HoneyBot: A Honeypot For Robotic Systems

Celine Irvene, David Formby, Samuel Litchfield, Raheem Beyah
Communications Assurance & Performance Group, Twitter: @GTCAPGROUP, CAP URL: www.ece.gatech.edu/cap
School of Electrical and Computer Engineering, Georgia Institute of Technology

Abstract

- Historically, robotics systems have not been built with an emphasis on security
- Their main purpose has been to complete a specific objective
- As more and more robotic systems become remotely accessible through networks, they are more vulnerable than ever
- HoneyBot: The first software hybrid interaction honeypot
  - Specifically designed for networked robotic systems
  - Simulates unsafe actions and physically performs safe actions to fool attackers into believing their exploits are successful
  - Logs all communication to be used for attribution and threat model creation

Methodology

- Reconnaissance: Studied the building blocks of robotics and grouped them into three categories:
  - Sensors: enable the robot to judge features of the environment
  - Actuators: enable the robot to modify the environment and move
  - Controller: enables the robot to “think”
- HoneyBot Simulator: Build software GUI to establish the importance and feasibility of the HoneyBot
- Sensor Correlation and Model Creation: Reconnaissance revealed that the most important factor to being a believable robot is yielding sensor data that corresponds to the actions and environment of the robotic system
  - Ultrasonic Sonar: Experiments were performed to measure ultrasonic sonar responses attached to a robot driving at different speeds towards a static target. Speed was determined to be an independent factor
  - Collision Sensor: This device model is triggered by reading from the sonar and infrared distance sensors. If below an empirically determined threshold it renders a collision event
  - Infrared Distance Sensor: We built a plot of voltages outputted from the sensor versus distance (measured manually) and derived an equation from the fit of the line

Results

- A combination of experimentation and physical process modeling was used to simulate device behavior.
- The goal was to query these models at runtime and generate “spoofed” responses
- Ultrasonic Sonar: Experiments were performed to measure ultrasonic sonar responses attached to a robot driving at different speeds towards a static target. Speed was determined to be an independent factor
- Collision Sensor: This device model is triggered by reading from the sonar and infrared distance sensors. If below an empirically determined threshold it renders a collision event
- Infrared Distance Sensor: We built a plot of voltages outputted from the sensor versus distance (measured manually) and derived an equation from the fit of the line

Evaluation

- Experiment: Surveyed user experiment used to test effectiveness of HoneyBot
  - Participants connected to web GUI and navigated an onscreen robot through a maze
  - This physically navigated the HoneyBot through an identical maze in a lab environment.
  - Participants had access to robot sensor and no explicit instructions were given regarding danger signs.
- Subjects weren’t aware danger signs indicated shortcuts though the maze and triggered HoneyBot simulations

Results: Collected from user surveys

<table>
<thead>
<tr>
<th>Scale (1 is very inaccurate and 5 is very accurate)</th>
<th>How accurate did the sensor values displayed on the control panel seem throughout the experiment?</th>
<th>How accurate did the sensor values displayed on the control panel seem after you crossed through danger?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 (2.5%)</td>
<td>1 (7.14%)</td>
</tr>
<tr>
<td>2</td>
<td>4 (10%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>3</td>
<td>13 (32.5%)</td>
<td>3 (21.43%)</td>
</tr>
<tr>
<td>4</td>
<td>15 (37.5%)</td>
<td>6 (42.86%)</td>
</tr>
<tr>
<td>5</td>
<td>7 (37.5%)</td>
<td>4 (26.57%)</td>
</tr>
<tr>
<td>Total</td>
<td>40 (100%)</td>
<td>14 (100%)</td>
</tr>
</tbody>
</table>