International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.2, Mar-Apr 2012 pp-021-024 ISSN: 2249-6645

Security of Embedded Systems Applied in a Green-House Enviroment For Monitoring and Controlling Physical Parameters

Trailokya Oraon
Faculty, Master of Computer Application Department
Jorhat Engineering College
Jorhat, Assam, India

Nelson R Varte
Faculty, Master of Computer Application Department
Jorhat Engineering College
Jorhat, Assam, India

***Abstract*— Embedded systems are extensively use in the field of pervasive computing. These systems are use to such an extent that embedded systems are now controlled and monitored from remote locations. Both intranet and internet now extensively used to control embedded systems used in most areas of our life. With the introduction of Internet Protocol version 6 (IPv6) on the web, peer-to-peer communication between internet-enabled devices helped web services to make performance improvement. On the worse side, it created new attacks on the components used in the embedded systems. The paper discusses the details of security vulnerabilities of both stand-alone and web-enabled embedded systems used in greenhouse environment. To ensure the correctness of working of these devices monitored and controlled by different hardware and software components, security of the components is a major concern. Various vulnerabilities are introduced during entire development process of the greenhouse environment. The problem is to search the real threats, then define security policies and implement them during development process. The paper tries to find out possible security techniques to deal with the vulnerabilities. It shows to introduce security policies at various levels of life-cycle, be it before development or during development or after development.**

Keywords- ***IPv6;Cryptography;System vulnerabilities; Security analysis ;Sockets;Threat Model***

I. Introduction

Our lives and our businesses depend unavoidably on computing systems and, increasingly, on embedded systems in particular. Applications of embedded systems to monitor and control the greenhouse environment has become a notion of pervasive computing and now these computing devices can be applied and are available anywhere and anytime. Embedded systems are generally hidden from the user [1]. For example, its proper use in a greenhouse environment helps the crop-growers to maximize the productivity and better the quality of crops and seeds. Embedded products and tools that growers have at their disposal to control the environment are manipulated with respect to the important environmental influences on plant growth and development, for the actual optimization of the greenhouse environment. These embedded systems, which are ubiquitously used to sense, capture, store, process and transmit vital data used for control of the environment to maximize the photosynthetic process in the crop. Security observation is a basic requirement when an embedded system performs any of these tasks. As more and more embedded systems came to be accessed from remote locations, security became a major concern. Security of embedded systems employed in greenhouse environment provided new business opportunities and also prevents dangers of economical loss. For example, it prevents undue change of environmental changes and should help preserve grower's expectations. Examples of new business opportunities are secure embedded devices that are applied for use in timely production of crops. With the adoption of Internet Protocol version 6 (IPv6), network solutions for embedded devices, peer-to-peer communication is now possible and thereby giving easy access and control of such devices from remote locations. This has resulted an increasing number of security breaches, which have been detected in embedded systems in recent years, which also reveals the importance of fundamental security solutions. Security solutions, which are basically applied to embedded systems that also incorporate the concept of network model, are specific cryptographic algorithms, adding security functionality to the network security protocols or adding one more security layer to the embedded device [2]. In general, most developers apply security 'patches' only at the beginning of implementation phase or only if a security faults is detected. The embedded system data processing itself is vulnerable to its own system. The paper discusses the various embedded system security issues with reference to systems employed in greenhouse environment.
It is often argued for giving too much importance to security of embedded systems [3]. Security policies should be implemented to tackle real threats. Vulnerabilities arises when a system is connected to network and also when working as a single system. Mere encryption and decryption of incoming and outgoing data does not provide a complete integrated solution of security. Security should not be on short term benefits, but long-term benefits should be taken keeping the cost factors into account.

II. Life-Cycle Of Embedded System For Greenhouse Environment

In general, there are three phases of any embedded system: development, use and disposal.

- Development of the system for the environment
- Deployment and use of the system in the environment
- Disposal / Removal/ Transfer of the system from the environment

The "*Development of the system for the environment*" incorporates all activities starting from system requirement specification to acceptance testing. It also includes the activities until the final system is delivered. All activities are carried out only after proper study of the environment and various climatic factors that are going to affect the system. The various activities of this phase are Requirement specification, Design, Production, Product Shipment and Support/ Maintenance. The "*Deployment and use of the system in the environment*" starts when the system passes user's acceptance testing and the system is deployed in the environment. Deployment activities consist of installation of the system and configuring the system to match and create the greenhouse environment. Use phase consists of educating the users of the system, alternating periods of correct service delivery, service outage, service shutdown and maintenance. The "*Disposal / Removal/ Transfer of the system from the environment*" phase starts when the system no longer performs its activities. In such as case, the system may not be able to monitor and control the environment according to the requirement or the system may start malfunction under specific climatic changes. It can also start when the climate undergoes an abrupt change or the device is transferred from one person to another. Activities of this phase include removal of the storage media, software, components and data stored on the device. The significant aspect of this phase is that it prevents release of vulnerable data, information or software. In maintaining greenhouse environment, this phase will prevent flow or usage of data when the system is moved from one location to another. The security is needed to be considered in this phase to preserve and adhere to copyright, statutory and regulatory requirements, verify the authenticity, confidentiality and integrity of the system.

III. Sources of Vulnerabilities in Life-Cycle

It is important to analyze the vulnerabilities of each phase of life-cycle. Vulnerabilities and security threats can rise in every process and entities associated with processes created before, during or end of life-cycle of embedded system.

A. Vulnerabilities in Development Phase

The correct physical world phenomenon cannot be predicted at the start of a greenhouse environment project. Poor analysis of the environment conditions can be a source of vulnerability. In the development phase, the system and environment analyst who does not processes enough knowledge are a source of vulnerabilities. There may be some 'harmful' developers, who assist in this phase, whose aim is to harm the development phase with the available development tools, production tools and test tools, thus damaging hardware and software process and these can be the source vulnerabilities.

B. Vulnerabilities in Deployment and Use Phase

When system was brought to the environment, the persons responsible for installation and configuration, the administrators, users responsible for providing service, service providers, entities associated with the service, climatic conditions, natural disasters of the environment and adversaries from other living things of the environment are the source of the vulnerabilities. Improper handling of the system by users and lack of proper manual of running the system can be a source of vulnerability during the use phase. Though the basic source of new vulnerability during this phase is forcing the system to create the requisite environment even during the abrupt environmental climatic changes and conditions.

C. Vulnerabilities in Disposal/Removal/Transfer Phase

Security is not a major concern in design phase when the system is removed from the environment. A major source of vulnerability in this phase is when the system is transferred to another location with the information and data used in the previous location and the users are unaware about this fact. Available vulnerabilities and threats are ignored during security analysis when there is a complete discard of the system from the environment.

IV. Security in Product Design Methodology

Our requirement defines the embedded system design, what product we are going to apply, whom we are asking for a solution. It may range from a least expensive thermostat to computer that uses microprocessors that controls variety of equipments used to perform specific task. For some, it is analog device that considers multiple sensors in a single environment when it performs control actions. But it is always a device with some RAM, ROM, storage media and some peripheral device performing some desired input and output. Design should support development process so that the result is outcome-oriented. Developers have enough flexibility to adopt new tools so that they can deliver the best to meet the governance criteria and schedule tasks according to the design. Design methodology helps to analyze the requirements, conceptualizing, planning, developing, testing and supporting the components, products and solutions that respond to those requirements. So a product design advances through various levels down the completion and final delivery.

At the *Business level*, design process develops preliminary specifications for the proposed product and helps to enable investment risk. At the *operational level*, the product requirements are drawn in detail with adherence to plans and checkpoints. At the *schedule assignment, design, testing and debugging and delivery levels*, peer design reviews are conducted. The most critical stage in this process is testing and debugging where the test plans are executed. This is one area where the development process moves back to design. So the information assets need to be protected and the products developed to acceptable quality tolerance. At the *final release level*, the user manual and documentation are developed before taken to the deployment site. Here, the checkpoints help to test the quality of the product to acceptable conditions. Initial approaches to support security in product design methodology are described in [4, 5, 6, 7].It is necessary to know the threat(s) of a product under consideration and then devise security policies. Threats help to define policies and a secure product is designed based on those policies.

The embedded product design is a combination of hardware and software. The hardware is generally selected based on current availability. The most critical module is the software module selection. Most system vulnerability arises from this module, and even more, if it uses the concept of network model, hence this is one area where security is a major concern. The existence of hardware and software on a common platform is an area of vulnerability. There are certain vulnerabilities that exist side-by-side. For example the information gained from power consumption, electromagnetic leaks, timing, sound, physical implementation of cryptosystem that can make the system vulnerable[8, 9].Vulnerability assessment in design process is mostly done in software. Vulnerabilities arises when the *scope is too big for software* (software modules are written for existing software), when there is *too much interaction between software modules* (execution of a sub-function which never really belongs to the main executing function), *when software modules are not clearly defined* (there is no cleanly defined interfaces and function use different schemes to achieve results), when there is *no establishment of communication protocols* (lack of usage of proper global variables and not adhering to module development rules) and *inflexible platform* (modules calling the remote procedures of different platform and presence of security loopholes on the executing platform).

Embedded software products that are designed on client-server paradigm are highly vulnerable to network threats. Designing the software around sockets and messages opens up potential security holes in the system to the network. We require high degree of security in software module design as the users of the system always thinks the network as secure and thus forgets to apply security policies. Applying the network security policies in modular design, like client-server, will take the developer and other users quite some time to be aware of security functionalities.

No system is ever 100 percent secure. Security is not a mixture of cryptographic algorithms and security protocols to the system. Security is a process, not a product. It changes from time to time depending on requirement. Why we require security in embedded products varies from user to user and depends on what we are going to use, where we are going to use and who will use it. The basic requirement of security for most users is to preserve the identity of the system and authenticate the system so that others cannot access it. Different security responses of analyst, developers, administrators, maintenance personnel, authorize users of system are required to be generated during product design at any time, be it before development, during development, after development, during the use phase and during disposal phase. Good security policies will help the investors invest in the product confidently, thus making sure that their revenue is properly protected.

To remove some of the above vulnerabilities and threats of an embedded product, there is a need to enforce security to product design.

V. REQUIREMENTS FOR A SECURE PRODUCT DESIGN

The Security in the design should be considered from the beginning. In the previous sections, vulnerabilities in different phases in life-cycle and the requirement of security in design has been emphasized, now we see the possible security solutions for the threats[10, 11, 12].

Security analysis is an initial requirement in designing a secure embedded system. Most threats and their counter-measures are studied for a long time [13]. Security threats of and from both internal and external components should be considered. This will help to build a protected security policy. Generally, the security consideration of a system is frozen at the start, as development process for most products goes through adhoc approaches. There is always a need to hire a security specialist.

Security threats are analyzed by building threat models. A threat model is necessary to analyze the threats, assess the probability of potential harm and attack priorities. A hierarchical arrangements of threats in threat model helps to know the different attack goals of the attacker to the system.

A few basic points that should be taken in security requirements are:

- Hardware and software involvement in an embedded system design incurs cost. Securing an embedded product on heterogeneous platforms still incurs bigger cost; so only the real threats of the system should be considered.
- External sources of power needed by an embedded product should undergo change or recharge. Powerless systems open-up disposal security threats and loss of data.
- There should be a requirement to hire a experienced security specialist as most developers use tools they apply generally to all products without knowing the real threats.
- Too much security is harm to the product design itself [14]. So, equilibrium should be maintained between flexibility and security, thus saving resources.
- Violation of integrity should be checked so that components of the system do not behave abruptly under normal conditions.
- A system should learn to secure by itself. A self-adaptive, self-configuring or self-restoring techniques preserves security.
- Present security solutions should pave the way for new security solutions with less computational requirements, smaller size and lower energy consumption.
- Applying security policies should not cause the development process to cross the deadline in delivering the final product. Time-to-market is an essential which increases the value the product. So organizations should make to invest more money on right security practices to deliver the product within specified deadlines.
- A mandate for continuous security improvement in technology and manufacturing drives accountability and action.
- A community of hardware and software engineers that innovate and share practices and tools for secure product development.
- Integration across products that is achieved through client use cases, scenarios, and end-to-end usage threads in concert with an architectural framework that enables componentization.
- Consumability analysis that looks beyond product defects to the client experience of using the offerings.
- Organizations should be responsible for implementing security awareness programs for their development teams. Persons working in development teams should have different roles to create *security awareness programs* so that they deliver the right information to each other and the impact of security issues on the clients. The key concern on the technical details on security issues lies in the hands of developers.

Security standards, security practices and security compliance criteria play an important role in supporting the product development process of organizations. In the overall requirement, hardware, software and services development are governed by four necessary factors: assets security, check points in product development, security and quality plans, product testing[15].

Resources used in development projects are always required by an organization. In fact, they are never isolated. "*Security of assets*" aims in protecting disclosure of user's privacy, location and personal information and other proprietary information. "*Checkpoints Product in Development"* help to review the development when the project moves from one level to the next. These checkpoints can be used as control points for assessing project risk, expense control, product quality, security issue review, and for synchronization of a secure project plan. The project development team should define and remove the vulnerabilities for the prior work before transition to the next level. If there is any change in scope or content, it should prove the change from a security level. Though this paper highlights most of the security issues of embedded system with a special attention towards systems employed in greenhouse environment, but "*quality plans*" are also required. A *quality plan* highlights the technical and audit requirements for asset control, along with the standards and practices for quality engineering to be applied in the development process. The development checkpoint plan should help to review all of the security and quality plans, practices, and findings in the development phase. *"Product testing"* is required to verify the functionality of the various components of the product and whether it secures from the vulnerabilities in the product. It should go through all official design specifications of component, product, or solution. Security mechanisms and services included into the component, product, or solution should be verified.

VI. CONCLUSION

A very specific concept of life-cycle of embedded products and the vulnerabilities is discussed and why secure design is needed. The security philosophy of almost all greenhouse embedded systems is quite same. Some basic points that should be taken in security requirement are discussed and how the hardware security, software security and security of services are governed.To design a highly secure system, different design strategies are proposed. Further research is needed on strategies such as prevention, tolerance, removal and forecasting. Prevention and tolerance are basic security strategies while removal and forecasting are strategies for security assurance.

REFERENCES

[1] Peter Marwedel, "Embedded System Design", 1st edition,Kluwer Academic Publishers:Hardbound, pp.1-8, 2003.

[2] Rita C. Summers, "Secure Computing: Threats and Safeguards", pp. 3-11, McGraw-Hill, 1997.

[3] Paar, C., Weimerskirch, A. "Embedded security in a pervasive world", Inf. Secur. Tech. Rep. 12,3, pp.155-161. Jan.2007.

[4] Eric Uner, "A Framework for Considering Security in Embedded Systems", Embedded.com, Sept. 2005.

[5] Wayne Jansen , Serban Gavrila, Vlad Korolev, Thomas Heute, Clément Séveillac, "A Unified Framework for Mobile Device Security", Proceedings of the International Conference on Security and Management (SAM'04), pp. 9-14, June 2004.

[6] Ingrid Verbauwhede1 and Patrick Schaumont, "Design methods for Security and Trust", Design, Automation & Test in Europe Conference & Exhibition, pp. 1-6, DATE '07, 2007.

[7] Divya Arora, Srivaths Ravi, Anand Raghunathan and Niraj K. Jha, "Architectural Enhancements for Secure Embedded Processing", NATO Workshop on Security and Embedded Systems, VOL 2, pp. 18-25, August 2005.

[8] Weingart S., "Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses", Workshop on Cryptographic Hardware and Embedded Systems, 2000.

[9] J. J. Quisquater, D. Samide, "Side channel cryptanalysis", In proceedings of the SECI 2002, pp. 179-184.

[10] Srivaths Ravi, Anand Raghunathan, Paul Kocher, Sunil Hattangady, "Security in embedded systems: Design challenges", ACM Transactions on Embedded Computing Systems (TECS) ,Volume 3 , Issue 3, Pages: 461 - 491, 2004.

[11] David Hwang, Patrick Schaumont, Ingrid Verbauwhede, Shenglin Yang, "Multilevel Design Validation in a Secure Embedded System", IEEE Transactions on Computers archive, Pages: 1380 - 1390, 2006.

[12] Joe Grand, "Practical Secure Hardware Design for Embedded Systems", Proceedings of the 2004 Embedded Systems Conference, San Francisco, California.

[13] Ravi, S., Raghunathan, A., and Chakradhar, S. ,Tamper Resistance Mechanisms for Secure Embedded Systems", In Proceedings of the International Conference of VLSI Design. pp 605-611, 2004.

[14] Srivaths Ravi , Paul Kocher , Ruby Lee , Gary McGraw , Anand Raghunathan, "Security as a new dimension in embedded system design", In Proc. ACM/IEEE Design Automation Conf., pp. 753-760, June 2004.

[15] D. Allan, T. Hahn, A. Szakal, J. Whitmore and A. Buecker, Security in Development: The IBM Secure Engineering Framework. Available online: **http://www.redbooks.ibm.com/redpapers/pdfs/redp4641.pdf**