Project Proposals

Project Proposal 1

The Robotic Operating System (ROS) is a flexible framework for developing robotic software [1]. ROS is not technically an operating system and acts more as a framework which manages and provides services commonly required in robotic applications. These include sending messages between components, managing application parameters, managing large libraries of common functionality and robotic networking.

ROS, however, was designed with little to no security in mind. ROS has no authentication for either code running (nodes) or messages sent by the nodes. This lack of security means that any node can run and send messages to the robot. ROS also has no encryption thus messages are readable to anyone connected to the network, including sensitive sensor data. No security implies anyone with a connection to the network can read and change the behaviors of the robot.

Research has been done into providing ROS with standard security features [3]. This method, however, requires changing the underlying ROS protocol to allow for node and message verification. More research has been done to look at, and secure publicly networked ROS nodes [2, 4].

An intriguing research project would be to develop a method that can verify messages and nodes without changing the underlying ROS architecture. It would also be interesting to see what kind of actions a hacker would need to perform to trick the developed verification technique to still allow erroneous behavior.

I propose that we develop a deep network which monitors a system (nodes launched, actions done, messages sent) and identifies when a robotic application is acting erroneously or under attack. This can be done by developing a system, and making it perform a set of actions. These actions will be recorded and fed into the deep network. The network will then learn what is considered “correct” and “incorrect” behavior. Knowing this will allow developers to act upon this information and tell whether a system is failing for any reason, including attack.

Further research can be done into looking at what kind of erroneous behavior is misclassified as correct behavior. This can be thought of as real-world adversarial examples, where the adversarial example is a set of robot actions.

Project Proposal 2

This proposal requires more insight and guidance.

Drones are becoming increasingly popular. One of the significant features of drones is the return to home command. This command automatically flies a drone back to a predetermined location in the case of low battery, emergency or a user request.

If we were able to modify the memory location of the "Home" variable at the time of hardware flashing, we could, in essence, modify the location the drone will fly in the case of an emergency, user request, or low battery. This change will be completely unnoticeable to the end user, however, could lead to disastrous effects. Attackers could fly the drone into populated areas, predetermined locations (where they could steal the drone) and numerous other terrible scenarios.

This same modification of memory could be used for many other attacks, such as changing the flight control parameters, battery reading parameters or numerous other parameters used to fly drones.

This project would require me to identify the memory location of the variables used by the hardware flashing programs. I thought that a program like Pintool might suffice. However, it would be preferable to have an attack which is independent of the executable used. I would then need a program which can change the memory location before a user can flash the drone. A tool such as Row Hammer [5] might be able to do this. However, I am open to suggestions.

References

1) ROS (http://www.ros.org/about-ros/)
2) Message Authentication Codes for Secure Remote Non-Native Client Connections to ROS Enabled Robots (https://web.cs.wpi.edu/~cshue/research/tepra14.pdf)
3) Security for the Robot Operating System (https://www.researchgate.net/publication/320368007_Security_for_the_Robot_Operating_System)
4) Scanning the Internet for ROS: A View of Security in Robotics Research (http://hcr.mines.edu/2018-rss-workshop/abstracts/RSS18WS_scanning_the_internet_for_ros_a_view_of_security_in_robotics_research.pdf)
5) Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (http://users.ece.cmu.edu/~yoonguk/papers/kim-isca14.pdf)