A SURVEY OF SECURITY IN ROBOTIC SYSTEMS: VULNERABILITIES, ATTACKS, AND SOLUTIONS

Christopher Archibald,* Luke Schwalm,* and John E. Ball**

Abstract

Robotic systems are rapidly becoming more prolific and being used for an increasing range of tasks. The more integrated they are with sensitive tasks and with their users’ lives and likelihoods, the greater desirability they possess as targets to would-be attackers. Security has not yet been a major focus within robotics. This article surveys literature related to robotic systems security, specifically identifying vulnerabilities, attacks, and solutions in order to help roboticists understand why to focus on security throughout their design, potential problems they could face, and steps they can take to address these.

Key Words

Robotics, security, hacking, attacks, networking, sensors, drones, spoofing, encryption

1. Introduction

As robots become more sophisticated, capable, intelligent, and networked, they are being used for a broadening range of everyday tasks. Even non-traditional robots, like cars, are becoming equipped with more computing devices, sensors, and connections with the cyber-world around them. As they rise in commonality and importance it must be remembered that robots are essentially just computers with advanced means of interacting with the physical world. Much like regular computers, robots can be targeted by numerous security attacks with all sorts of malicious goals. Roboticists must consider how these devices can be made secure. As will be demonstrated, security in robotics is an important topic that has often been neglected.

This article surveys the state of robot security, including vulnerabilities of robotic security found in the literature, example demonstrated attacks which exploit some of these vulnerabilities, and finally best principles and practices for designing and fielding secure robots. The main take-away is that security must be a core consideration in any robotics application. The goal of the articles is to increase awareness of potential security issues in the robotics community and providing knowledge and tools to address these issues during the design and deployment of robotic systems.

2. Robotic Systems

A robot is a system with sensors, actuators, and computing ability. Robots are often mobile, carry out various tasks, and can be categorized in many ways. Domestic robots operate inside or near a home or business and are often administered by inexpert users [1], [2]. Utility robots generally operate outdoors in high risk or high threat scenarios. Domestic robots (e.g. Roomba [3]) are often used indoors and are less sophisticated and expensive than utility robots. Utility robots can be used for military, surveillance, crisis response, search and rescue, mapping, and other rigorous task genres. Utility robots (e.g. MQ-9 Predator or Dragonfly Guardian [4], [5]) are generally more expensive, more sophisticated, and more robust than their domestic counterparts. Swarm robots are groups of intelligent, autonomous robots operating in a decentralized manner that have use in military, surveillance, disaster relief, and commercial settings [6].

Modern car system security is included because cars fit the definition of a robot and they are an example of a large complex system which can sense the environment (wheel speed, tire pressure, GPS location) and take actions (automatic braking, information display, unlocking car). Additionally, their ubiquity and piecemeal development shed light on potential future scenarios involving widely adopted robotic systems, teaching valuable security lessons.

3. Robotic System Vulnerabilities

An overview of robotic system security vulnerabilities, organized into several types, is now provided.
3.1 Physical Vulnerabilities

Physical vulnerabilities must be exploited by physical access to or contact with the robot or controlling devices. Physical access can result in robot components being re-programmed or tapped into [7]–[9] robots being rendered unavailable or being modified to grant an adversary control [10]. A number of physical vulnerabilities exist in cars, including diagnostic tools for used by mechanics [11], [12] and devices like CD players, which have both been used to gain full control over a car’s systems [11]. Passive keyless entry systems in cars are vulnerable to relay attacks [13]. Relay attacks capture signals from a car’s keys and transmit them to a device close to the car, which gains complete access by pretending to be the keys. Swarm robots are vulnerable to the physical introduction of foreign robots or other entities into the swarm or environment [10]. Safe home or workplace environments can be compromised by the introduction of infected robots or electronic devices which can infect other robotic systems in the area [14].

3.2 Sensor Vulnerabilities

Sensors are vulnerable to adversary-manipulated signals. One example is GPS-spoofing, which gets a GPS-sensor to use fake GPS signals resulting in false location information [15]. This has been done to drones [16], [17] and boats [18]. Cars’ automatic braking systems can get false wheel speed information from magnetic devices attached near tires [15], [19], leading to crashes in simulated experiments. Adversary-generated signals can interfere with sensors, leading to invalid or no output [7], [20].

3.3 Communication Vulnerabilities

Many robots rely on communication with either user or outside world, and these communication methods present vulnerabilities, especially when tried and true security measures are not followed [21].

3.3.1 Types of communication

Robots employ many different communication methods: ad-hoc networks or the Internet [1], [22]; Bluetooth for short-range connection of personal computing devices and phones [22]; wireless sensors (e.g. Tire Pressure Monitoring System in cars) [11], [21]; and RFID for wireless communication [11]. Long-range communication channels include satellite and digital radio and traffic status channels [11], where information is mainly received. Other channels are for crash-reporting, anti-theft car-tracking, vehicle diagnostics, and user convenience (e.g. GM’s OnStar), which require information transmission [11]. Future robotic systems might have similar long-range monitoring and communication capabilities.

3.3.2 Passive Adversary Vulnerabilities

Information about a robot (e.g. is robot present?) can be passively gathered from communication channels. This can be done using packet interception or injection over the local network or Internet [1], [23]. Search engines could conceivably discover Internet-connected robots much like was done with webcams [24]. By intercepting traffic on a communication channel, or eavesdropping [1], [12], adversaries can gain more information such as which robots are active; what they are doing; where they are [12], [21]; or sensitive user information like audio or video data from sensors, or unencrypted usernames and passwords [1].

3.3.3 Active Adversary Vulnerabilities

Many communication vulnerabilities involve a more active adversary, such as intercepting legitimate network traffic and/or transmitting illegitimate traffic. Dropped messages are deleted and do not reach their destination [12], [25]. Intercepted messages can be retransmitted later, which is called a replay attack [12], [25]. Illegitimate messages, either completely constructed or modified, can be transmitted, which is message spoofing [12], [21], [25]. A masquerade attack involves an adversary imitating an authorized party [12]. A man-in-the-middle attack, or double masquerade, occurs when robot-recipient traffic is intercepted [12], [25]. The popular Robot Operating System (ROS) is vulnerable to this attack [26]. Communication channels can also be closed down either through jamming [10], [14] or Denial of Service (DoS) attacks [8], [12], [27], [28]. DoS attacks flood a network with messages, preventing appropriate handling of legitimate traffic.

3.4 Software Vulnerabilities

Any software has vulnerabilities, stemming from poor programming practices or from lack of security consideration during design. For example, the CAN bus in cars, the electronic connection between different components, was designed without a focus on security [8], [12], [23]. Subsequent security protocols have often been poorly implemented and misused [8], [23]. ROS suffers from similar design issues. ROS nodes are killed when another (i.e. illegitimate) node with the same name is connected to the ROS network [26]. Connected nodes could also generate fake messages on a topic, which are published without any authentication or validation [26]. ROS also has a single point of failure, the ROS master node [26].

3.5 System-level Vulnerabilities

Security vulnerabilities can arise in robotics systems as they become more complex, integrating various subsystems from different manufacturers and designers, and increasing the opportunities for security problems at the interfaces due to sub-system interactions [29]. This has been true for cars [11]. These difficulties will also be the case as robotic systems become more complicated.

3.6 User Vulnerabilities

A robot’s users and environment can impact its security. In chaotic, cluttered environments an adversary could discretely deploy or modify a robot [1]. Users with special
needs, elderly people, or children will likely be among service robots’ early users and are less likely to have security experience [2], [22], [29], making them more vulnerable to attacks which wouldn’t succeed against more experienced users. The robot–user interaction, for example, how a robot and user get feedback from each other, is a potential source of vulnerabilities. An adversary could modify this behaviour in either direction, causing a user to issue unnecessary or repeated commands [25], or the robot to behave abnormally or dangerously [25]. A user’s expectations about a robot will also influence potential vulnerabilities [1]. Studies have found that humans take different privacy measures around a camera than a robot [30]. Robotic designers and marketers need to help users create proper mental models of what robots are capable of, as well as subsequent security implications. Confidence in security will impact how readily humans adopt robotic technology [31]. Few security solutions can replace a vigilant and security-aware user.

4. Potential Robotic Systems Attacks

Specific attacks are now outlined, each exploiting one or several of the previous vulnerabilities.

4.1 Control-based Physical Attacks

Many robot security attacks seek to gain control over the robot or some of its sub-systems. Related vulnerabilities are now highlighted, as well as subsequent attacks this can facilitate. Hackers can gain control of remotely controlled robots through ill-gotten legitimate credentials or by man-in-the-middle attacks [1]. Full control over a car can be gained through numerous sub-systems, including diagnostic checkers, CD in the entertainment system, Bluetooth, remote keyless entry, tire pressure monitoring system, RFID car keys, 802.11 WiFi, and the in-car phone system [11]. Most attacks were fairly involved, but were able to gain control of the car’s system. Control can be gained over car radio and instrument display panels, brakes, turning the car off, and the car’s HVAC system [8]. Cars can be opened and driven away using a relay attack [13], or have dashboard warnings triggered in the car, distracting or delaying a user [21].

Possible control-based attacks include: vandalizing home or other property; spying on homes or people; directly harming humans, especially vulnerable children or elderly [1]; terrorizing pets or children while adults are away; moving items or creating sounds to unhinge users [1]; opening a door from the inside, providing access to the adversary; manipulating home devices (e.g. thermostat) to incur large costs [14]; coordinating attacks to overload power grids [14]; or incapacitating itself so it can no longer care for the elderly [14]. The capabilities of multiple compromised robots can be uniquely combined to create new attacks. Researchers used one robot to pick up keys and hold them for another’s webcam, which then duplicated the keys, granting access to the home [1]. A high profile real-world control-based attack occurred in 2011 when a US Drone was hijacked by Iran [17], [32]. Attackers spoofed the drone’s GPS system, causing it to mistake an Iranian location for a US military base in Afghanistan [12], [32].

4.2 Informational Attacks

Informational attacks leverage access to robot or sensor information. Such information could reveal presence in a home of owner’s or their valuables [14]. Knowledge of an expensive robot’s presence could attract thieves. Vulnerable cars could be mapped and access to them sold [11]. Surveillance can be done by microphones, as in cars [11], or video, which can be used for cyber-stalking, distribution of elicit material, or selection of victims for predators [22]. Adversaries can use location information for tracking [21]. Surgery robots could harm patients by repeating actions under a packet-dropping attack [25].

While examining security holes in a police cruiser’s on-board networking system, researchers could access the live video feed from inside the cruiser and upload, download, or delete video footage from the car’s onboard computer [33]. Thus, the validity of any video obtained by police forces in car cameras could be brought into question or even faked to implicate the police officers. Additionally, officers could be tracked and targeted using their vehicle’s video stream. Another example is the teleoperated MQ-9 Predator, used widely in Iraq for surveillance and strike missions and which transmitted unencrypted video that was intercepted by the enemy who may have used this information to defeat surprise attacks and to better understand US surveillance tactics and schedules [34].

5. Robotic System Security Solutions

In this section security principles, practices, and solutions are presented.

5.1 Designing for Security

Robots should be secure by design. Security is an issue which must be addressed in the design process early and often [12], [25], [35], [36]. Any component in or connected to the robot must be secure by design [25], including any remote web interfaces as these are prime attack targets [1]. Designs need to be informed by practical feasibility [8]. This could involve restructuring low-level software to provide a more robotics-appropriate abstractions [37], or rethinking if a connection to the Internet is really necessary [38]. A greater engagement between the robotics and security communities is necessary [11]. Additionally, providing proper legal protections for robotic developers requires security awareness by lawmakers [29].

The following questions should be asked during robotic design [1], [7], [11], [12], [14], [39], [40]: What are the robot’s intended function, users, and environment? Are there any unintended impacts on its users or environment? How mobile is it? What actuators and sensors does it possess? How will it communicate and be powered? What communication protocols are supported? What software and hardware modules are combined in this robot? How
will they securely interact? What type of databases or service, if any, will the robot be connected to? Should users always be in the control loop? How and where will necessary data be stored? What people (and animals) will be in its environment? How will users authenticate themselves to it? How will users information and data be kept private? What kind of development processes are in place? What type of market share is the robot expected to obtain? Will it be an attractive target to thieves or hackers? Does the robot create new or amplify existing privacy, physical integrity, physical safety, or psychological vulnerabilities? Can it be combined with other robots or technologies to facilitate an attack? How can we mitigate the chances for it being used in any type of an attack?

5.2 Secure Hardware and Software Design

Tried and true, standard, traditional computer security practices must be followed [11], [33]. This includes using established means such as distributed programming tools, multi-agent system architectures [12], and architectural security features [23]. Other suggested practices include using authentication for access control to the system combined with logs for accountability [12]; detecting conflicting or inconsistent information within the software, and alerting the user [21]; using a micro-kernel, which means including only the subset of the operating system and drivers necessary for the system [12]; setting up software and firmware to be updated with security patches and fixes, [8], [14], [22], [41], while preventing unauthorized upgrades or counterfeiting [7], [8]. Robot systems development teams should include people who are well-versed in standard security practices who contribute on all software and hardware design decisions.

5.3 Detecting Security Breaches

Systems can detect security breaches after they have occurred. Examples include: intrusion detection systems (IDS) [7], [23], [42], which can be specification or anomaly based; learning and detecting physical attack symptoms, like inexplicable halting or response delay [28]; systems for detecting and rectifying GPS spoofing attacks [43], [44], [45]; and honeypots, which has an easily hackable non-critical portion of the system that is monitored for intrusion [23]. These systems must also handle intrusion alerts. To investigate security violations, pertinent message and data information must be logged [46].

5.4 Secure Sensor Processing

Robots can deal with potentially corrupted sensor input or keep sensor information private. Attack-resilient state estimation and control algorithms are being investigated [7], [15], [20], which might prove fruitful especially as they are expanded across the many ways sensor data is used. They do, however, require redundancy, time-stamping, or data-coding. One approach strives to prevent sensor information leakage by embedding a privacy protection layer in the trusted low-level sensor processing library, e.g. OpenCV [47]. Untrusted application programs only have handles for the images, never access to raw image features. The trusted library performs necessary image operations, returning the final result or decision. If unsupported processing is required, the image is first transformed by a program or user determined amount, keeping sensitive information private. One example transform only keep edges in the image, yielding only a cartoon-like sketch. Another approach also involves the underlying image processing library, which now looks for specific physical markers instructing it to remove or obscure certain areas in the image [48]. This form of augmented reality requires careful marking of sensitive information in the environment.

5.5 Secure Communication

The literature is replete with suggestions to improve communication vulnerabilities. The first is to simply use standard encryption and cryptographic algorithms and follow standard best security practices for networked communication [7], [11], [22], [23], including Message Authentication Codes (MACs) [23], data packet formats with built-in security information such as time-stamped packets to avoid replay attacks [21], [25], avoidance of adhoc mode [22], transmitted data encryption [49], and restricting access to the robot’s network to desired entities [11]. The robot’s operation environment can be made more secure by requiring default username and password changing immediately after first log-in, not broadcasting the WiFi SSID, enabling network firewalls, allowing connections only from a whitelist, logging all interaction data, or requiring robots to be set up using a wired network connection [22]. DoS attacks can be resisted using jamming resistant modulations [12], malicious activity detection, or network traffic monitoring [27].

A positive example, which should be emulated in robotics, is the Secure Interoperable Telesurgical Protocol (ITP) [50]. It was designed for teleoperated surgery robot communication, which is of critical importance [29], and uses a multifold protocol which includes encryption and authentication schemes [50].

5.6 Worst Case Assumptions

All software should be designed assuming outer layers will be breached and rendered insecure [11], [12]. Redundant, layered security must be breached at each level, instead of having a single security breach yield complete control of the robot [12]. Designers should never assume the robot is too mundane, unimportant, obscure, or arcane to be attacked. For example, US officials knew about the unencrypted video transmission mentioned in Section 4.2 since the 1990s, but assumed adversaries wouldn’t know how to exploit it [34]. A system’s compromised utility may not be apparent to its designers, but always exists [1].

Robots should be designed assuming security measures will be defeated. This means not only that security should be implemented, but that a way to respond to breaches exists. This means different things for different systems, but the following questions provide guidance: Can manual,
legitimate control be reasserted? Can a kill-switch be engaged? Can the robot be removed from the network? For swarm robots, is the system (or the individual devices which make up the swarm) robust to preventing the spread of attack from one device to another [6]? Can users review what occurred to understand the attack and preventing more? Any logging to accomplish this would have to be done securely as well.

5.7 User Security and Understanding

Robot users should adopt good security habits. Strong passwords should be chosen and connected computers protected [22]. Access should be restricted [11] and proper configuration defaults utilized that give basic security levels [14]. For young users, parental approval should be required for camera, voice-over-IP connectivity, or other robot functionality [22]. Warning labels about the potential attacks and hazards to their privacy and well-being could be provided [22]. Users should be security skeptics and never assume a robot is 100% secure. Designers should seek to understand a typical user’s robotic attitudes, understanding, and knowledge, which have occasionally been ignored [1]. Future smart homes, which may include service robots [51], have similar challenges: surprisingly capable systems are being integrated into homes piecemeal without thought as to the level of domain knowledge of the homeowner [52]. As utility robots become more advanced a wider range of personnel will oversee or maintain them. Designers cannot assume robot operators will know best security practices, or comprehend the threat a poorly administered robot represents to their privacy [1], [30]. Basic security should be highly autonomous, with defaults that err on the side of being too secure. Clear and concise information about security settings should be provided, along with serious warnings when the system is in an insecure state [1].

6. Conclusions

Security in robotic systems will only grow in importance as robots become more common, are tasked with more important missions, and are granted more autonomy. Security cannot rely on vulnerabilities being too obscure, nor should it rely on an adversary being too incapable, too inept or too uncommitted. Instead, security must be folded into the entire design process. For robots to be truly useful, trustworthy tools, reasonable confidence must be had not only in people’s physical safety in their presence but also in the secure operation of their systems. By familiarizing themselves with potential robot security vulnerabilities, attacks, and solutions, robot designers will be better equipped to design robots that are more secure and safe for all users.

References


Biographies

Christopher Archibald is Assistant Professor of Computer Science at Mississippi State University. He received his Ph.D. in Computer Science from Stanford University and was a postdoctoral fellow at the University of Alberta. His research interests include many areas of AI, including Strategic Reasoning, Game Theory, Game Playing, Robotics, and Multi-Agent Systems.

Luke Schwalm is a graduate student studying Computer Science at Mississippi State University. His primary research focus is in security.
John E. Ball is Assistant Professor of Electrical and Computer Engineering at Mississippi State University. He earned the B.S. and Ph.D. degrees in Electrical Engineering from Mississippi State University, and a M.S. degree in Electrical Engineering from Georgia Institute of Technology. His research interests include signal and image processing, radar, and remote sensing, scene understanding, sensor data and image analysis, radar signal processing, and machine learning. He is a senior member of Institute of Electrical and Electronics Engineers (IEEE) and a member of the American Society of Engineering Education (ASEE). Dr. Ball has 16+ years of industry and government engineering experience.