SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants

Raul Quinonez\textsuperscript{1}, Jairo Giraldo\textsuperscript{4}, Luiz Salazar\textsuperscript{2}, Erick Bauman\textsuperscript{1}, Alvaro Cardenas\textsuperscript{2}, Zhiqiang Lin\textsuperscript{3}

\textsuperscript{1}The University of Texas at Dallas
\textsuperscript{2}University of California Santa Cruz
\textsuperscript{3}Ohio State University
\textsuperscript{4}University of Utah

USENIX Security 2020
Autonomous Vehicles (AVs) include aerial, sea, and ground vehicles.
Levels of automation range from 0 to 5.
AVs evaluate their environment with a variety of sensors.

[Gon17]
GPS Spoofing Mystery Affirms Need for Protection
The spoofing of GPS systems at the Geneva Motor Show gave us an unfortunate example of how vulnerable vehicles are to “spoofery.”

[MIT19]

Researchers trick Tesla Autopilot into steering into oncoming traffic
Researchers trick Tesla Autopilot into steering into oncoming traffic. Stickers that are invisible to drivers and fool autopilot. Dan Goodin - 4/1/2019, ...

[GOO19]

Autonomous vehicles can be fooled to ‘see’ nonexistent obstacles
BY YULONG CAO, Z. MORLEY MAO | MAR 06, 2020

[YC20]

Mysterious GPS glitch telling ships they're parked at airport may be anti-drone measure
Elizabeth Weise, USA TODAY Published 1:41 p.m. ET Sept. 26, 2017 | Updated 3:03 p.m. ET Oct. 3, 2017

[WEI17]
AVs Are Vulnerable to Sensor Targeted Attacks

Main Problem
- AVs rely on sensors to evaluate and interact with their environment.
- Sensors are susceptible to GPS spoofing and transduction attacks that manipulate environmental physical signals.

Previous Research Has Exposed Sensor Vulnerabilities
- Camera [DWJ\textsuperscript{+16}, PSFK\textsuperscript{15}, YXL\textsuperscript{16}]
- LiDAR [PSFK\textsuperscript{15}, SKKK\textsuperscript{17}, CXC\textsuperscript{+19}]
- RADAR [YXL\textsuperscript{16}]
- Inertial Measurement Unit (IMU) [SSK\textsuperscript{+15}, TWX\textsuperscript{+17}, TLLH\textsuperscript{18}]
- GPS [NKS\textsuperscript{+19}, HLP\textsuperscript{+08}, TPRC\textsuperscript{11}, ZLS\textsuperscript{+18}]
Transduction Attacks and GPS Attacks Cannot be Addressed with Classical Security

Transduction Attacks

Attacker can inject out-of-band signals

GPS Attacks

Spoofer Actual position Spoofed position
We introduce our SAVIOR (Securing Autonomous Vehicles wIth rObust physical invaRiants) framework contributing to the following:

1. We use well-known nonlinear dynamic models for aerial and ground AVs
2. We introduce a stronger stealthy attacker
3. We implement a Cumulative Sum (CUSUM) algorithm that improves detection performance over previous defenses that keep track of anomalies using time windows
4. The implementation is done in real vehicles including including an Intel drone, and our autonomous car
Sensors and Movement Variables

**Drones**
- 3 axes: roll, pitch, yaw
- Sensors: accelerometer, gyroscope, magnetometer, and GPS (lat, lon, alt)

**Ground AV**
- 2 axes: pitch, yaw
- Sensors: line data (angle, position) and speed
Nonlinear Models

Dynamics of a Quadcopter [CFCH14, Luu11]

\[ \dot{\phi} = \omega_\phi \]
\[ \dot{\theta} = \omega_\theta \]
\[ \dot{\psi} = \omega_\psi \]
\[ \dot{\omega}_\phi = \frac{U_\phi}{I_x} + \dot{\theta} \dot{\psi} \left( \frac{l_y - l_z}{l_x} \right) \]
\[ \dot{\omega}_\theta = \frac{U_\theta}{I_y} + \dot{\phi} \dot{\psi} \left( \frac{l_z - l_x}{l_y} \right) \]
\[ \dot{\omega}_\psi = \frac{U_\psi}{I_z} + \dot{\phi} \dot{\theta} \left( \frac{l_x - l_y}{l_z} \right) \]
\[ \dot{x} = v_x \]
\[ \dot{y} = v_y \]
\[ \dot{z} = v_z \]
\[ \dot{v}_x = \frac{U_t}{m} \left( \cos \phi \sin \theta \cos \psi + \sin \theta \sin \psi \right) \]
\[ \dot{v}_y = \frac{U_t}{m} \left( \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \right) \]
\[ \dot{v}_z = \frac{U_t}{m} \cos \phi \cos \theta - g \]

Dynamics of a Car [KPSB15]

\[ \beta = \tan^{-1} \left( \frac{l_r}{l_r + l_f} \tan(\delta) \right) \]
\[ \dot{x} = v \cos(\psi + \beta) \]
\[ \dot{y} = v \sin(\psi + \beta) \]
\[ \dot{\psi} = \frac{v}{l_r} \sin(\beta) \]
\[ \dot{v} = a \]
SAVIOR Design

- Online sensor pre-processing to convert raw data into usable form
- Offline pre-processing stage to learn physical invariants and a build model
- Online stage to predict measurements and compare observe values
- Anomaly detection will raise an alert if the anomaly is persistent
An Extended Kalman Filter (EKF) [RG14] is used to predict AV’s physical behavior by estimating unknown parameters from noisy sensor input.

The algorithm is divided into two main routines: prediction and correction.

The prediction will be compared against the observed data to be analyzed for sensor tampering.

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted state estimate</td>
<td>Kalman gain</td>
</tr>
<tr>
<td>$\hat{x}_{k+1} = f(\hat{x}_k, u_k)$</td>
<td>$K_k = P_k^- H_k^T [H_k P_k^- H_k + V]^{-1}$</td>
</tr>
<tr>
<td>Predicted covariance estimate</td>
<td>State correction</td>
</tr>
<tr>
<td>$P_k^- = F_k P_{k-1} F_k^T + Q$</td>
<td>$\hat{x}<em>{k+1} = \hat{x}</em>{k+1} + K_k (y_k - h(\hat{x}_k^-))$</td>
</tr>
<tr>
<td></td>
<td>Updated covariance estimate</td>
</tr>
<tr>
<td></td>
<td>$P_k = [I - K_k H_k] P_k^-$</td>
</tr>
</tbody>
</table>

Initial estimates
For $\hat{x}_0^-, P_0^-$
Anomaly Detection

- The residual associated with each sensor is calculated\(^{(1)}\)
- A Cumulative Sum (CUSUM) algorithm is then used to detect persistent attacks\(^{(2)}\)
- An alarm is raised if the residual difference is larger than a predefined threshold\(^{(3)}\)

**CUSUM Algorithm**

1. \[ r_i(k) = \tilde{Y}_i(k) - \hat{Y}_i(k) \]
2. \[ S_i(k+1) = (S_i(k) + |r_i(k)| - b_i)^+ \]
3. \[ S_i(t_k) > \tau_i \]

\( \tilde{Y}_k \), \( \tilde{Y}_{k-1} \), \( u_k \) to \( r_k = \tilde{Y}_k - \hat{Y}_k \), CUSUM, Alert
Controllers follow a publish-and-subscribe architecture to provide inter-process communication via topics.

We are interested in the following topics for aerial AVs: sensors_combined, vehicle_magnetometer, and vehicle_gps_position.

Anomaly detector is situated right before the control signals are being sent to the actuators.

The code runs in its own module in parallel with the controller.
Evaluation

- Aerial AV: Intel Ready-To-Fly drone using PX4 flight controller (v1.9.2)
- Ground AV: Custom build on top of a Traxxas Ford Fiesta ST Rally chassis using ROS Kinetic Kame controller
Ground AV Camera Attack and Detection Video

Videos available: https://www.youtube.com/watch?v=Ljrbtfo0gvM&list=PLmicm3IoL28eLU5v1FH3Z0FSn5N10uQLG
Aerial AV GPS Attack and Detection

Flight started
Attack started
GPS timeout
Desired destination
Actual landing site

Position X
Position Y
Detection Statistic

Time (sec)
0 50 100
0 50 100
0 50 100

Detection detected
Attack detected

Attack detected
Attack detected
Attack detected
Comparison of SAVIOR with Baseline

- SAVIOR uses a nonlinear model for predicting the observations, and a CUSUM algorithm for anomaly detection (NLC)
- We will use Choi et al.’s [CLA+18] algorithm as a baseline since their anomaly detector was the current state-of-the-art
- Choi et al.’s [CLA+18] algorithm uses linear models for predicting observations and a Time-Window algorithm for anomaly detection (LTW)
- Our results show that our algorithms outperform state-of-the-art detection tools for AVs by detecting more attacks, detecting attacks faster, and having less false alarms
Linear (LTW) vs Nonlinear (NLC) Prediction Comparison

![Graphs showing comparison of Linear and Nonlinear predictions for Roll, Pitch, Position X, and Position Y over time.](image-url)
Window (LTW) vs CUSUM (NLC) Detection Time and ROC Curves

**Drone**
- NLC detects attacks faster
- NLC has a better ROC curve than LTW

**Ground AV**
- Detection is better for both, drones and ground vehicles
Stealthy Attacks

- We want to maximize the value of sensor tampering without raising any alarms.
- The goal is to maximize deviation without increasing the added discrepancies.
- This stealthy attack allows us to consider the worst case scenario of our PBAD system, where an attacker is not detected while it persistently injects the maximum amount of false information in the system.
Purely Stealthy Attacks Against NLC Have Less Impact Than LTW

- NLC (blue) is able to follow the signal closer while the attacker performed an stealthy attack on the gyroscope and GPS.
- LTW (orange) allows more tampering which ends up deviating the final destination more than NLC (blue).
### Performance Overhead

**Drone**

On average, SAVIOR consumes 5.4332% of CPU resources on Intel Aero

<table>
<thead>
<tr>
<th>Module</th>
<th>Armed</th>
<th>Hovering</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>30.1444%</td>
<td>29.4379%</td>
<td>30.6056%</td>
</tr>
<tr>
<td>mavlink_if1</td>
<td>16.0183%</td>
<td>15.6195%</td>
<td>15.8956%</td>
</tr>
<tr>
<td>EKF2</td>
<td>14.3242%</td>
<td>14.3779%</td>
<td>14.3006%</td>
</tr>
<tr>
<td>logger</td>
<td>6.8647%</td>
<td>7.1288%</td>
<td>6.8752%</td>
</tr>
<tr>
<td>mc_att_control</td>
<td>5.4349%</td>
<td>5.4007%</td>
<td>5.3425%</td>
</tr>
<tr>
<td>reference_monitor</td>
<td>5.3572%</td>
<td>5.4332%</td>
<td>5.5093%</td>
</tr>
<tr>
<td>tap_esc</td>
<td>4.4742%</td>
<td>4.4357%</td>
<td>4.4285%</td>
</tr>
<tr>
<td>sensors</td>
<td>4.2744%</td>
<td>4.4792%</td>
<td>4.5200%</td>
</tr>
<tr>
<td>hpwork</td>
<td>2.5077%</td>
<td>2.4462%</td>
<td>2.4750%</td>
</tr>
<tr>
<td>mavlink_if0</td>
<td>2.3323%</td>
<td>2.1384%</td>
<td>2.2667%</td>
</tr>
<tr>
<td>mc_pos_control</td>
<td>1.4911%</td>
<td>2.4727%</td>
<td>1.4693%</td>
</tr>
<tr>
<td>commander</td>
<td>1.4824%</td>
<td>1.4478%</td>
<td>1.4448%</td>
</tr>
<tr>
<td>gps</td>
<td>0.3662%</td>
<td>0.3323%</td>
<td>0.3077%</td>
</tr>
</tbody>
</table>

**Ground AV**

On average, SAVIOR consumes 2.2501% of CPU resources on Traxxas Ford Fiesta ST Rally

<table>
<thead>
<tr>
<th>Module</th>
<th>Line Following</th>
<th>CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>lidar_collision_avoidance</td>
<td>12.6886%</td>
<td>13.0694%</td>
</tr>
<tr>
<td>elp_cam_bridge</td>
<td>11.0179%</td>
<td>15.6009%</td>
</tr>
<tr>
<td>process_line</td>
<td>10.3861%</td>
<td>11.7353%</td>
</tr>
<tr>
<td>image_processing</td>
<td>6.0726%</td>
<td>7.8523%</td>
</tr>
<tr>
<td>reference_monitor</td>
<td>2.5192%</td>
<td>1.9809%</td>
</tr>
<tr>
<td>arduino_node</td>
<td>2.4150%</td>
<td>2.5133%</td>
</tr>
<tr>
<td>line_follower</td>
<td>1.0097%</td>
<td>1.0488%</td>
</tr>
<tr>
<td>low_level_controller</td>
<td>0.7948%</td>
<td>0.4503%</td>
</tr>
<tr>
<td>perot_demo</td>
<td>0.6990%</td>
<td>0.6589%</td>
</tr>
<tr>
<td>rplidarDemo</td>
<td>0.4541%</td>
<td>0.2678%</td>
</tr>
<tr>
<td>rplidarNode</td>
<td>0.3074%</td>
<td>0.3020%</td>
</tr>
<tr>
<td>rosmaster</td>
<td>0.2973%</td>
<td>0.1569%</td>
</tr>
<tr>
<td>rosout</td>
<td>0.0658%</td>
<td>0.0250%</td>
</tr>
</tbody>
</table>
Conclusion

The Key Elements of Our Proposal

1. Use of well-known physical invariants
2. The use of offline system identification
3. The use of CUSUM algorithms
4. Evaluating the effectiveness of the anomaly detection tool with stealthy attacks that attempt to maximize the damage to the system

SAVIOR Source Code
https://github.com/Cyphysecurity/SAVIOR.git

Videos
https://www.youtube.com/watch?v=Ljrbtfo0gvM&list=PLmicm3IoL28eLU5v1FH3Z0FSn5N10uQLG
Thank You

Contact

Raul Quinonez  rxq100020@utdallas.edu
Jairo Giraldo  jairo.giraldo@utah.edu
Luiz Salazar  luedsala@ucsc.edu
Erick Bauman  exb131030@utdallas.edu
Alvaro Cardenas  alacarde@ucsc.edu
Zhiqiang Lin  zlin@cse.ohio-state.edu
References I


Hongjun Choi, Wen-Chuan Lee, Yousra Aafer, Fan Fei, Zhan Tu, Xiangyu Zhang, Dongyan Xu, and Xinyan Xinyan, Detecting attacks against robotic vehicles: A control invariant approach, Conference on Computer and Communications Security (CCS), ACM, 2018, pp. 801–816.


References II


