Security for Robotics

Emily Longman, Zach Rogers, and Dominic Giacoppe

CS461 Capstone

10/13/2016

Abstract

In drones and other networked robotics there is a broad array of security vulnerabilities that can be leveraged in an attack. We evaluated the ROS and some real-time drone operating systems to find as many of these security holes as we could 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 so there is minimal security present in the technology available today. Because of this they are very vulnerable to attacks on systems that were designed without any preexisting security protocols to follow. While the community developing robotics is still largely academic and there is little worry of being attacked, the security holes still exist. It's only a matter of time before robotics become ubiquitous in society and hackers exploit these vulnerabilities for personal gain. There have already been reports of smart appliances being used for bot nets, it's only a matter of time before drones and other robotics are similarly abused.

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) if the device is running it, or an a real-time OS like nuttx. 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 the publish/subscribe system ROS uses. 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.

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. We need to document the specifics of how we break it and why we were able to. Assuming we are able to do that, we should attempt to find a fix for that weakness which doesn't adversely affect the functionality of the system. This would mean the type of attack we used would no longer be effective, and there would be no noticeable difference in performance to the user.

Proposed Solution

As stated above, the specific problem is still very vague at this point. 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 would hope to find some sort of 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.