

Security Aspects of Cyber-Physical Device Safety in Assistive Environments

Steven J. Templeton
Computer Security Lab
Department of Computer Science
University of California, Davis
One Shields Ave.
Davis, CA 95616
templets@cs.ucdavis.edu

ABSTRACT

As more devices that affect their environment come into use, their proper functioning to protect the welfare of their charges is a concern. Examples include assistive transport devices, robotics, drug delivery systems, etc. Here privacy is not the primary concern, instead it is safety. Given that there are many instances of medical devices not being developed to be secure, plus the standard of practice for security with robotics and other cyber-physical devices, the issue needs consideration. These systems are not only vulnerable to intentional attack, but can cause harm inadvertently by unexpected interaction from other systems. This paper discusses security challenges of expanded use of cyber-physical devices in assistive environments and provides suggestions to improve the security and safety of these devices in the future.

Categories and Subject Descriptors

K.4.1[Computers and Society]: Public Policy Issues – Human Safety

K.6.5[Management of Computing and Information Systems]: Security and Protection – authentication, invasive software, physical security, unauthorized access

General Terms

Security, Human Factors, Standardization.

Keywords

Pervasive computing, assistive-environments, cyber-physical systems, security, safety, standards.

1. INTRODUCTION

Pervasive computing consists of two related classes of technology: (1) those systems primarily involved in data collection, management and analysis, and (2) those systems that respond to the environment or commands and cause changes to

the physical world. Implantable microcontroller based pacemakers and drug delivery systems, computer controlled vehicles, mobility devices, robotics and other assistive and decision support technologies are examples of current and coming technologies. These *cyber-physical systems* (CPS) offer substantial benefit to society. They can however be a source of physical harm when they malfunction or are used improperly.

When the security issues of pervasive computing are discussed, they generally focus on privacy. Although a concern, issues of safety are of greater importance. As use of these technologies expands and people come to rely on them, particularly children, the handicapped, and the growing aging population, the security and safety of these becomes an increasingly significant issue. Flaws in their controlling software or the protocols used for remote or intra-device communication can result in harmful failures or can be maliciously exploited to the detriment of those persons the devices were intended to help. Many examples exist in the control system and medical community.

In [17] researchers found that they could take control of an automobile's electronic control unit (ECU) and override safety critical functions. Many successful attacks were conducted including preventing the brakes from being applied, preventing an occupant from unlocking the doors, or selectively causing one wheel to brake, potentially causing the vehicle to roll. In one attack, the researchers used a modified music file to compromise a vehicle's audio system and infect other systems in the car.

A May 2010 report[24] stated that an estimated 650,000 people in the United States have either cardiac pacemakers or implantable cardioverter defibrillators (ICD). According to Medtronic, a major implantable medical device manufacturer, over 80,000 people have received implantable deep-brain stimulation (DBS) devices to treat movement disorders, depression and other psychiatric conditions[25][28]. Like ICDs these can be remotely reconfigured to meet the patient's needs. Implantable drug delivery systems are also common. Like the imbedded vehicle controllers, because these devices are remotely reconfigurable and are weakly secured, the risk of attacks targeting these devices and affecting their safety are real. In several studies cardiac devices were shown to be able to be remotely reprogrammed without authorization[11][12]. Researchers showed that a malicious person could cause the device to deliver a shock sufficient to induce ventricular fibrillation and potentially cause death.

The Global Positioning Satellite (GPS) system provides location and time data to devices capable of receiving the satellite's

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PETRA'11, May 25 - 27, 2011, Crete, Greece.

Copyright ©2011 ACM ISBN 978-1-4503-0772-7/11/05 ... \$10.00

signal. This is used for navigation; location tracking (e.g. vehicle anti-theft devices, transponders to verify a fishing boat's location to prevent commercial fishing in restricted areas); for sub-millisecond time synchronization of data collected from a large number of widely distributed sites; or to control autonomous vehicles. GPS signal strength is very weak at the Earth's surface making it relatively easy to broadcast a stronger signal to block the actual GPS data and substitute it with other chosen values[32]. This allows a car thief to have the anti-theft device report the vehicle a false location, or the fishing boat to be in restricted water, while broadcasting that it is operating elsewhere. GPS attacks would also make it difficult if not impossible for autonomous vehicles to navigate or emergency responders to find the location of an injured driver.

Fortunately in these examples, while possible, no reports of harm have occurred. However, in 2008 a 14 year old Polish boy was able to control the Lodz tram system using a reprogrammed television remote control[5]. He was able to operate track switches sending cars where he chose. Ultimately 4 trains were derailed and 12 persons injured. Sadly, in the case of the Therac-25 radiation treatment device which was found to have multiple software errors, 6 persons were severely injured, of which 4 died[19]. While this case was not an attack, it is an example of the harm that could be done by an attack. Any CPS incident, regardless of the initiating cause, must be considered as a potential attack – nearly always they could have been.

Clearly flaws in these systems can and has caused harm. These cases, while not pervasive systems, are illustrative of the types of issues that we can expect to see. Extrapolating them to the functions pervasive systems will provide will give us insight into the problems to expect and to work to prevent in the future.

Although many people have a strong and at times unfounded fear of new technology, the prevalent belief in the general population is that these cyber-physical devices are safe, that they were strictly engineered and tested to be safe, and passed evaluation by commercial or government organizations such as Underwriters Laboratory, the US Consumer Product Safety Administration or Food and Drug Administration (FDA). Such evaluations may be true for basic physical or chemical safety, but not for evaluation of the security of the software and communications protocols used to control them, and clearly not for the safety issues related to the software. For example, although the FDA's approval process for medical devices differs based on the criticality of the devices, the validation requirements are not clearly specified and rely on the manufacturer to assure that the programming is secure. Devices are not expected to be absolutely safe to receive approval, only that they are not obviously flawed under basic assumptions[22]. FDA guidance documents for software validation[31] are concerned with the software used to control manufacturing processes and for large-scale computer-based equipment, not computerized devices. The focus is on access, privacy and audit records, not secure programming practices.

Numerous guidelines for secure programming exist, but they are typically focused on traditional information technology, not cyber-physical systems, focusing on data security, not safety. In [20] the author presents the state of software engineering for safety in 2000 and six directions for future work. The author argues that to improve software engineering for safety that advances in other fields of computer science should be exploited, and that better integration of safety techniques into the industrial development process is needed. We feel it is

important to use advances in fields beyond computer science is also needed. The problems solved, and the methods used often have analogs in other areas.

The field of safety engineering is large, well developed, and shown to be effective when followed. Two aspects, *functional safety* and *intrinsic safety*, illustrate some ways systems are designed to be safe.

A device is intrinsically safe if it is physically impossible for an unsafe condition to occur. This is common in medical devices such as EEG monitors where opto-isolators link the electrodes to the device via an optical (light-based) connection, preventing electrical shock from a back-flow of stray device currents across the electrodes. In industrial environments where explosive gasses are present, intrinsically safe devices operate at voltages and currents sufficiently low to eliminate any risk of heat or spark sufficient to cause ignition.

Functional safety refers to adding parallel components or systems to detect and prevent an unsafe condition. Safety interlock switches to de-energize a circuit if an access panel is opened, ground fault circuit interrupters that prevent electric shock from wet equipment, over temperature monitors on motor windings, fire and toxic gas detectors are all related to functional safety.

Both of these methods of safety engineering involve study of the physics, mechanics, electronics, or chemistry of the system. When these devices are controlled by programmable electronic or controlled by remote (e.g. radio) communications, little is done to ensure that this aspect of the devices are safe, and particularly when considering the effects of malicious actions, secure.

This paper discusses security issues related to safety for cyber-physical devices in pervasive environments, major challenges that exist, and some steps toward improving the situation.

2. SECURITY + SAFETY ISSUES IN CPS

The goal of pervasive computing is to imbue the environment and devices within that environment with the ability to sense, process, and inter-communicate information to assist the people in that environment, to include their well-being (health, safety), support (enhance and extend daily functions), and leisure (gameplay, entertainment, socializing). These systems must minimize the risk of harm to people who use and/or become to depend on it. Systems of the first class and those of the latter 2 (for cases where injury or direct harm may occur) have clear safety concerns. A secure environment will minimize the risk of harm from malicious actions, whether targeted or incidental to other activity. The goal of security relative to safety is to prevent malicious agents from causing a situation that results in or is likely to result in bodily harm or damage to the environment. This can be extended to accidental actions that, if done maliciously, would result in similar situations.

Pervasive environments take actions (including making recommendations) based on data. Their security then ultimately rests with securing data.

We see 6 classes of data:

- 1) Needed data originating within the environment is not received in a timely manner by other element(s) within the environment.

- 2) False data originating within the environment is received by other element(s) within the environment.
- 3) Needed data originating within the environment is not received in a timely manner by element(s) outside the environment.
- 4) False data originating within the environment is received by element(s) outside the environment.
- 5) Needed data originating outside the environment is not received in a timely manner by element(s) within the environment
- 6) False data originating outside the environment is received by element(s) within the environment.

We consider the “environment” to be a defined area – a car, a home – with pervasive elements. For example, given a smart-car, it will need external communications (e.g. GPS, communications with emergency services) and internal communications (vehicle speed, direction, position of nearby vehicles, emergency status). How malicious failure of these flows can affect safety will need to be assessed to select the proper means of securing the system.

Because safety is the goal, an analysis that starts with the ways people (or the environment) could be harmed and investigate how these situations could occur through malicious actions. This is similar to the functional safety/security methods mentioned earlier. It requires a bottom up failure analysis and what could cause such a failure as a consequence of mission or bad data.

A more general approach considers basic categories of security flaws that affect safety that are commonly encountered in fielded cyber-physical devices. These issues are responsible for most all vulnerabilities that can compromise safety. The classes listed do not have hard boundaries and often intersect other classes. These are not intended to be a taxonomy such as [13], rather categories of security problem to be considered when developing or evaluating cyber-physical systems.

Of particular interest are situations that will result in a loss of control, loss of visibility into the process, or loss of control data. These are respectively where (1) you can see what is going on, but you are unable to make changes to the system; (2) you may be able to control the system, but you have no information about the system; (3) similar to 2, but unknown to you, the information about system is in some way flawed.

2.1 Vulnerabilities

2.1.1 Poor access control / Lack of authentication

Most cyber-physical systems are built using simple, open communications that lack authentication or encryption. This has long been a concern for privacy, but is also one of safety. Because an attacker can send unauthorized, potentially malicious commands to the devices, it is possible to cause harm. The earlier example of the Medtronic pacemakers is an example[11]. This is a typical situation; passwords are not needed to access the device. Wireless communication with the device is not encrypted. This makes it easier for an attacker to collect data and in the absence of technical documents for the device, to reverse engineer the protocol, and analyze it for weaknesses. With knowledge of the protocol, an attacker can disrupt communications with the device (i.e. create a denial of service attack), or inject packets to communicate with and control the device, possibly shutting it off, or changing the devices configuration, which could cause injury or death.

Some system developers consider limited range of remote wireless access effective. A weak transmitter/receiver on the device will restrict communications; however an attacker with a more capable transceiver can easily defeat this.

2.1.2 Poor input validation

The most common types of attacks on any system, pervasive or not, send data that is in some way ill formed relative to what the receiving process expects, for example, sending too large a message, an undefined function code, or a timestamp that is much earlier or later than the actual time. Ensuring that all data is validated, i.e. meets expectations, before being further processed is an effective way to mitigate such attacks. Validation checks verify that data received is well-formed, meets design expectations, and makes sense given local context. This is true for all types of systems.

Poorly formed messages can cause parsing or overflow errors, resulting in unexpected behavior such as the device halting, or allowing the attacker to execute arbitrary code on the device. This is a common attack vector for malware. It is becoming common to see embedded SQL database servers in low-level cyber-physical devices. While questionable from a security standpoint, this provides additional and familiar functionality to what were traditionally simple devices with limited capabilities. It is reasonable to expect the same trend to affect pervasive systems. Database applications are notoriously vulnerable and subject to many attacks. Failure to validate input on these systems may allow an attacker to bypass the user interface or command structure and improperly execute database functions, modify web pages, or even bypass authentication when present.

Depending on the device and how it is used will determine how unsafe the security vulnerability causes it to be. Clearly stopping or misconfiguring devices assistive systems or medical support systems could cause harm.

2.1.3 Lack of robustness

Many designs make too many assumptions about the conditions under which the software will operate and the data it will process. When exceptions are encountered the device can fail. Input validation can prevent some errors; others are more fundamental and are not as easily mitigated. For example, to save space and limit complexity, a system assumes only one network communication will occur at a time, and implements the system to accept only one network connection. An attacker can connect to the device, and using a limitation of the TCP/IP protocol, block all other communication with the device. In another example, the developers determined that at most 4 connections could occur at a time; however an attacker forced more than 4 causing the device to crash.

Robust software is introspective. All error conditions are caught. All exceptions are caught. All unexpected or out-of-spec situations are monitored and handled in a safe, deterministic manner. This level of detail is very challenging and resource intensive, which is why it is rarely done. However, identifying those sections that could lead to an unsafe condition should be identified and protected.

Single points of failure are primary attack targets. To be robust against defects or damage to their components critical systems are often implemented with redundant components and communications paths. If one fails, the added components will continue to function. However, simple redundancy is not sufficient; how the redundancy is organized can be critical. For

example, redundant network cables in an aircraft will not be located together, but distributed across the craft such that damage is less likely to affect more than one. This same concept should be considered for any CPS where safety is an issue.

2.1.4 Implementation errors

Programming errors, rather than design flaws, are generally considered the major source of vulnerabilities in any software systems. This can be caused by overly complex designs, incompletely specified designs, the use or improper use of insecure library functions, and simple programming mistakes. A common example is the use of string copy functions that do not ensure that the destination is large enough to hold the string being written. The extra characters are written into the space for other data, which can cause unexpected behavior, crashes, or in the worst case, allows the attacker to control the device.

2.1.5 Limited Interoperability

Pervasive environments will have many different devices which must work properly in the presence of each other and in many cases work cooperatively. Without shared, standardized protocols (with clear and complete published specifications) interoperability is not likely to occur. Attempts to bridge different protocols together can negate security features and allow opportunities for attacks.

Most all wireless communication uses the 2.4Ghz ISM-band. Any time a large number of devices attempt to share spectrum, interference will occur. Failure of devices to communicate in a timely manner may lead to unsafe conditions. For safety critical systems, protocols that are robust against interference must be used.

Bluetooth, an IEEE 802.15.4 based communication protocol for short range communication is becoming increasingly popular. This is in part because rather than each manufacturer attempting to implement the standard, interoperable libraries that implement the standard are available. Bluetooth, while far from perfect does provide for authentication using pre-shared out-of-band identity codes (analogous to a key).

Two other IEEE 802.15.4 based protocols ZigBee and WirelessHART, both implement self-healing wireless mesh networks and offer full NIST standard encryption and message authentication, and increased range. They are becoming increasingly popular for environments where longer range and robustness is desired. Unfortunately, while compatible at the MAC to Network layer, without a common application protocol devices cannot interoperate.

To support interoperability, the ZigBee Alliance provides “gateway protocols” that allow exchange of task specific data using well defined protocols. These include energy monitoring, home automation, telecom/retail services, and health care monitoring. While imperfect, these are an important beginning at providing secure interoperability.

2.1.6 Lack of preventative safety

Many cyber-physical devices and systems will have specifically designed features to ensure safety. This can be as simple as an interlock switch to stop the spin-cycle on a washing machine, to a complex program to safely shut down a chemical process when equipment failure is detected. Implantable medical devices should have a fail-safe mode to prevent a device failure from escalating. These safety systems monitoring and mitigate mechanical or electrical failures, as well as some software

errors. Robust software will have many integrity checks to ensure that the program is operating as intended, that values are within range, and pre/post conditions for an operation are maintained. However, the overhead for this level of integrity checking, and the effort required to implement it, exceeds its perceived worth. Safety monitoring should be independent of the devices basic control software. If not it may be bypassed. Consider the case where a buffer-overflow vulnerability is exploited, allowing the attacker to change the flow of execution of control program; if not separate, the safety system will also be bypassed.

2.1.7 Naïve assumptions about security

Often developers scoff at the idea that their product is insecure, and even when the problem is demonstrated, they argue, “Who would figure that out?” or “Who would do that?” Failure to fully understand that there are many individuals who are highly intelligent, skilled, and motivated toward finding and exploiting weaknesses in computer systems and devices is dangerously naïve. Part of the problem is a lack of understanding of security, but commonly an ego that fails to accept that there are people smarter than themselves with malicious intent.

The recent Stuxnet virus[9], perhaps the most sophisticated malware event to date, required deep understanding of the Windows operating system, Siemens programming environment and programmable logic controllers, detailed operations of the Netanz uranium enrichment facility in Iran, and the business relations of the facility. This illustrates the extent that a group will go to reach a desired goal. Just as prior to the 9/11 terrorist attacks which used loaded passenger aircraft as suicide bombs, “who would do that?” is no longer a question. Prudent thought has moved to “what could happen?” and “how can we prevent it?”

2.1.8 Proprietary solutions

It is common to hear from developers that their product is secure because they have developed their own, proprietary solution. This is particularly common with respect to cryptography. In almost all cases this is a bad idea. Rather than use a method that is well understood, developed by experts in the field, and has been subject to evaluation by thousands of experts, some organizations believe their proprietary solution is more secure. This is based on a severe lack of understanding of security.

Security experts have long understood the limited protection provided by a secret design (security by obscurity). True security comes from a system that is completely open, yet provably unbreakable under realistic conditions. Secret protocols are commonly reverse engineered. The security of in-house cryptographic solutions is generally naïve and easily defeated, even when derived from proven low level cryptographic algorithms. A common mistake is to use them in a way that defeats the security of the underlying algorithm.

2.1.9 Safety lockouts

Pervasive environments have multiple independent systems used either concurrently or in support of each other. At times these may not interoperate safely. Often no formal analysis of how an additional system may adversely affect others is done prior to their integration. The initial effect of microwave ovens on pacemakers was a surprise to many. When an interaction is known to be hazardous, safety interlocks should be used to prevent unsafe interaction, but this is commonly not the case.

During laser surgery it is crucial to not have anesthetic gas present. Cases where a fire was started inside the patient as a result of this have resulted in injury and death. The safety issue is well known. Airway Fire Protocol during Laser Surgery requires that anesthetic gas, which provides a highly oxygen enriched environment, to be discontinued and vented prior to use of the laser. Rather than rely on operators to follow the protocol, the devices should be linked to provide safety interlocks to prevent the situation from occurring[7].

2.2 Trends

Malware such as computer viruses and worms are the single largest class of security problem for cyber-physical systems[26]. This will increasingly be the case as systems are built using common, feature-rich operating systems such as Microsoft Windows. The complexity of these, plus their being widely known, with widely known vulnerabilities, increases their risk.

According to DHS sources, the Conficker virus has infected computerized pacemakers through wireless and other LAN connections[33]. Frustratingly, government regulations prevented removal of the virus because to do so was considered a modification to the certified software[8].

A growing threat is targeted malware. This refers to malware crafted to affect a specific system while benignly transiting others. Once the stuff of science-fiction thrillers, these can target an individual or a type of device with the intent to harm. For example, because medical devices such as pacemakers have unique identifiers, malware to target the device of a particular person is possible.

3. CHALLENGES

The individual problems that result in insecure and unsafe systems are themselves the result of several fundamental challenges in creating safe and secure products.

3.1 Usability vs. security

As discussed earlier, devices such as pacemakers have no encryption or authentication to protect against unauthorized communication with the device. In many cases this was a design decision that put usability above security. It was judged that in a crisis, it was more important for a physician to be able to modify the function of the patient's pacemaker than to secure the device. The complexity of managing cryptographic keys and having access instantly when needed is significant. Given that the nominal range of communication was several meters, the risk was considered small. ICDs were "not designed to withstand terrorist attacks"[14] stated Bruce Lindsay, president of the Heart Rhythm Society. However, the risk of attack must not be underestimated. To quote[18], "Permitting control of a component in a human body without authentication seems grossly negligent, and should raise the ire of the FDA." Several suggestions of how to add encryption yet not significantly impact usability have been proposed. These include tattooing the encryption key on the patient in an invisible, UV light readable ink[29], adding a subcutaneous push switch which must be depressed to reprogram the device, RF shielding to restrict communication to a few millimeters, to an embedded synchronized secure token. An additional feature suggested is to provide an audible notification when the device connects remotely or critical configurations change[12]. This would warn a patient if the device's security might be at risk.

3.2 Push to market

A constant challenge for businesses is getting their products to the market in a timely manner. This leads to employee pressure, short cuts, mistakes, lack of testing, all of which contribute to selling insecure and potentially unsafe products. Not only are products sold before the associated security aspects are understood, in sync is the tendency of individuals and organizations to adopt these new technologies. The excitement and perceived benefit of the product eclipses concern for the deeper aspects of safety and security.

3.3 Lack of oversight

Product safety is highly regulated. The US Consumer Product Safety Administration has strict standards for product safety; however safety as affected by computer security is not directly covered. As mentioned earlier the FDA has oversight of medical devices, but does not do software vulnerability assessments.

3.4 Lack of understanding of cyber-security/safety issues

All of the above issues are ultimately a result of lack of understanding and lack of concern of the cyber-security issues and how security flaws can result in safety issues.

This problem is also a factor with consumers. One reason cited by manufacturers for not building more secure products is a lack of market pressure from consumers, where price and function are more important than safety. This is clearly seen in surveys of automobile purchasing decisions. According to Consumer Reports, while 64% of 2010 respondents considered safety a factor in vehicle selection. It was reported as the top factor by 31% of women, 19% of men[8]. In a separate 2010 study[15] safety was the top consideration by 22% of those polled. Interestingly, safety was considered more important by those with lower incomes (25% vs. 15%) and by parents (27% vs. 18%). In both studies quality (~reliability) was the most common top choice. Price was generally second for all respondent classes.

Without external incentives, manufactures continue to place insufficient focus on safety. While the number of incidents is currently low, the expansion of cyber-physical devices into the environment will result in more incidents. Malicious actions tend to follow trends – the more common a device or systems is, the more focus on finding and attacking the product becomes.

4. SOLUTIONS

Improving the security and safety of cyber-physical systems will require many different approaches over a period of time and will require effort across the spectrum of consumer, developer, industry and government. Early steps will enable for future efforts, making solutions more approachable and cost effective.

4.1 Education and Awareness

Educating developers about software and communications security and how it affects the safety of their products, and convincing them of its importance, is perhaps the most fundamental step towards a solution. Without the acceptance of those involved problems will continue to exist. Acceptance must not be just by the programmers, but by corporate management who has the power and authority to make it a driving force throughout the organization.

Often financial decisions push products to market without proper time and effort to make the products secure. Part of awareness education must point out the ramifications to the organization for failure.

4.2 Standards for Security and Reliability

Standards are useful in creating secure products. These can help a developer ensure that all known issues have been considered, particularly for complex systems or when the individuals involved are not experts at security and reliability. Perhaps the best standards for security, reliability and safety for control systems are the ANSI/ISA 99, ANSI/ISA 100 and ANSI/ISA-84 documents[1][2][3]. These evolving, international, industry-created standards cover security, wireless communications and safety. Many others exist such as NIST 800-82. The FDA and other government and industry groups have similar standards.

One concern with standards, particularly regulatory compliance standards, is they often are compromises to improve acceptance by the regulated group, and specifically exclude certain areas from needing to be secured. For example, the NERC-CIP cyber security standard for the North American bulk power system define narrowly which devices are to be considered critical assets and require securing and similarly excludes non-routable protocols. Given that organizations typically do the minimum to meet regulatory compliance – not to be secure. This directs attackers where to focus their efforts. A growing number of people are concerned that these standards as written and enforced may actually undermine security and that other means to promote security be investigated.

4.3 Security Certifications for Cyber-Physical Devices

Security certification of devices can help consumers make informed purchasing decisions, or even allow devices to be sold. Currently these are limited to general safety, not cyber-security, and as discussed earlier, are not robust. Effective methods of certification for software security, particularly as it relates to security needs to be created. This has already begun, although perhaps accidentally.

The US Department of Defense Information Assurance Certification and Accreditation Process (DIASCAP) program mandates strict certification requirements for its computer systems. Coincidentally, this includes programmable medical devices and other computer controlled devices (e.g. prosthesis) provided for the military. For these devices to be sold to the DoD, they must meet computer security certification. This has caused some problems as many if not most manufacturers are not currently willing or capable of the cost to meet these standards. Some manufacturers feel this market is not large enough to warrant the effort.

Started in 2010 as part of the ISA Secure Certification Program, the Embedded Device Security Assurance (ESDA) program[16] is providing a rigorous set of evaluation criteria for the security of control system devices and a trademarked certification logo that can be used to promote the device's security. ESDA is divided into three areas: functional security assessment (FSA), device communication robustness test (CRT), and organizational software development security assessment (SDSA). While the first two evaluate the subject device, SDSA verifies and validates that the software was developed using appropriate engineering practices. FSA tests to see if the software behaves as intended, that is, relative to software requirements documents.

How effective this program will be and its perceived value is still to be determined, but is a good first step.

4.4 Certification of Developers

Software vulnerabilities are the result of a programmer's failure. These may simply be errors, but are most commonly the result of a lack of understanding of secure development methods and oversight by project management. By having personnel specifically trained in security and secure development procedures in place, most all vulnerabilities can be removed. This belief is being seen in the IT industry, now commonly requiring certifications for their employees and is now required for those working on US DoD systems under DoD Directive DoDD 8570[30].

4.5 Product Liability Reforms

At present it is not clear who is liable for harm caused by insecure software products and to what extent. If organization can avoid responsibility, they have little corporate reason to build secure products. There is some movement in this direction. Under Sarbanes-Oxley act of 2002 director level personnel can be held personally liable for failure to take due diligence in computer security. Until their financial well-being is at risk security and safety are not likely to improve. The 2010 study by August and Tunca[4] looked into what was the most effective way to get developers to improve the security of their products. They compared vendor liability for damages, vendor liability for patching, and government regulations across different incidence level of zero-day attacks. They found that for low occurrences of zero-day attacks, government regulations were most effective, but at higher levels liability for patching was most effective. Other studies generally conclude that vendors whose product resulted in physical harm as a result of safety problems due to software flaws, would be liable for damages, even if the injury were the result of hacking[23][27]; however this will ultimately be resolved by the courts.

4.6 Apply Safety Engineering to Security

Formal security evaluations of products are typically either risk-based or a functional security assessment. Risk-based evaluations examine the threats and focus efforts based on perceived damage and probability of security events. Functional security assessments are driven by software requirements and whether the stated security controls are in place. In many cases this becomes a check-list form of evaluation.

As mentioned earlier, functional safety refers to the design and implementation of components that are added on in parallel to the primary system that monitor and prevent safety problems from occurring. These may be mechanical, electric, electronic, or programmable electronic, the key aspect is that they are separate and parallel. We propose a similar approach to software security, functional security, but defined relative to functional safety, not functional security assessments as described above.

Functional security is intended to be part of the software development phase. It is a methodology to analyze software and add functions to identify insecure and/or unsafe situations. Our idea is to use the wealth of information in security engineering and adapt it to security analysis as is appropriate, to help design software that is secure and safe.

Reusable models to allow engineering of safety interactions such as [7] though non-trivial will be needed, but must be extended to consider security issues related to malicious actions taken to

send false data, prevent needed data, or otherwise compromise the environment.

4.7 Autonomic Safety/Security Response Models

In pervasive environments (PE), its computing should be invisible to those in the environment and should operate with limited direct interaction by its occupants. To that end we are investigating an autonomic response system to manage system functions to maintain safety.

The PE should be able to perform self-monitoring and be able to determine its safety status. Inspired by an operational model from electric power grid operations[21], we define operational states: (1) safe – no event occurring while in this state can result in harm; (2) correctively safe – a situation has occurred that if not corrected could result in hazardous conditions; (3) hazardous – immediate harm likely if operations continue. This is further divided into (3a) correctable and (3b) non-correctable hazardous states, the difference being that in the correctable state the system has maintained integrity and can by some corrections be made safe. In the non-correctable state, recovery is not possible and must halt and enter (4) the restorative state, where significant corrections are needed before operation of the system can be restarted. Under this model, *disturbances* and *actions* will move the system between states. A correction to move the state from correctively safe to safe is a *preventative action*. While actions to move the state from hazardous to a safe state are referred to as *corrective action* and involve more significant changes. For completeness we define a *restorative action* – making corrections and restarting the system. A *safety assessment* is made by collecting environment data, comparing it to a system model, and determining the new system state. A 5th state is also defined: rather than silently correct the situation, human intervention or direction is required. This may transit to another if a response is not received in a timely manner.

Corrections within this model includes repair of undesired situations; reducing functionality; halting operations; report situation to occupants. To support this, the pervasive environment will need to provide sufficient information to make state determination and the means to affect corrective actions.

Response actions are in part based on the function of the PE a system is providing (see section 2), different actions may be taken without negatively impacting safety. If a game console were overheating and may result in a fire, it could be turned off without loss of safety. If a power outage were occurring, the game console could be turned off for the duration of the event, the refrigerator to the extent that food would not spoil, and life support functions never disabled. If the problem persisted, the food would be allowed to spoil, rather than turn off life support. These are simple examples, but illustrate the overall concept.

4.8 Security Event Monitoring in Pervasive Environments

In pervasive environments (PE), the computing should be invisible to those in the environment, and should operate with limited direct interaction by its occupants. The embedded computing nodes will be limited in resources: CPU speed, memory, power consumption. Including a host based security event monitoring system in each node will not be feasible. Network based monitoring, given distributed mesh networks

will not have a single monitoring point, and encrypted communications, is also not feasible. A lightweight distributed monitoring system is required. Also, this system must operate invisibly, rarely requiring limited interaction with its occupants.

We have been investigating such a system inspired by the behavior of social insects (e.g. ants)[10]. This model uses the foraging behavior of ants to direct resources as needed, and minimize overall impact to the system.

This work can be adapted to pervasive environments. The overall design is to collect the minimal amount of information, only as needed, and to increase monitoring to focus on problems when they appear. Each node in the environment will include a very lightweight process to provide sensing to the virtual ants. A central aggregator device will collect sensor data from a selected subset of the nodes, and using that information, determines what data will be collected where on the next pass. This simulates ants moving between nodes, looking for a particular type of information, and if found, leave a scent trail to direct other different types of ants toward the node of interest.

This model has similarities to an artificial immune system; the system is constantly looking for undesired situation, and when found the system is active to determine its nature and handle it appropriately. The system should deal with the problem without the traditional high-volume of alert reporting common to traditional security event monitoring systems. This design will interoperate with a response system as described above.

5. CONCLUSION

Ensuring the security and safety of pervasive environments will be challenging, but needed to protect the welfare of those aided by them. As the use of cyber-physical systems expands and takes on more life impacting functions such as heart/brain control, drug delivery or transportation, the risk from unsafe/unsecure systems will be substantial. Significant changes to product development and testing must occur. Unfortunately, historical behavior and market forces are against this; the industry has a poor track record for developing secure software and communications. Without strong pressure to change little will. Consumers must change their buying habits to opt for secure products, even if initially more costly. Until then pressure from government bodies and the courts may be needed. Even with pressure, security and safety will not improve without good techniques to support their creation. Research into the safety engineering of cyber-physical systems, particularly in the presence of malicious activity is a necessity.

6. REFERENCES

- [1] ANSI/ISA-84.00.01-2004 Part 1 (IEC 61511-1 Mod) Functional Safety: Safety Instrumented Systems for the Process Industry Sector
- [2] ANSI/ISA-TR99.00.01-2007-Security Technologies for Industrial Automation and Control Systems
- [3] ANSI/ISA-TR100.00.01-2007-Security Technologies for Industrial Automation and Control Systems
- [4] August, T., Tunca, T.I. 2010. Who Should be Responsible for Software Security? A Comparative Analysis of Liability Policies in Network Environments. Management Science. (Dec. 2010)

- [5] Baker, G. 2008. Schoolboy hacks into city's tram system. The Telegraph. (Jan 11, 2008) <http://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html>
- [6] Cheolgi, K., Sun, M., Mohan, S., Yun, H., Sha, L., Abdelzaher, T.F. 2010. A framework for the safe interoperability of medical devices in the presence of network failures. ICCPS '10: Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems. (Apr. 2010)
- [7] Condon, S. 2009. Red tape keeps Conficker on medical devices. (5 May 2009) <http://www.zdnet.com/news/red-tape-keeps-conficker-on-medical-devices/295270>
- [8] Consumer Reports. 2010. Most important factors in buying a car. Consumer Reports. (Jan. 2010) <http://www.consumerreports.org/cro/cars/new-cars/news/2010/01/2010-car-brand-perceptions-survey/most-important-factors/brand-perceptions-most-important-factors.htm>
- [9] Falliere, N., Murchu L.O., Chien, E. 2011. W32.Stuxnet Dossier, version 1.4. Symantec (Feb. 2011) http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf
- [10] Haack, J.N., Fink, G.A., Maiden, W.M., McKinnon, A.D., Templeton, S.J., Fulp, E.W. 2011. Ant-Based Cyber Security. 8th International Conference on Information Technology: New Generations (Apr. 2011)
- [11] Halperin, D., Heydt-Benjamin, T. S., Clark, S.S., Defend, B., Morgan, W., Fu, K., Kohno, T., Maisel, W. H. 2008. Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. IEEE Symposium on Security and Privacy. (May 2008)
- [12] Halperin, D., Heydt-Benjamin, T. S., Fu, K., Kohno, T., Maisel, W. H. 2008. Security and Privacy of Implantable Medical Devices. IEEE Pervasive Computing. (Jan. 2008)
- [13] Hansen, J.A., Hansen N.M. 2010. A Taxonomy of Vulnerabilities in Implantable Medical Devices SPIMACS'10. (Oct. 2010)
- [14] Highfield, R. 2008. Hacking fears over wireless pacemakers. The Telegraph. March 13, 2008. <http://www.telegraph.co.uk/science/science-news/3336025/Hacking-fears-over-wirelesspacemakers.html>
- [15] IPSO/Carmax. 2010. IPSO/Carmax vehicle buying decisions survey. (Dec 2010) <http://www.ipsos-na.com/news-polls/pressrelease.aspx?id=4960>
- [16] ISA. 2010. ISASecure Program Description. (Mar. 2011) <http://www.isasecure.org/Certification-Program/ISASecure-Program-Description.aspx>
- [17] Koscher K.; Czeskis A.; Roesner F.; Patel S.; Kohno T.; Checkoway S.; McCoy D.; Kantor B.; Anderson D.; Snacham H.; Savage S. 2010. Experimental Security Analysis of a Modern Automobile. Proceedings of the IEEE Symposium and Security and Privacy. (May 2010)
- [18] Krush, W. 2008. Who's Hacking Your PACs?. Government Health IT. (May. 2008)
- [19] Leveson, N.G.; Turner, C.S.. 1993. An investigation of the Therac-25 accidents. Computer. v.26-7, pp 18-41 (Jul. 1993)
- [20] Lutz, R.R., Software engineering for safety: a roadmap, Proceedings of the Conference on The Future of Software Engineering, p.213-226, June 04-11, 2000, Limerick, Ireland (Jun. 2000)
- [21] Maharana, M.K.; Swarup, K.S. 2008. Identification of Operating States of Power System Using Transient Stability Analysis. Joint International Conference on Power System Technology and IEEE Power India Conference, 2008. (Oct. 2008)
- [22] Maisel, W.H., Kohno, T. 2010. Improving the Security and Privacy of Implantable Medical Devices. N Engl J Med 2010; 362:1164-1166 (Apr. 2010)
- [23] Mead, N.R. 2004. Who Is Liable for Insecure Systems?. IEEE Computer. Volume 37 Issue 7. (July 2004)
- [24] Meier, B. 2010. Lifesaving Devices Can Cause Havoc at Life's End. New York Times. May 13, 2010. (May 2010)
- [25] Moore, S.K. 2006. Psychiatry's Shocking New Tools. IEEE Spectrum. (Mar. 2006)
- [26] RISI. (2010) 2009 Annual Report on Cyber Security Incidents and Trends Affecting Industrial Control Systems. Security Incidents Organization (2010) <http://www.securityincidents.org>
- [27] Ryan, D.J.; Heckman, C.;. 2003. Two views on security software liability. Let the legal system decide. IEEE Security and Privacy. (Feb. 2003) <http://singularityhub.com/2010/09/08/80000-and-counting-brain-implants-on-the-rise-world-wide/>
- [28] Saenz, A. 2010. 80,000 and Counting, Brain Implants on the Rise World Wide. Singularity Hub. (Sep. 2010) <http://singularityhub.com/2010/09/08/80000-and-counting-brain-implants-on-the-rise-world-wide/>
- [29] Schechter. S., Security that is meant to be skin deep:Using ultraviolet micropigmentation to store emergency-access keys for implantable medical devices. Technical Report SR-TR-2010-33. Microsoft Research (Apr. 2010)
- [30] U.S. Department of Defense. 2010. DoD 8570.01-M, Information Assurance Workforce Improvement Program. Securing the Nation's Critical Cyber Infrastructure. (Apr. 2010) <http://www.dtic.mil/whs/directives/corres/pdf/857001m.pdf>
- [31] U.S. Department Of Health and Human Services Food and Drug Administration. 2002. General Principles of Software Validation; Final Guidance for Industry and FDA Staff (Jan. 2002)
- [32] Warner, J.S., Johnston, R.G. 2003. GPS Spoofing Countermeasures. LAUR-03-6163. Los Alamos National Laboratory. (Dec. 2003) <http://library.lanl.gov/cgi-bin/getfile?00852243.pdf>
- [33] Willke, B.J. 2010. Securing the Nation's Critical Cyber Infrastructure. (Apr. 2010) http://www.us-cert.gov/control_systems/icsjwg/presentations/spring2010/01%20-%20Case%20studies%20-%20Bradford%20Willke.pdf