SECURITY CONSIDERATIONS FOR AUTONOMOUS ROBOTS

Douglas W. Gage

Autonomous Systems Branch, Code 442
Naval Ocean Systems Center
San Diego, CA 92152

ABSTRACT

The security aspects of autonomous robots are analyzed by modeling a robot as a set of sensors, effectors, optional communications resources, and processing elements whose behavior is tightly coupled to the sensed characteristics of its environment. A simple taxonomy of potential generic threat types is presented, comprising both the possible direct external threat paths and the derived consequent internal threat states. Several generic countermeasure strategies are proposed.

INTRODUCTION

The past decade has seen the development of a well established technology of computer security. Models have been devised which define what it means for a computer to be "secure" [1], and system characteristics that provide both security of operation and assurance of that security have been developed. An analogous thrust is now beginning for the area of network security. This paper is intended to initiate consideration of security requirements and strategies for a class of systems which have not yet even come into widespread use: autonomous robots.

Recent developments in a number of hardware and software technology areas (sensors, processors, knowledge based programming techniques, complex system control) will soon allow feasibility demonstrations of autonomous robots. These devices promise to eventually find application in a variety of roles that are difficult or dangerous for humans, such as combat. Before such devices can be deployed in a battlefield role, however, a number of technical issues must be resolved beyond those involved in the feasibility demonstration itself: e.g., reliability, communications and coordination, doctrine for use, and, perhaps most critically, security [2].

It is difficult to talk about autonomous robots in the same way that computer security researchers customarily use to talk about computers. An autonomous robot does not contain data which is to be made available to authorized "users". In fact, it does not even have "users" in the usually understood sense. An autonomous robot can perhaps best be though of as a behaving machine. The robot consists of sensors, effectors, and internal computing resources. But there is (or at least there should be) no direct access to the internals of the computing resources. The robot's raison d'etre is to exhibit the behavior appropriate to its current sensor inputs, its mission, and the history of its interactions with its environment. The principal goal of the enemy's attacks on the robot is to prevent the robot from performing the appropriate behavior, either by disabling effector capabilities or by somehow causing the robot to choose to perform "incorrect" behavior.

While it may seem premature to examine the security characteristics of a class of systems which arguably do not yet exist at all, much less find widespread use, it is critical applications in which security plays a major role (such as combat or sentry functions) that will most likely pay for the development of autonomous robots, and their deployment in such applications will not be possible until security concerns have been satisfactorily addressed.

The paper begins with a brief review of security technology for computers and networks, then proceeds with the development of a simple model for an autonomous robot. The robot model leads directly to an enumeration of generic security threats to the robot, both the direct attack paths and the resultant derived internal consequences. Finally, several countermeasure strategies are proposed to maximize the enemy's cost in mounting these attacks.

COMPUTER SECURITY

The Department of Defense's concern with the issue of computer security can be traced from the formation of a task force in 1967, through the sponsorship of the KSOS and SCOMP projects to develop secure operating systems in the late 1970s, the launching of the DoD Computer Security Initiative in 1977, and the formation of the DoD Computer Security Center in January 1981.

Computer security is concerned with three broad classes of threats: violations of secrecy (data being read by the wrong person), violations of integrity (data being written by the wrong person), and denial of service (the right person being unable to access data). The DoD Trusted Computer System Evaluation Criteria [3] describes the goal of a "secure" computer system as follows:

"In general, secure systems will control, through use of specific security features, access to information such that only properly authorized individuals, or processes operating on their behalf, will have access to read, write, create, or delete information. Six fundamental requirements are derived from this basic statement of objective: four deal with what needs to be provided to control access to information, and two deal with how one can obtain credible assurances that this is accomplished in a trusted computer system... (1) there must be an explicit and well-defined security policy enforced by the system... (2) access control labels must be associated with objects... (3) individual subjects must be identified... (4) audit information must be selectively kept and protected so that actions affecting security can be traced to the responsible party... (5) the computer system must contain hardware/software mechanisms that can be independently evaluated to provide sufficient assurance that the system enforces requirements 1 through 4 above... and (6) the trusted mechanisms that enforce these basic requirements must be continuously protected against tampering and/or unauthorized changes..."

The formalization of these requirements reflects the development over the past decade of a number of techniques for constructing secure computers (kernels, capabilities) and methodologies for verifying the security of computer systems (specification languages, formal verification tools).

Computer security researchers have now begun to address the task of developing secure computer networks [4],[5]. The complication introduced is that the users and data associated with each machine must be protected against those of other machines and against intruders on the network. The communications component introduces additional channels for compromise, but communications encryption provides a powerful tool to guard against conventional passive and active tapping threats. Since the communications elements themselves do not introduce serious problems, the chief difficulties come from the interaction of multiple computers. Fundamentally, however, building a secure computer network is fairly analogous to that of building a secure computer.

ROBOT SECURITY

While the security concepts and mechanisms developed for computers are fairly applicable to computer networks, robots present a new and greater challenge. The behavior of a robotic system is more complex in that it is determined by many more variables. It is not always possible for an external observer -- even the designer of a robot -- to have confidence that he understands the detailed reasons for each element of the robot's behavior in terms of its mission and environment. Indeed, the introduction of mechanisms to maintain robot security will make such a detailed interpretation of behavior more difficult, as will be seen below. Thus it may be difficult to detect an enemy's attack on a robot, and may be even more difficult to assess the attack's success. The purpose of this paper is to stimulate consideration of this application domain by the computer security community, so that the need for security will not be the pacing factor in the deployment of robotic systems in such high risk applications as the battlefield.

A MODEL FOR AN AUTONOMOUS ROBOT

An autonomous robot is a device which possesses sensor(s) and effector(s) and uses these resources to act purposefully, autonomously, and continuously. To restrict our consideration in this paper to systems with non-trivial processing component, we use the following model, adapted from reference [6]:

(1) A robot is a device which consists of sensors and effectors coupled by processing capabilities, and (optionally) communications elements that provide communications with other robots and with controllers.

(2) The bandwidth between a robot's sensors, processing elements, and effectors is much greater than the available communications bandwidth.

(3) The output of (at least a subset of) the robot's sensors is a function of the external environment (i.e., the sensors sense the environment).

(4) A subset of the environment is a function of (at least a subset of) the robot's effectors (i.e., the effectors affect the environment).

(5) The output of (a subset of) the robot's sensors is a function of the robot's effectors' actions (the robot can sense the effects of its effector actions).

(6) The bandwidth between a single robot's sensors and effectors through the environment is much greater that the available communications bandwidth.

(7) Each robot maintains an internal representation of the relevant characteristics of the environment.

(8) Each robot maintains a goal structure or representation of the desired state of the task environment based on its assigned mission.

(9) Communicating robots share some commonalities in their respective world views (representations of the environment and goal structures).

This generic model is presented here to distinguish the class of autonomous robots under consideration from other superficially similar devices (such as teleoperators) sometimes referred to as robots, and to serve as the basis for the development of the taxonomy of threats presented below.

DIRECT AND DERIVED THREATS TO AUTONOMOUS ROBOTS

Because the behavior of a robot is essentially "calculated on the fly" by the system's processing resources as a function of its current memory state and inputs from its sensor and communications resources, it is useful to distinguish between the immediate intended effects goals of the enemy's actions (direct threats) and the intended resultant changes to the robot's internal state which would lead to incorrect behavior (derived threats). For example, two direct threats that the enemy could use against a weapon-carrying robot deployed in combat might be (1) sending false commands to it, and (2) disguising its own forces so that the robot believes they are friendly. In the first case the derived threat is one of changing the robot's high level goal structure, while in the second the attempt is to pervert the robot's world model. A direct threat is thus closely coupled to the mechanism by which the robot and the enemy interact, and a single direct threat can cause any of a number of derived threats, identified in the next section.

Sensor Mediated Direct Threats

(1) Detection avoidance: the enemy attempts to manipulate inputs to a sensor so that it produces no indication of the enemy's presence.

(2) Jamming: the enemy attempts to manipulate inputs to a sensor so that it is prevented from providing its intended information (often characterizing the nature or position of the enemy forces). The system detects that the sensor is not providing the desired information, but does not necessarily ascribe this to enemy action. The sensor functions properly when jamming is discontinued.

(3) Spoofing: the enemy attempts to manipulate inputs to a sensor so that it produces plausible but inaccurate information. The system does not detect that the information provided is incorrect, and therefore bases decisions on it.

(4) Disabling: the enemy attempts to manipulate inputs to a sensor so that its capabilities are impaired to the extent that it can not function accurately even when the disruptive force is discontinued.

(5) Feedback: the enemy attempts to detect and interpret any behavioral cues which a sensor might produce as a byproduct of its operation, for example to determine whether its presence has been detected.

(6) Characterization: the enemy attempts to understand the capabilities of the sensor so that it can predict whether the sensor will detect it in any given situation. This allows the enemy to take countermeasures .

Effector Mediated Direct Threats

(1) Disabling: the enemy performs some act that reduces or destroys the effector's capability, and the effector is disabled until it is repaired.

(2) Paralyzing: the enemy performs some act that reduces or destroys the effector's capability, but the capability is restored as soon as the enemy discontinues the act.

(3) Characterization: the enemy attempts to understand the capabilities of the effector so that it can predict what effect the effector can have in any given situation. This allows the enemy to take countermeasures.

Communications Mediated Direct Threats

(1) Detection: the enemy attempts to detect the occurance of a communications act (transmission or reception).

(2) Jamming: the enemy attempts to prevent a communications act from succeeding. The system detects that the communications subsystem is not providing the desired information, but doesn't necessarily ascribe this to enemy action. The communications subsystem functions properly when jamming is discontinued.

(3) Spoofing: the enemy transmits data which the receiver believes was transmitted by a friendly force. It may be data wholely of the enemy's composition, altered data originally generated by a friendly, or transmission of an unaltered friendly message in a context of the enemy's choosing (e.g., playback). The enemy may not even know what the message contains.

(4) Disabling: the enemy performs an act so that the communications subsystem's capabilities are impaired so that it does not function accurately even when the disruptive force is discontinued.

(5) Characterization: the enemy attempts to understand the capabilities of the communications subsystem so that it can take countermeasures

(6) Interception: the enemy attempts to intercept the communications and may or may not be able to decrypt it.

(7) Traffic analysis: the enemy attempts to gain information concerning our capabilities and/or intentions by analyzing the patterns of our communications traffic (without having to understand the content of the communications).

Processing Element Mediated Direct Threats

Barring a physical penetration of the device, the processing elements of an autonomous robot have no direct interaction with outside forces during a mission; all interaction is mediated by the sensor, effector, and communications elements discussed above. A direct attack is possible before the mission begins, however, through the implantation of a Trojan horse in the software. Because the behavior of an autonomous robot can be more complex than that of "conventional" systems (at least in the sense of having its internal states be less comprehensible to an external observer), a carefully implemented Trojan horse would be detectable only through very careful analysis of the "infected" implementation at multiple levels of abstraction.

Derived Threats

If the processing element of a robot is modeled as an engine which operates on internal representations of (1) the external environment and (2) a mission derived goal structure, then the state of the system is essentially embodied in these two data structures. Attacks can be made on the secrecy or integrity of these structures as follows:

(1) The enemy can seek to deliberately change the robot's goal structure.

(2) The enemy can seek to deliberately change the robot's model of its environment.

(3) The enemy can seek to learn what the robot's model of its environment contains.

(4) The enemy can seek to learn what the robot's goal structure contains.

The direct threat path mechanisms described above can be used by the enemy to achieve any of these indirect derived goals.

Of these derived threats, the second one, model spoofing, appears to require the most research. The fusion of data from multiple sensors into a coherent world model that can support the performance of a complex mission is a process that humans can do much better than machines (at least machines using currently available technologies). A determined enemy will always be able to introduce some feature into the robot's environment that does not easily fit into the robot's predetermined model of the anticipated task domain. The robot must be able to use its own resources, and the other robotic and human resources that it can call upon via its communications, to determine how to deal with each "surprising" situation it encounters.

COUNTERMEASURE STRATEGIES

The basic strategy for protecting the secrecy and integrity of the robot's internal state is to make it too expensive for the enemy to effectively attack the robot system. This can be done by maximizing the breadth of the environmental data upon which the robot's behavior is based and minimizing the information conveyed to the enemy by observation of the robot's behavior. The robot should make use of as many independent sources of information about its environment (and especially about the enemy) as is possible, while at the same time affording the enemy the least opportunity to understand its actions, capabilities, and intentions.

Use of a diversity of sensor types makes it much more difficult for the enemy to spoof or disable the robot's sensing capability with a single countermeasure, or to avoid detection. Correlating inputs from multiple "orthogonal" sensor types employed to measure the same features of the environment (such as using both an imaging and a range finding sensor to identify objects in the neighborhood) provides a means to detect spoofing attacks. Spatially distributing sensors by using multiple coordinated robots affords additional powerful verification capabilities.

One simple technique that can be used to minimize feedback of information to the enemy is that of randomizing the robot's behavior. By obscuring the dependency of the robot's actions on the sensed characteristics of the environment, this can force the enemy to plan a costly margin of safety around the robot's actual capabilities.

ISSUE AREAS

One key issue that should be addressed quickly is deciding the degree to which secrecy of mechanism should be utilized to provide security for autonomous robots. What are the risks implicit in such a strategy? What are the alternatives? Can a robot be made robust enough to counter a sophisticated spoofing effort mounted with exact knowledge of sensor capabilities and processing algorithms? If it is decided to employ secrecy of mechanism, at what stage should the development effort be classified? At what level? What are the implications for the robotic development community?

Unlike an autonomous robot, a computer system does not typically maintain an explicit world model that might include its enemies (except in some auditing functions). Must the fact that there is an "enemy" trying to spoof the robot's sensors be made an explicit part of the robot's model of the world? How difficult will it be to model the more sophisticated spoofing threats? Or can simple strategies provide adequate security for most applications of interest? In what sense does a "security policy" substitute for an explicit model of the threat?

As discussed above, the model/sensor spoofing threats appear to be the most interesting (and most difficult) to deal with. It will be necessary to establish objective measures of system performance in this area and to weigh them against the requirements of specific applications.

The discussion presented here has merely scratched the surface of an unexplored area that will require a great deal of research. It is not yet clear how current computer security approaches can play an important role in robot security. Many of the issues involved are also encountered in communications security and electronic warfare.

CONCLUSIONS

Autonomous robots represent a new class of devices whose security requirements appear to differ greatly from those of conventional computer and computer networking systems. As feasibility demonstrations of autonomous robots are achieved in the next few years, a push will begin to develop such systems for deployment in real world applications. This will require a satisfactory treatment of the security issue. It's not too soon to start now.

ACKNOWLEDGEMENT

The author wishes to express his appreciation to S.Y. Harmon for numerous stimulating discussions and for his useful critical reading of the draft manuscript.

REFERENCES

1. D.E. Bell and L.J. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation, MTR-2997 Rev 1, MITRE Corp., March 1976.

2. S.Y. Harmon and D.W. Gage, Current Technical Research Issues of Autonomous Robots Employed in Combat, Proceedings of the 17th Electronics and Aerospace Conference (EASCON 84), Washington DC, 10-12 September 1984.

3. DoD Computer Security Center, Department of Defense Trusted Computer System Evaluation Criteria, CSC-STD-001-83, 15 August 1983.

4. S. Walker, Introduction to Network Security Evaluation, Proceedings of the 7th DoD/NBS Computer Security Initiative Conference, Gaithersburg, MD, 24-26 September 1984.

5. Defense Intelligence Agency, DODIIS Network Security Architecture and DNSIX, Final Draft, 1 August 1984.

6. S.Y. Harmon and D.W. Gage, Protocols for Robot Communications: Transport and Content Layers, Proceedings of the 1980 International Conference on Cybernetics and Society, Cambridge, MA, 8 October 1980.