arrays of the two sensors. This results in an essentially identical horizontal FOV, and is the closest thing to a purely redundant sensor setup—however, this redundancy is less realistic because it does not mirror how current LiDAR system are configured. For the triple sensor setup, we again place one sensor on the middle of the roof, but we place the other two sensors 0.554m on either side of the center LiDAR, better mirroring current AV systems. Optimal LiDAR sensor placement is an ongoing field of research [54], [55] and placement configurations vary between manufactures [20]–[22]. Sensors may be set up such that there is one sensor on the roof with a roughly 360° FOV and individual sensors covering the front, sides, and rear views. This results in a similar level of overlapping coverage as our three sensor setup, with each directional LiDAR's FOV overlapping with the central roof mounted sensor. To verify that there is a negligible difference in FOV between our redundant LiDARs, we calculate the intersection over union (IoU) of the FOVs, considering each to be a circle with radius R (maximum range of the LiDAR sensor) at horizontal distance d apart.

$$A_{\text{intersection}} = 2R^2 \cos^{-1}\left(\frac{d}{2R}\right) - \frac{d}{2}\sqrt{4R^2 - d^2} \quad (1)$$

$$A_{\text{union}} = 2\pi R^2 - A_{\text{intersection}} \quad (2)$$

$$\text{IoU} = \frac{A_{\text{intersection}}}{A_{\text{union}}} \quad (3)$$

Since our sensors in our two sensor configuration are only displaced vertically, we consider their IoU to be 1. With a horizontal displacement d of 0.554m and R of 100m, we

get an IoU of 0.993, i.e. each of our redundant sensors in this configuration has over 99% overlap in FOV with the main sensor. This allows us to consider any object within the main sensor's FOV to also be in the additional sensors' FOVs, giving us complete redundant coverage of the scene. This provides us with the largest region within which we can test attack effectiveness even if the LiDAR configuration of a real AV may not have the same full coverage. The combined redundant regions of a particular LiDAR configuration from an existing AV for example could be represented as a subregion of the redundant coverage in our fully redundant configuration. Thus our experiments can generalize to configurations used in practice efficiently without needing to emulate so many unique setups.

B. Dataset

We generate our datasets in AWSIM Labs. AWSIM Labs is an open-sourced, Unity [56] based 3D simulator created for use with Autoware. For our LiDAR sensor, we use a virtual Velodyne VLP-16 spinning LiDAR sensor supported by AWSIM Labs which is built on RobotecAI's GPU LiDAR package [57]. We use AWSIM Labs' provided digital twin of the Shinjuku Tokyo area as our setting, which consists of a 3D model of the area and an accompanying lanelet2 map. Within this setting, we generated a dataset of 3,115 scenes containing varying numbers of pedestrians and vehicles per scenes. We used the same pedestrian and vehicle model for all scenes. We record the raw point clouds of each sensor within each scene as a to serve as the input data in our evaluations as well as the bounding boxes of the surrounding vehicles and pedestrians as ground truth. Autoware is built using Robot Operating System 2 (ROS 2) [**?**], allowing us to save the input and ground truth data in "rosbags." We process these rosbags using the Python utilities packaged with Autoware. The front-near region of the point cloud is often of particular importance, as it is the region of space into which the vehicle is most likely traveling. However, we place pedestrians and vehicles in all directions around the vehicle to generate the most instances of detection and tracking. For each of the three LiDAR placement configurations and for both ORA and PRA, we apply the chosen attack as described in Section III to all pedestrians and vehicles throughout the scene. We also vary the parameters of each method: point budgets of {0, 100, 200, 400} for ORA and attack angles of {0°, 5°, 10°, 20°, 40°} for PRA. In all scenes, we attack the same central LiDAR sensor as its placement is constant in all configurations. We do not consider attacks on multiple LiDAR sensors simultaneously. The spoofing device described in Section III could theoretically be deployed as an array of spoofing devices, each aimed at a different sensor. However a synchronized attack between multiple spoofing devices has not appeared in literature, to the best of our knowledge, and could be a target for future work. We revisit this topic briefly in the context of our experimental results in Section V.

C. Model and Hardware

We use Autoware's implementation of CenterPoint [25] with a PointPillars [30] backbone as our object detection/tracking model. The baseline model packaged with Autoware is trained on the nuScenes dataset [58] and the internal dataset of TIER IV, with about 39,000 total LiDAR frames. We did not perform any tuning on our datasets. We run all processing steps of the sensing and perception modules in Autoware leading up to the object detection/tracking results. We generate our dataset and run all simulations with an Intel Core i7-14700K CPU, RTX 4080 Super GPU, and 32GB of RAM. Each simulation run is about 90-100 seconds on average with the slight variance attributed to Autoware startup times.

V. Results and Discussion

We compare object detection results across the parameters of point budget for ORA and attack angle for PRA and across our three sensor placement configurations. The following section will describe our performance metrics and the results of these experiments.

A. Evaluation Metrics

We evaluate the effectiveness of ORA and PRA in the context of the performance of our CenterPoint object detection model using several metrics: confidence score, attack success rate (ASR), and recall. Confidence score is a direct output of CenterPoint which expresses on a scale of [0.0, 1.0], resembling a probability that the detection is "correct". For inference, we use the default confidence threshold of 0.35 to filter valid detections. However, we can also plot confidence over attack parameters, which can show how the model's certainty changes as the attack becomes more extreme. We consider the attack success rate to be the proportion of detections that were missed in the attack dataset versus the clean dataset:

$$\text{Attack Success Rate} = \frac{\text{TP}_{baseline} - \text{TP}_{attack}}{\text{TP}_{baseline}}$$

where $\text{TP}_{baseline}$ is the count of true positives on the clean (no attacks) dataset and TP_{attack} is the count of true positives on a dataset with attacks. This ratio communicates the effectiveness of the attack overall and is less coupled to a specific object detection model. For ORA, we also measure performance on recall to match the evaluations from the original ORA experiments [18].

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Recall is a relevant metrics for ORA since we want to test the model's ability to detect all the objects, including those the attacker is trying to hide. In object detection evaluations, it is common to use an intersection over union (IoU) threshold in addition to a confidence threshold to assess whether the predicted bounding box overlaps sufficiently with the ground truth bounding box to be considered a valid detection. Thus, a low IoU threshold would mean that even if the bounding box is poorly localized, a predicted detection would still

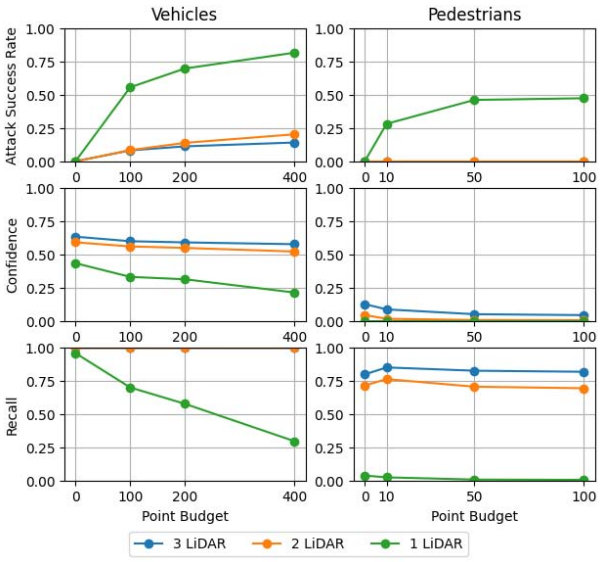

Fig. 4. Attack Success Rate, Confidence, and Recall over point budget for one, two, and three LiDAR configurations, with a point budget of zero representing no attack.

be considered correct. We use IoU thresholds of 0.7 and 0.2 for vehicles and pedestrians, respectively, in our metrics. Thresholds of 0.7 and 0.5 are used in the original ORA and PRA works, however we relax the IoU to 0.2 for pedestrians to account for poor domain adaptation of our CenterPoint model.

B. Object Removal Attack Results

In the first row of Figure 4, we see ASR of ORA on vehicles and pedestrians, evaluated with IoU thresholds of 0.7 and 0.2, respectively. We can see a clear difference in the effectiveness of ORA on the single vs multi-sensor configurations. While the ASR does increase slightly with point budget for the two and three LiDAR configurations, the single LiDAR configuration is notably more susceptible. There is a similar trend with pedestrians, with no successful attacks on the multi-LiDAR configurations. The two redundant configurations maintain a more consistent ASR, with no successful attacks with any point budget for pedestrians and a $\sim$75% and $\sim$83% decrease in ASR with the two and three LiDAR configurations.

The second row of Figure 4 shows the object detection confidence over point budget for each LiDAR configuration. For vehicles, we see a steady decrease in confidence in the single LiDAR configuration as the point budget increases. The confidences for the two and three LiDAR configurations stay relatively steady, although there is a slight decrease as well. Our model is not confident on pedestrians in the single LiDAR configuration even with no attack present (point budget of zero), making it difficult to identify a trend from the confidence plot.

The bottom row of Figure 4 shows the recall curves for vehicles and pedestrians over point budget for each LiDAR configuration. There is a clear negative trend in recall as point budget increases for vehicles, reducing the recall to $\sim$30% at a point budget of 400 points. As more points get perturbed, the 3D features of the vehicles are more disrupted resulting in fewer detections. However, recall for the two and three LiDAR configurations remains at >99% for all attacks. This is a $\sim$330% increase in recall at a point budget of 400 points—redundant configurations effectively prevent vehicle removal attacks. For pedestrians, object detection performs poorly in the single LiDAR configuration, even with no attacks. Since we cannot see a trend in recall in the single, LiDAR configuration, we cannot conclude that ORA is less effective against multi-LiDAR configurations despite seeing much higher recall in the two and three LiDAR configurations.

Discussion: In both recall and model confidence, the perception system appears to be more resilient in the two and three LiDAR configurations than in the single configuration for vehicles. As points are spoofed away from the original bounding box, the 3D features of the vehicle are somewhat preserved by the additional points from the redundant LiDAR sensors. While we would expect higher performance with redundant points even without an attack, we would also expect similarly decreasing performance trends to the single LiDAR configuration for the two and three LiDAR configurations unless the redundant points were adding resilience to the system. Based on the consistent performance of the two and three LiDAR configurations as the point budget increases across all metrics, we conclude that redundant LiDARs improve the resilience of the system to ORA attacks for vehicles.

For pedestrians, the poor confidence and recall performance in the single LiDAR configuration makes it difficult to make a strong conclusion about the effects of the redundant LiDAR sensors, despite seeing a large reduction in ASR when using redundant LiDAR sensors. We note that original CenerPoint detection model [25] reports pedestrian tracking accuracies of about 58% on the Waymo Open Dataset [59]. In our implementation, we see reduced performance for pedestrians detection, with recall scores less than 5%. We attribute this performance drop primarily to poor domain adaptation. As such, fine tuning of the CenterPoint model may improve recall on pedestrians in our simulations. Even so, we can still investigate the trends in attack success rates for the CenterPoint model.

C. Physical Removal Attack Results

We evaluate PRA based on the *attack success rate* (ASR) and the confidence of the CenterPoint model with IoU thresholds of 0.7 and 0.2 for vehicles and pedestrians, respectively. In some attack samples, only a portion of the object is removed—e.g., when a vehicle close to the target is spoofed with a small attack angle. While the goal of PRA is to completely remove the object, an evaluation of confidence will identify trends in confidence for vehicles and pedestrians which were only partially removed.

The first row of Figure 5 shows the trends of ASR as the attack angle increases. We see in the one LiDAR configuration how the effectiveness of PRA increases as the attack angle

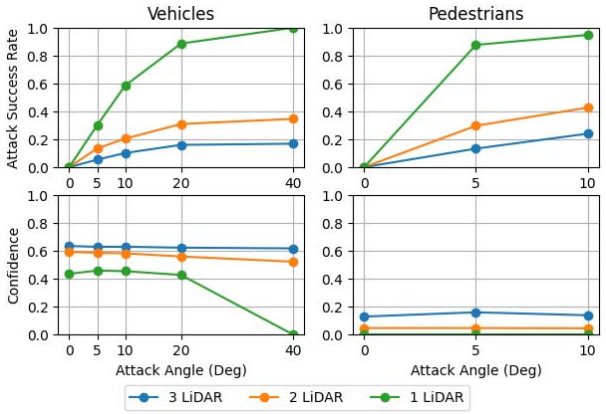

Fig. 5. Attack Success Rate, Confidence, and Recall over point budget for one, two, and three LiDAR configurations, with an attack angle of zero representing no attack.

increases and thus a greater proportion of each objects bounding box is hidden by the attack. The ASR for the two and three LiDAR configurations remains more consistent. At an attack angle of 40°, we see an ∼65% and ∼83%, respectively, decrease in ASR compared to the single LiDAR configuration. This means that the two and three LiDAR configurations maintain some degree of detection in ∼65% and ∼83% of the attacks at the largest attack angle. We see a similar ASR result for pedestrians, reaching a maximum ∼55% and ∼75% reduction in ASR. As discussed in our ORA results, the CenterPoint model performs poorly on pedestrians in our experiments, resulting in a similar but less pronounced contrast between the one LiDAR configuration and the two and three LiDAR configurations.

The second row of Figure 5 shows the confidence of the object detection model (CenterPoint) as the attack angle increases. For vehicles, we see that for small attack angles, only portions of the vehicle's point cloud are removed, and thus the model maintains confidence even in the single LiDAR configuration. However, this confidence decreases to 0 at an attack angle of 40° when the majority of a vehicle's point cloud is removed but a small portion remains. As with all previous pedestrian results, the poor performance in the single LiDAR configuration even with no attacks prevents us from identifying confidence trends across attack angle.

Discussion: The trends in ASR with increasing attack angle demonstrate the effectiveness of PRA at hiding objects. Especially with vehicles, the redundant LiDAR sensors afford a great deal of resilience to these attacks, as they compensate for the regions hidden from the main LiDAR. While it is not the main goal of PRA, the confidence of the model is still degraded significantly in the single LiDAR configuration even when the vehicle is well localized. The redundant LiDARs restore the missing portions of the vehicle, maintaining the model's confidence. The pedestrian results show a reduced level of resilience in our ASR evaluations and inconclusive results in our confidence evaluations. That is, the overall trend for resilience is similar to vehicles, but the detection performance for pedestrians is not sufficient to make strong conclusions. We also note that the reduction in ASR from the single LiDAR to two LiDAR configurations shows that as long as a single sensor is operating normally, the attacks are significantly less effective. This indicates that in the event of a synchronized, multi-spoofer attack on multiple LiDAR sensors as referenced in Section IV, the attacker would need to deploy spoofers for every sensor whose FOV contains the target object in order for the attack to be effective.

VI. Conclusion

In this work, we test the resilience of a multi-LiDAR system to two "remote object hiding" spoofing attacks: the Object Removal Attack (ORA) and the Physical Removal Attack (PRA). We employ two redundant LiDAR systems using digital twins, where each configuration has an overlapping field of view with the original LiDAR sensor. We find that a redundant LiDAR sensor is able to supplement any perturbed or missing points caused by these attacks, providing the target system inherent resilience. We find strong resilience when vehicles are hidden, reporting a ∼75% and ∼83% reduction in ASR with two and three LiDAR sensors, even at maximum attack intensity (point budget of 400 points). Furthermore, we find an ∼65% and ∼83% reduction in attack success rate for two and three LiDAR configurations for PRA at maximum attack intensity (attack angle of 40°). We also find that redundant LiDAR sensors provide slightly less consistent resilience when pedestrians are hidden—likely stemming from the poor detection performance of LiDAR in detecting our 3D pedestrian model, particularly from longer distances. Fine tuning the detection model, CenterPoint, could lead to improved pedestrian performance in future work.

Many AV's in production or development today have multiple LiDAR sensors [20]–[22] which, based on this work, may afford them this same resilience to remote object hiding attacks. In our literature review, we find few works investigating attacks which can affect redundant sensors simultaneously. Extending existing attack methods to account for sensor redundancy is an avenue of future work. If successful, another area of future work will be in enhancing existing spoofing countermeasures to account for these redundant sensors attacks.

References

[1] R. Baldwin, "Self-driving-car research has cost $16 billion. what do we have to show for it?" 2020, accessed: 2025-02-12. [Online]. Available: https://www.caranddriver.com/news/a30857661/autonomous-car-self-driving-research-expensive/

[2] C. Metinko, "Funding to autonomous driving startups surprisingly starts to move again," 2020, accessed: 2025-02-12. [Online]. Available: https://news.crunchbase.com/transportation/autonomous-driving-startup-funding-wayve-cruise/

[3] Waymo, "Waymo official website," accessed: February 12, 2025. [Online]. Available: https://waymo.com/

[4] Wayve, "Wayve official website," accessed: February 12, 2025. [Online]. Available: https://wayve.ai/

[5] C. LLC, "Cruise official website," accessed: January 27, 2025. [Online]. Available: https://www.getcruise.com

[6] A. Yoganandhan, S. Subhash, J. Hebinson Jothi, and V. Mohanavel, "Fundamentals and development of self-driving cars," *Materials Today: Proceedings*, vol. 33, pp. 3303–3310, 2020, international Conference on Nanotechnology: Ideas, Innovation and Industries. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785320333848

[7] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/2140

[8] H. A. Ignatious, Hesham-El-Sayed, and M. Khan, "An overview of sensors in autonomous vehicles," *Procedia Computer Science*, vol. 198, pp. 736–741, 2022, 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921025540

[9] S. Jain and I. Malhotra, "A review on obstacle avoidance techniques for self-driving vehicle," *International Journal of Advanced Science and Technology*, vol. 29, no. 06, pp. 5159–5167, 2020.

[10] J. Li, H. Bao, X. Han, F. Pan, W. Pan, F. Zhang, and D. Wang, "Real-time self-driving car navigation and obstacle avoidance using mobile 3d laser scanner and gnss," *Multimedia Tools and Applications*, vol. 76, pp. 23 017–23 039, 2017.

[11] N. S. Manikandan, G. Kaliyaperumal, and Y. Wang, "Ad hoc-obstacle avoidance-based navigation system using deep reinforcement learning for self-driving vehicles," *IEEE Access*, vol. 11, pp. 92 285–92 297, 2023.

[12] M. Pham and K. Xiong, "A survey on security attacks and defense techniques for connected and autonomous vehicles," *Computers & Security*, vol. 109, p. 102269, 2021.

[13] C. Wang, X. Yang, X. Xi, S. Nie, and P. Dong, *Introduction to LiDAR Remote Sensing*, 1st ed. CRC Press, 2024.

[14] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[15] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, no. 2015, p. 995, 2015.

[16] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *Cryptographic Hardware and Embedded Systems – CHES 2017*. Cham: Springer International Publishing, 2017, pp. 445–467.

[17] Y. Cao, S. H. Bhupathiraju, P. Naghavi, T. Sugawara, Z. M. Mao, and S. Rampazzi, "You can't see me: Physical removal attacks on LiDAR-based autonomous vehicles driving frameworks," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 2993–3010. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/cao

[18] Z. Hau, T. Kenneth, S. Demetriou, and E. C. Lupu, "Object removal attacks on lidar-based 3d object detectors," in *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, vol. 2021, 2021, p. 25.

[19] Y. Cao, S. H. Bhupathiraju, P. Naghavi, T. Sugawara, Z. M. Mao, and S. Rampazzi, "You can't see me: Physical removal attacks on LiDAR-based autonomous vehicles driving frameworks," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 2993–3010. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/cao

[20] S. Jeyachandran, "Meet the 6th-generation waymo driver: Optimized for costs, designed to handle more weather, and coming to riders faster than before," accessed: February 12, 2025. [Online]. Available: https://waymo.com/blog/2024/08/meet-the-6th-generation-waymo-driver

[21] Wayve, "Introducing radar: Wayve's sensor stack explained," accessed: February 12, 2025. [Online]. Available: https://wayve.ai/thinking/introducing-radar-wayves-lean-sensor-stack-explained/

[22] V. Vijayenthiran, "Gm's ultra cruise drive-assist tech employs 20 sensors, lidar," accessed: February 12, 2025. [Online]. Available: https://www.motorauthority.com/news/1134110_make-payments-from-your-mercedes-with-just-a-fingerprint

[23] T. A. Foundation, "Autoware," 2025, accessed: January 27, 2025. [Online]. Available: https://github.com/autowarefoundation/autoware

[24] ——, "Awsim labs," 2025, accessed: January 27, 2025. [Online]. Available: https://github.com/autowarefoundation/AWSIM-Labs

[25] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[26] B. Shahian Jahromi, T. Tulabandhula, and S. Cetin, "Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles," *Sensors*, vol. 19, no. 20, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/20/4357

[27] Y. Wang, Q. Mao, H. Zhu, J. Deng, Y. Zhang, J. Ji, H. Li, and Y. Zhang, "Multi-modal 3d object detection in autonomous driving: a survey," *International Journal of Computer Vision*, vol. 131, no. 8, pp. 2122–2152, 2023.

[28] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.

[29] J. Mao, S. Shi, X. Wang, and H. Li, "3d object detection for autonomous driving: A comprehensive survey," *International Journal of Computer Vision*, vol. 131, no. 8, pp. 1909–1963, 2023.

[30] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[31] W. Wang, Y. Yao, X. Liu, X. Li, P. Hao, and T. Zhu, "I can see the light: Attacks on autonomous vehicles using invisible lights," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1930–1944. [Online]. Available: https://doi.org/10.1145/3460120.3484766

[32] N. Wang, Y. Luo, T. Sato, K. Xu, and Q. A. Chen, "Does physical adversarial example really matter to autonomous driving? towards system-level effect of adversarial object evasion attack," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4389–4400.

[33] F. Xu, Y. Li, C. Yang, W. Wang, and B. Xu, "Adversarial attacks against traffic sign detection for autonomous driving," in *2023 7th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, 2023, pp. 1–6.

[34] R. Komissarov and A. Wool, "Spoofing attacks against vehicular fmcw radar," in *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*, ser. ASHES '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 91–97. [Online]. Available: https://doi.org/10.1145/3474376.3487283

[35] O. Toker, "Chapter 4 - radar architectures and cyberattacks from an autonomous vehicles perspective," in *Handbook of Power Electronics in Autonomous and Electric Vehicles*, M. H. Rashid, Ed. Academic Press, 2024, pp. 45–57. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323995450000051

[36] T. Sato, Y. Hayakawa, R. Suzuki, Y. Shiiki, K. Yoshioka, and Q. A. Chen, "Lidar spoofing meets the new-gen: Capability improvements, broken assumptions, and new attack strategies," *arXiv preprint arXiv:2303.10555*, 2023.

[37] Z. Jin, X. Ji, Y. Cheng, B. Yang, C. Yan, and W. Xu, "Laser-based lidar spoofing: Effects validation, capability quantification, and countermeasures," *IEEE Internet of Things Journal*, 2024.

[38] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[39] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1711–1719.

[40] B. Inc., "Apollo," https://github.com/ApolloAuto/apollo, 2025, accessed: January 27, 2025.

[41] Z. Hau, S. Demetriou, L. Muñoz-González, and E. C. Lupu, "Shadow-catcher: Looking into shadows to detect ghost objects in autonomous vehicle 3d sensing," in *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26*. Springer, 2021, pp. 691–711.

[42] Z. Hau, S. Demetriou, and E. C. Lupu, "Using 3d shadows to detect object hiding attacks on autonomous vehicle perception," in *2022 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2022, pp. 229–235.

[43] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial

sensor attack and countermeasures," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 877–894.
[44] C. You, Z. Hau, and S. Demetriou, "Temporal consistency checks to detect lidar spoofing attacks on autonomous vehicle perception," in *Proceedings of the 1st Workshop on Security and Privacy for Mobile AI*, 2021, pp. 13–18.
[45] J. Liu and J.-M. Park, ""seeing is not always believing": detecting perception error attacks against autonomous vehicles," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2209–2223, 2021.
[46] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.
[47] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*. IEEE, 2020, pp. 1–6.
[48] Y. Li, W. Yuan, S. Zhang, W. Yan, Q. Shen, C. Wang, and M. Yang, "Choose your simulator wisely: A review on open-source simulators for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2024.
[49] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," in *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*. IEEE, 2021, pp. 62–70.
[50] Y. Cao, J. Ma, K. Fu, R. Sara, and M. Mao, "Automated tracking system for lidar spoofing attacks on moving targets," in *Proc. Workshop Automot. Auto. Vehicle Secur.(AutoSec)*, 2021, p. 1.
[51] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
[52] I. Ouster, *VLP-16 User Manual*, 2022. [Online]. Available: https://data.ouster.io/downloads/velodyne/user-manual/vlp-16-user-manual-revf.pdf
[53] O. S. R. Foundation, "Velodyne pointcloud ros 2 package." [Online]. Available: https://index.ros.org/p/velodyne_pointcloud/
[54] T.-H. Kim and T.-H. Park, "Placement optimization of multiple lidar sensors for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 2139–2145, 2020.
[55] R. R. Kini, "Sensor position optimization for multiple lidars in autonomous vehicles," 2020.
[56] Unity Technologies, "Unity," 2025, game development platform. [Online]. Available: https://unity.com/
[57] RobotecAI, "Robotecgpulidar." [Online]. Available: https://github.com/RobotecAI/RobotecGPULidar
[58] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.
[59] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.