Principles of Skeptical Systems

Steve A. Harp and Christopher W. Geib

Honeywell Labs
3660 Technology Drive.
Minneapolis, Minnesota 55418
steve.a.harp{christopher.geib}@honeywell.com

Abstract
The high degree of autonomy that is being built into today’s control systems, coupled with their vulnerability to attack at the computer network level, mandates a recasting of the man-machine social contract. We now have the technology and the capability to enable these control systems to entertain “a healthy skepticism” about the validity of the commands being given then and the identity of the issuers.

Introduction
The line of research described herein arose from considering the problem of securing network-connected control systems. Automated systems now play a large role in controlling chemical plants, refineries, buildings and aircraft. As these systems have grown in sophistication, they have become more capable of autonomous action, performing complex tasks on the operator’s behalf, and capable of running for long periods of time with little assistance from the operator.

The owners of such systems want to communicate with them remotely and to integrate them with other systems in their enterprise, or those of their partners. It is frequently not possible to do this well with only private networks—the cost to develop and maintain them are prohibitive. However, the use of public (internet) infrastructure exposes hazards: involuntarily sharing of proprietary information with the world, the tampering of pranksters, or worse, saboteurs and terrorists.

These autonomous systems are very valuable assets. The health and livelihood of many people depend on them. What can be done to reduce the risks entailed by connectivity to an acceptable level? Happily, many of the hazards can be averted through the enforcement of a sound security policy and the use of defensive tools such as firewalls, cryptography, network management, vulnerability checking, and intrusion detection software. Notably, virtual private networking technology allows secure communications to tunnel through a public network. When implemented correctly, it gives very strong protection from unauthorized snooping and tampering, thus apparently solving the motivating problem of secure connectivity.

Sound security policy and implementation cannot be purchased in shrink-wrapped form. However, the general principles are widely known, and reasonable tools and services are available to help implement it. The primary impediments are the cost and the cultural willingness of the organization to maintain a secure posture in the face of ever changing threats.

So, is the problem is solved? Unfortunately, serious issues remain, even with these measures. First we mention some reservations about existing defenses. The remainder of this paper will discuss the deeper problems that remain and propose a philosophical shift in the way we think about autonomous systems for computer control as a path to a solution.

Conventional Defenses: Good, but Imperfect
Like all other software, security software—cryptographic products, firewalls, intrusion detection systems and the like—is not always correctly designed and implemented by the vendor, nor is it always (even often) correctly configured and operated by the customer. Most firewall products can claim a history blemished by a vulnerability or two. Worse, they are notoriously difficult to configure and validate. This is not to belittle the value of these tools. Without them, we would be defenseless. Rather, the point is that they have flaws that motivated intelligent adversaries will discover and exploit.

But this source of problems pales in comparison to the financial and cultural barriers to maintaining a strong security policy. Because security is perceived as a cost (time and money) with no apparent benefit, it will tend to be starved by managers attempting to optimize more tangible benefits. Risks associated with a lax security posture are not readily perceived unless an organization has had recent bad experiences; even then, organizational memory can be depressingly short.

Further, well-intentioned measures for security can also be incompetently implemented. In these cases, enforcing an employee-unfriendly approach to security policy can be worse than poor enforcement of a good policy.
Living with Imperfect Defenses
If we have bought the best defensive software and nurtured a healthy security culture, can we now rest peacefully? At least three things should still worry us:
- Insiders—people who have been granted authorization—can still largely do as they please.
- Outsiders can slip through the holes in the defenses and gain authorization.
- The list of holes is large and growing. [CVE, Bugtraq]

The threat from inside is particularly troublesome. Bruce Schneier (2000) quotes an unnamed NSA source as stating “Someone whom you know can violate your security policy without getting caught.” Reports by Ernst & Young and The Aberdeen Group estimate that perhaps 75% of security breeches come from authorized users. The average cost of an authorized user attack, according to the FBI, is $2.7 million.

In a law enforcement panel discussion at a well-known Las Vegas hacker convention in August 2001, a panel member informed the audience that they were by no means his biggest problem. Rather, he averred, the truly dangerous threat came from insiders like FBI counterespionage agent Robert Hanssen, who allegedly sold secrets including names of double agents and U.S. electronic surveillance methods, revelations that severely damaged national security.

That holes (vulnerabilities) exist and that new ones are continuously being discovered is evidenced by the growth of vulnerability databases such as CVE and Bugtraq. Two industry trends have amplified the problem. The first is the increased reliance of commercial-off-the-shelf (COTS) software packages. These products typically place security as a distinctly secondary goal behind power and convenience features. A related trend towards software monoculture has not helped either. While it is easier to manage training and installation when all of the nodes are identically configured, it is a risk amplifier. If one node in the system is susceptible to some vulnerability, nearly all of them will be. The success of viruses and worms such as Melissa, NIMDA, CodeRed etc in bringing corporate networks to a standstill is a recurring demonstration of this weakness.

Recasting the Man-Machine Social Contract
For some critical systems, a further layer of security is warranted. To address this, we propose the development of Skeptical Systems. Current systems promptly obey any commands issued by the last authenticated operator so long as they fall within the privileges granted. By contrast, a skeptical system entertains doubts about the tasks that it is asked to perform. It questions the authenticity, integrity, and intent of the requester and acts with due consideration of its doubts.

The skeptical system is not a new technology per se; rather, it is a nascent philosophy of man-machine interaction that grants considerable autonomy to the machine under certain conditions. Nor is the skeptical system an entirely new idea. Some systems already exist that embody this stance, at least in limited ways. More ambitious conceptions of skeptical systems have populated fiction for some decades. We will touch upon some of the precedents later, but first will sketch out some of the precepts of skeptical systems as they apply to critical assets, and then consider a notional example.

Precepts of Skeptical Systems
The following set of design principles is meant to define a specific sort of autonomy that would prove useful in safeguarding control of important assets.

P1. The skeptical system shall monitor continuously the authenticity of the commands it receives. This means asking whether the issuer of a command is really the entity it claims to be. The word ‘continuously’ is important, in this precept and others; see notes.

P2. The skeptical system shall continuously monitor the integrity of the sources issuing commands. The command issuer may be a human user or another system.

P3. The skeptical system shall continuously evaluate the likely consequences of the commands it receives. This implies two activities: envisionment of consequences, and assessment of the losses should those consequences occur. The system must consider all command streams as a whole, avoiding the possibility of coordinated attacks from multiple users.

P4. The skeptical system shall continuously evaluate the probable intentions behind the commands it receives. Intent recognition is difficult to achieve in general, however certain normal and abnormal patterns of commands can strongly indicate either benign or malign intent. This precept is closely allied with the evaluation of possible consequences (P3).

P5. The skeptical system shall actively intervene only in high stakes cases. Stakes may be defined in terms of safety, significant economic losses, and/or environmental jeopardy as appropriate. Failure to follow this precept will tend, at the very least, to annoy operational personnel.

P6. The skeptical system’s response to user commands shall be conditioned by its best estimate of authenticity, integrity, consequences and intent in a graded manner. There is a spectrum of possible responses possible, from instant compliance to counter action.

P7. The skeptical system shall conceal its suspicions from doubted agents where revelation runs a risk of being exploited. Providing too much feedback will be both a distraction and allow the system to be more easily fooled.
P8. **The skeptical system must be suitably tamperproof.** There should be no way for an attacker to subvert or sidestep the skeptical mechanisms. No artifact is impervious to tampering, but the threshold for attacks of this sort must be set high enough to make it very unattractive.

P9. **The skeptical system’s authority must have a limit.** Some person or group of persons must ultimately be able to trump the system’s judgements. The limit should be set in reference to a worst-case analysis.

P10. **The authorized users of a skeptical system must know of its autonomy, its limits, and the protocol to override it.** This information need not be widely shared, but its disclosure should not diminish its security.

Several of these precepts demand continuous evaluation by the skeptical system. A defect of many current systems is that authentication happens just once—at the beginning of a session. Hijacking such a session (in ways discussed further below) defeats the authentication. Ways to do continuous authentication without irritating users will be discussed.

A key difference between skeptical systems as defined by the precepts and previous attempts to build safety interlocks into controls is the subtlety of the interaction between user and system. It must deal with world states that are shrouded in uncertainty. Perfect knowledge of the intentions, integrity or even authenticity of a user is seldom possible. The skeptical response to uncertainty is hedging (P6). Instead of a lock that is either fully open or closed, the skeptical system has gradations of compliance, dynamically varied in proportion to its trust in the command issuer and the consistency of the command stream with the system’s mission.

**Example**

Consider a chemical plant safeguarded with a skeptical system. The skeptical system stands between all external commands and the control system that regulates the plant. Chief operator Dave interacts with the process via the normal interface.

As Dave presses buttons, the skeptical system asks itself the following sorts of questions about the interactions between operator and control system:

- What are the possible implications of the current command sequence with respect to plant safety, production, environment, and property?
- Is the issuer of this command really Dave? Or is Dave tied up in the broom closet and someone else is pressing the buttons?!
- Is Dave in his right mind today? Or is he behaving erratically? Are there others with him?

Some of these questions may be dismissed easily. If the system’s camera watching the console chair sees someone matching Dave’s official photograph, and if the typing pattern from the keyboard matches the one learned for Dave, some confidence is built in the authenticity of the command issuer. However, positive answers to these skeptical questions would lead to an elevated level of suspicion. A skeptical system would respond in a graded manner:

- **Mildest response:** Logging the suspicion for later analysis. “For a moment this morning, I didn’t recognize Dave.”
- **Asking Dave whether the perceived consequences are well understood before proceeding.** “Closing valve FC101 that far might jeopardize reactor cooling, Dave, do you really want to proceed?”
- **Demanding a greater level of authentication, e.g. probing for a password, personal information, asking for a physical key device to be produced, or biometric confirmation.**
- **Notifying others (operators, managers, engineers) of its doubts:** "I’m curious why Dave just disabled the auxiliary cooling water supply to unit 2..."  
- **Enabling “go-slow” restriction to prevent catastrophic changes to the controlled plant, e.g. making a setpoint move slowly over 45 minutes instead of 45 seconds, or injecting a delay of 30 minutes.**
- **Demanding confirmation from others before proceeding.** “Dr. Martin, Is it really okay to disable the auxiliary cooling water supply to unit 2 now, as Dave has suggested?”
- **Strongest response:** Refusing to proceed, locking out the console until a trusted manager can be authenticated, and taking automated steps to “park” the plant in a safe operating region. “I’m sorry, I can’t do that Dave.”

**Situations Warranting Skepticism**

Skeptical systems are not appropriate for every application. A skeptical Coke machine would have customer acceptance problems. When would wisdom direct me to permit the skeptical system to disobey, or to exercise autonomous control? Here is a partial taxonomy of (first person) answers:

**Limited rationality.** “I’ve forgotten some detail or was unaware of a special condition, and hence inadvertently asked it to do something very costly or dangerous.” Shift changes in process plants are notorious sources of problems like this. E.g. a special action must be taken during shift 1 to deal with a problem; it is initiated but not completed. Shift 2 arrives, but is not fully briefed and proceeds as though all was normal.

**Clumsy fingers.** Data entry devices and fingers are not 100% reliable, allowing incorrect commands (or more often their parameters) to enter the system. E.g. entering 027º instead of 270º into an inertial navigation system. A 1983 Korean Airlines aircraft that strayed into Russian airspace may have made a mistake like this.

**Medical conditions.** Biological problems can go much deeper than clumsy fingers. “I cannot effectively control
since: I am mentally impaired—drugged, sleep-deprived, febrile, etc, I am unconscious or asleep, I am having a seizure, or have died.”

Physical force. These are the actions of saboteurs, terrorists, hard-core criminals, and are rare but may be devastating. “I have lost control because have been: physically restrained, injured, tortured, threatened, or killed.”

Hacked. This is a basic mode for computer crackers, perhaps in the employ of serious villains. “My command channel has been breached to some extent, and an unauthorized agent is, with or without my knowledge preventing me from entering the correct commands, and/or injecting its own commands, perhaps spoofing my authorization.”

Tricked. “I have been given false information by some external agent, with or without malevolent intent, and now I am acting with good intentions on a dangerously erroneous model of the world.” Examples include: Social engineering. Exploiting natural cooperativeness, existing social conventions to extract some gain or privilege, or cults—the total immersion form of social engineering. Rumors, urban legends—highly virulent but inaccurate communications; these need not be maliciously formulated to have dangerous consequences.

Areas of Applicability

Several domains have the inherent potential danger and complexity to warrant consideration of skeptical systems.

Process Control: Defeat sabotage, espionage, careless error. As the future of control systems places more and more intelligence in the field and not in the central supervisory system, it may be possible/desirable to push this skepticism into the field. The interdiction of hostile agents aside, this technology could also save us all from costly careless mistakes arising from sleepy or impaired users.

Transportation: Hijack prevention, accident prevention. A skeptical system in an airplane could detect malicious intent on the part of the person at the controls, using evidence such as deviation from flight plan, disabling transponders, and flying below 5000 feet.

Building Security: Prevent unauthorized or inappropriate defeat of environmental control and security.

Weapons Systems: Prevent unauthorized examination or use. The notion of intelligent weapons that will only shoot when directed by their authentic owner have been explored elsewhere. The skeptical weapons systems generalizes this idea.

Precedents

Simple Skeptical Systems

There are a number of existing machines that behave in a rudimentary skeptical fashion, under certain conditions.

• Interlocks on machinery, electrical equipment, and plumbing that prevent anyone from opening or operating the device while certain conditions persist, e.g. spinning, energized, pressurized. These are the simplest possible sorts of robots, yet have saved many lives and property.

• Safety systems on trains that prevent drivers from entering junctions that are not clear, or from taking curves at too high a speed.

• Dead-man switches on trains that demand periodic button presses to prove the attentiveness of the driver; failure to press on time will brake the train.

• Launch control for certain weapons demands agreement by multiple operators.

• Automatic Ground Collision Avoidance to automatically fly an aircraft away from the ground if the pilot became unconscious. See also experimental versions of Traffic Alert and Collision Avoidance System (TCAS).

• Warning dialogs on computers that offer the chance to continue or abort an operation that cannot be easily reversed, e.g. permanently erasing a file, choosing a new password demands confirmation.

Interestingly, while many of these systems are not well liked by the system operators that use them, they are at least grudgingly tolerated. Other people—such as the owners of airplanes and trains—have provided powerful reasons to respect these mechanisms. In the end, the happiness of the controlling professional runs a poor second to safety and/or broader socioeconomic goals in these situations. This give us good reason to believe that more advanced Skeptical Systems will be likewise accepted. We will discuss this in more detail later.

Computer Security Systems

Some interesting cases of skepticism have been applied to safeguarding software applications from hostile code—viruses, worms, Trojans etc. Conventional virus protection recognizes signatures of known malware. More interesting are defensive programs such as those pioneered by Crispin Cowan—StackGuard and FormatGuard—monitor for the exploitation of particular classes of vulnerabilities (Cowan et al., 1998, 2000). StackGuard detects when an application’s program stack is overwritten with executable instructions—the classic buffer overflow exploit. FormatGuard detects when format string exploits are exercised, using weaknesses in the “printf” family of system functions to inject malicious code. Some virus detection tools also operate on related general principles. In each case, the doubted application is immediately terminated and an incident is logged.

Still more interesting, artificial immune systems for programs attempt to do the same thing, but without resort to the first principles of software vulnerabilities. Examples are discussed by Hofmeyr & Forrest (2000). These systems learn the normal behavior of applications by monitoring such things as their sequences of system calls, resources consumed etc. When the application exhibit
behavior inconsistent with its past, an alert is issued. Many anomaly detection tools have a similar kernel idea.

None of these systems are fully integrated in the sense described in this paper. In most cases, they fail to provide a graded skeptical response: Their responses are either non-existent, or brutal. Some subtlety has been proposed however; Somayaji and Forrest (2000) suggest delaying system calls in computer applications suspected of having been compromised. Dean and Stubblefield (2001) suggest using a protocol that forces clients (applications) to solve "puzzles" under certain conditions to mitigate denial of service attacks.

Objections

The exposition of skeptical systems heretofore is probably sufficient to raise objections and criticisms in the minds of many readers. This section will anticipate several of these issues, and serve as apologia for skeptical systems. The objections we consider may be summarized:

[A] False alarms: What if I need to do something that looks suspicious but is actually the "right thing to do" and the skeptical system stops me? What if it’s also an emergency situation? And there are only seconds to spare?

[B] Misses: Such a system cannot work since it would be unable to adequately envision consequences of a stealthy attacker’s fiendishly subtle plan, leading to compromise. I know I could always fool it.

[C] Turf Control: I will not defer to a machine.

[D] Fear and Loathing: Thinking machines are not to be trusted—they have their own agendas...

The first two objections are two sides of the same coin: one trades off errors of type [A] against those of type [B]. In a perfect system, both error rates would be zero. But systems that are not perfect can still be very useful. Vehicle airbags have prevented many injuries and deaths in collisions; they have also sometimes deployed when they should not, causing injury; on rare occasions, they may fail to deploy when needed. Vaccines against infectious diseases have saved countless lives and made an enormous reduction in human suffering. However, they do not provide protection in every case, and can sometimes have serious, even fatal, side effects.

Where one sets the balance is a valid topic for analysis. Consider first how often will these types of errors occur, and what will be the expected consequences. Then, how does this balance against the utility purchased by the skeptical system via saving us from some expected number of incidents of (varying) severity? From the utilitarian standpoint, the skeptical technology should be deployed if it can be operated at a net expected gain. There are rational reasons to not deploy a system that cannot be made to operate at some acceptable tradeoff point. For example, a vaccine with side effects as frequent and severe as the illness should not be administered.

What level of benefit could be expected from skeptical systems built with existing technology? If deployed with care, they could be quite valuable. Many control platforms could readily benefit from skepticism built to hug the conservative corner of the error tradeoff. By providing gradation of response, they would reduce losses due to false alarms, while still providing a substantial line of defense in cases where more certainty is obtainable.

The second two objections are more overtly cultural and psychological. The issue of deference to a machine [C] might be discarded as being an indignant expression rooted either in hubris or perhaps in issues 1 or 2. However, an alternative interpretation must be considered. The desire for personal control can be due more to relations between people than to relations between people and machines. Executive control implies prestige—losing this to a machine well be felt as subtly humiliating to some. Positions with high prestige may raise the strongest forms of this objection. Yet there is evidence that people will tolerate at least some loss of control in situations that involve high stakes. All of us regularly encounter situations in which our choices are constrained by laws, which though not usually embedded in an automated system, are nonetheless usually respected.

Finally, fear of the machine [D] has roots in many cultures. Popular fiction has a rich cast of computers that have been entrusted with various powers more often sadly misused. Arguably the most famous is the homicidal HAL 9000 from Kubrick/Clark’s work 2001: A Space Odyssey (Clark 1968; see Stork, 1997, for a technical perspective). Anecdotally, several of the precepts of skeptical systems were violated by HAL, not the least of which being tamper resistance.

Technology Required for Skeptical Systems

Skeptical systems are not a technology per se. However, their successful implementation will rest on several technologies, now in varying states of development. The performance curve of skeptical systems will be driven most immediately by the quality of its sensors and models of threat. The challenging problems in these areas are likely to yield slowly to a combination of theoretical and empirical research, which should be nurtured soon.

Passive biometrics. Identify the command issuer on a frequent basis without interrupting workflow. Examples with promise include:

- Multispectral face recognition capable of seeing through disguises.
- Touch screens with fingerprint identification capability.
- Speaker identification through voice print technology.
- User-specific models of input-device operation such as keystroke timing, pressure, typical errors.
- Remote stress detection through acoustic, visual and infrared sensing.

Learned user identification and integrity checking through interaction monitoring: This would include learning operators’ individual interaction patterns, and the learning of standard operating procedures.
Intent recognition/task tracking: This would include logging the control actions of the user and using a library of known plans to infer the agent’s goals. The recognition of hostile plans would allow us to infer intent, and take steps to mitigate its effects.

Reasoning models for physical systems: The ability for an automated system to reason about how a protected asset will respond to user commands. High precision mathematical models are desirable, but not always practical. Techniques for approximate models that would enable a skeptical system to envision consequences are called for.

Contingency planning: Rapidly plan responses commensurate with the changing security situation. The plans of a skeptical system must hedge. They must build in slack to account for uncertainty, and weigh the worst case of an intelligent adversary attempting to counter them.

Conclusions

The high degree of autonomy that is being built into today’s control systems, coupled with their vulnerability to attack at the computer network level, mandates a recasting of the man-machine social contract. We now have the technology and the capability to enable these control systems to entertain “a healthy skepticism” about the validity of the commands being given them and the identity of the issuers. Further, the autonomy of these systems, and their ability to take graded responses to the instructions given them can be their own best defense against honest mistakes by users as well as hostile actions by authorized insiders and outside attackers.

Acknowledgments

This material is based on work supported by the Air Force Research Laboratory under contract No. F30602-02-C-0116.

References


