Security for Robotics

Emily Longman, Zach Rogers, and Dominic Giacoppe

CS461 Capstone

10/13/2016

Sponsor	Date
Group Member	Date
Group Member	Date
Group Member	Date

Abstract

In drones and other networked robotics there is a broad array of security vulnerabilities that can be leveraged in an attack, leaving the potential for disaster. To attempt to prevent and mitigate these we evaluated ROS on a drone to find security holes and document them. The different vulnerabilities found were categorized into malware, sensor hacks, network and control channel attacks, and physical attacks. For some of these attacks were able to implement solutions, which were also documented. These findings and any solutions were added to an ongoing academic effort to make robotics more secure.

Problem Definition

Robotics is still a relatively up and coming field and as most efforts in robotics are in pursuit of furthering capabilities, security has been left largely untouched. Because of this many functioning, deployed robots have limited to no security in their internal systems, making them very vulnerable to attacks. While the community developing robotics is still largely academic and there is little worry of being attacked, the security vunerbilities still exist. Soon robotics will become ubiquitous in society and hackers will exploit these vulnerabilities for personal gain. There have already been reports of smart appliances being used for botnets, it's only a matter of time before drones and other robotics are similarly abused. Through our work we hope to eliminate some of this abuse before it begins.

Since robotics are continually in development and there is such a wide variety of devices we have to focus our efforts onto a small subset of robotics to have any hope of making progress within a year. While most likely subject to change, we are currently hoping to focus on the weaknesses of drones. We can choose focus on the risks within Robotic Operating System (ROS) or the currently in development secure version, SROS. These operating systems can be attacked at the driver level with malware, they can have the configuration files modified, and they can have data intercepted or spoofed in their internal communication subsystems. We could also investigate the ability to spoof sensor data, such as the camera, IR guidance, accelerometer, or gyroscopic systems. Outside of the OS level there is a huge amount of room for exploitation in the communication and control channels used by networked devices. Even physical attacks are something that needs to be addressed as a security concern. It's very important that we ackowledge all of these risks, since exploitation of them could be a big setback for global trust of robotics.

Essentially the problem we face is twofold; first we need to isolate a specific feature of a specific device, and then we need to attempt to break that feature, then document our work. This could be a crucial improvement for the use of robotics running ROS in a wide variety of sectors. The military uses and plans to use them widely, and for them more than anyone they need to be as secure as possible. Amazon and other commercial shipping companies have a vast use of robotics, and we've all heard about their plans for delivery drones, which are a huge security risk. If consumers think these will be abused they won't trust this upcoming technology and it will be slowly or never adopted, which will make companies hesitant to invest in their development. Consequentially, the world as a whole will be slower to advance robotic technology. Hopefully our work in securing robotics can help to better develop consumer trust in this emerging technology.

Proposed Solution

The minimal solution we hope to reach is identifying and proving that a vulnerability exists in some sector of a device. This would include full documentation and research reports of our findings, specifically a threat model analysis based off Adam Shostack's methods. [1, p.203]

If possible we hope to find a reasonable fix for the vulnerability and implement it without noticeably hurting the functionality of the device. This implementation would follow any regulations in place and should not introduce other security holes. If a successful fix were to be created it would need to be fully documented and tested.

Performance Metrics

Because this is such a nebulous problem at the moment, it's hard to define concrete metrics by which to measure our success. The best way to measure our efforts seems to simply be documentation. We're unsure if we even will be able to isolate and exploit a vulnerability successfully, but we can still record everything we try and our findings to help better the academic community. Failure for us would mean that we not only were unable to crack a security hole, but we also failed to record our process and attempts, or didn't gather necessary data.

References

[1] A. Shostack, *Threat Modeling, Designing for Security.* Indianapolis, Indiana: John Wiley and Sons, Inc, 2014.