Analysis of Structure and Integrating Security in Embedded System

Isyaku Uba Haruna[1], Idyawati Hussein[2]
[1]Taraba State University, Federal University Dutse, Nigeria
[2]School of Computing, Universiti Utara Malaysia, 06010 UUM, Sintok, Malaysia.

Article Info

Article history:

Received Jan 10, 2025
Revised Feb 19, 2025
Accepted Mar 16, 2025

Keywords:

Embedded Systems
Threats
Cyber attack
Security

ABSTRACT

With a developing number of cases of data security-penetrates, there is an incredible requirement for endeavors towards making sure about electronic frameworks. Embedded systems are omnipresent frameworks that are used to catch, store, control, and access delicate information. Embedded systems aid humans to direct a suitable life and it is widely utilized in electronics, communication, home applications, aviation, health care, and many more. The security of these frameworks presents a few techniques and intriguing security challenges. In the current scenario, integrating security in an embedded system had become the main anxiety. So, there is a need to intend and establish techniques that can treat the attacks in the best manner. We project a technique to join security modeling into the design of an embedded system. In this paper, different security needs, attack technologies, and treatments for such attacks have been reviewed and analyzed.

Corresponding Author:

Isyaku Uba Haruna,
Taraba State University, Federal University Dutse, Nigeria.

1. INTRODUCTION

An embedded system is an application-explicit PC framework that is incorporated with a bigger mechanical or electrical framework, frequently with continuous processing imperatives. An embedded system in this way alludes to a framework that is constrained by a PC that lives inside the framework. 98 percent of all chip fabricated are utilized as parts in inserted frameworks. Embedded computing frameworks are consistently adjusted in a wide scope of applications, for example, car/transportation, government/military, clinical hardware, broadcast communications, flight/aviation, aviation hardware, office robotization, information correspondence, modern mechanization, and purchaser electronics. Embedded systems are liable for a huge number of wellbeing and security-basic applications. These frameworks likewise oversee basic data. Installed frameworks have interesting security issues with testing plan issues. Embedded systems keep on giving the center to a wide scope of utilizations, from RFIDs to satellites. Endeavors to consolidate security into many embedded frameworks have as of late been started. The asset and force reliance on implanted frameworks keep on being a test for condition of-workmanship security rehearses. There are numerous interesting difficulties in building security into embedded gadgets. Security is a significant issue due to the functions of embedded systems in numerous mission and security basic frameworks. Assaults on digital frameworks have demonstrated to cause physical harm [1].

The use of embedded systems in many applications has the unwavering importance of being safe and forgiving of flaws in this cutting-edge modern world.

structure. Thusly, embedded systems need to make sure about sensitive data, guaranteeing accessibility furthermore, giving a secure correspondence framework. Unwavering quality is straightforwardly related to security in embedded systems hence a framework that isn't made sure about is likewise an inconsistent framework. Essentially, security evolved as a problem associated with systems; moreover, encryption and integrated system designers consider it an additional component. Developing some security laws has directed that bargain on security can prompt awful results. Embedded systems are a blend of Software also, Hardware. Along these lines, to make a safe and dependable framework, the two parts must be made secure.

Model Integrated Computing (MIC) [2] is picking up wide acknowledgment in the field of installed programming plans. Models speak to installed programming, its organization stage, and its connections with the physical climate. Models encourage formal investigation, check, approval, and age of installed frameworks [3]. Thus, this methodology is predominant in the customary manual programming improvement measures. Although, there is displaying apparatus uphold for the examination of usefulness, execution, power utilization, wellbeing, and so on, as of now accessible devices consolidate nearly nothing assuming any uphold for security demonstrating. Therefore, security is taken a gander at just once the total framework has been constructed. In the best case scenario, this methodology of tending to security in the last phases of advancement is wasteful taking huge measures of exertion to accomplish just unassuming enhancements in security.

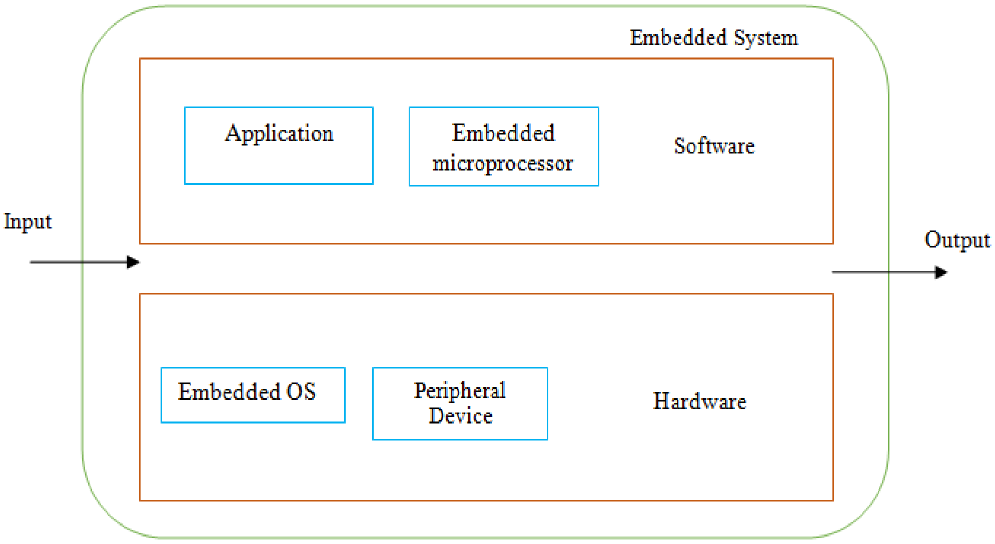

Figure 1. The Architecture of an Embedded System

Security in embedded systems is significant as it is incorporated into the constant well-being of basic applications. Most functionalities of the framework are upgraded by giving organizing capacities to them. In any case, it correspondingly builds the dangers for the aggressors. The aggressors can take points of interest of the organization based assaults, for example, Denial of Service (DoS), Man In The Middle (MITM), and replay assaults. The establishment of security into an implanted framework is obstructed by the limitations present in it. Those requirements incorporate cost, energy, processor, and memory [4]. Cost and energy assume a key function in the inserted frameworks as they help the producers to hold a spot in the market. The absence of handling power makes it hard for the engineers to actualize a cryptographic calculation and made sure about organizing conventions, for example, Secure Hypertext Transfer Convention (HTTPS) and Secure Socket Layer (SSL) without influencing its presentation. Then again, almost certainly, the headways in the innovation may support the framework to measure these calculations later on. Online protection is a disturbing issue in this timeframe as an ever-increasing number of gadgets are related to the web. The aggressors are quicker on assuming responsibility for these gadgets utilizing the most fragile security connection of the chain. Assaulting through the web gives aggressors an edge since it is hard to follow them back because of conflicting guidelines around the globe.

2. CHARACTERISTICS AND VULNERABILITIES OF EMBEDDED SYSTEM

Many of the embedded systems' constitutional characteristics have straight contact on safety-related problems. Some of the suggestions on characteristics and vulnerabilities in embedded systems are discussed [5].

2.1 Characteristics

Embedded systems are used in appropriate functional areas where conservative workstation or computer servers are not apt due to cost, power consumption, functionality, size. The individuality of the embedded system frequently comes with a few limitations of the following category:

Restricted handling power infers that an installed framework commonly can't run applications that are utilized for safeguards against assaults in customary PC frameworks (e.g., infection scanner, interruption identification framework).

Limited accessible force is one of the key limitations in inserted frameworks. Some such frameworks work on batteries and expanded force utilization diminishes framework lifetime (or on the other hand expands support recurrence). Consequently, the installed framework can devote just limited power assets to giving framework security.

It is typical for embedded frameworks that are communicated outside of the owner's or administrator's direct supervision to be physically introduced (e.g., public area, client premise). In this manner, inserted frameworks are innately powerless against assaults that misuse the physical nearness of the aggressor.

Remoteness and automated activity are fundamental for the installed framework that is sent in blocked off areas (e.g., cruel climate, distant field area). This constraint infers that conveying updates and fixes as finished with traditional workstations is troublesome and must be mechanized. Such computerized systems give possible targets for assaults [6].

Network availability through remote or wired admittance is progressively normal for inserted frameworks. Such access is essential for a controller, information assortment, refreshes. In cases where the implanted framework is associated with the Internet, weaknesses can be misused distantly from anyplace.

These traits result in a distinct set of embedded system vulnerabilities that must be taken seriously.

2.2 Susceptibility

Embedded systems are vulnerable to a wide range of abuses, including stealing confidential data, flexibly using up force, destroying the system, or obtaining the system for a purpose other than the one for which it was intended. Examples of installed platform vulnerabilities include:

- Energy waste (weariness assault): Installed structures with little battery power are defenseless against attacks that draw down this valuable resource. Increasing the amount of processing the workload, which decreases time spent resting, or increasing the use of sensors or other peripherals can all help achieve energy leakage.
- Physical disruption (altering): When inserted frameworks tend to be close to a prospective attacker, they become vulnerable to attacks where physical access to the framework is essential. Models are sneaking around attacks or power investigation attacks on the system of transport. Presentation of fabricated data (credibility): regardless of whether the off-base information is presented directly to memory or through the framework's sensors, embedded frameworks are powerless to stop it. Models are erroneous; for example, monitoring cameras' video recordings or power meters or more' overwriting of measurement information.
- Confusion or damage to sensors or other peripherals: Installed frameworks are vulnerable to attacks that result in incorrect sensor or peripheral activity, much like when malicious information is presented. A model is interfering with a sensor's adjustment.
- Thermal event (warm infection or cooling system failure): Embedded systems must function in environment-appropriate settings. There is an anticipated vulnerability to attacks that overheat the system (or create other ecological harm) because of the embedded frameworks' unusually exposed operating environment.

3. ATTACKS ON EMBEDDED DEVICES

Fundamentally, assaults on three categories may be employed to group integrated systems, for example, programming assaults, physical assaults, and side-channel assaults. Programming assaults having the biggest offer altogether several assaults on embedded systems and in this manner, it is very hard to ensure against such assaults. Here we will talk about programming assaults and countermeasures against these assaults.

Physical attacks: Embedded system is categorized into two parts. They are,

1. Circuit board systems
2. Chip-based system

Aggression on circuit board integrated systems can be initiated by attempting to intercept communications between peripherals. While introducing aggression on chip-based system, micro probing approaches are needed. Physical attacks are moderately tricky because they need a costly framework, as well as composite technologies, are utilized. Since it regulates every action of the installed framework, the microchip is the most important component of every inserted framework. Before handling the finished product, it is crucial to use JTAG to disable access to the mcu's internal components since it can be used to launch attacks on the microcontroller [7, 8].

Side Channel Attacks-Side channel assaults rely upon watching framework properties for example time, power utilization while the framework is performing calculations, for example, cryptographic activities. Timing data may point to the mystery key in some inserted frameworks, but it can also yield very little information. Nevertheless, it has been observed that the entire enigma answer can be found with a proper analysis of timing succession. In addition to the fact that power consumption can also trigger the whole enigma essential, and well-equipped labs are more reasonably priced and have the equipment necessary to measure variations in force use with a precision of approximately one percent. Scheduling attacks can be prevented by adding erratic planning delays to various operations. Similarly, energy consumption attacks can be defeated by adding arbitrary noise or by properly protecting the equipment, but this may result in higher equipment costs.

Software Attacks: A lot of well-known attacks in embedded systems are software attacks. Programming attacks are smaller than side-channel attacks and physical attacks, and they don't require large frameworks, which makes them difficult for installed framework plans to handle. These attacks could be further divided into three categories. 1. The attacks by viruses 2. Overflow of the buffer 3. Taking advantage of software flaws [9].

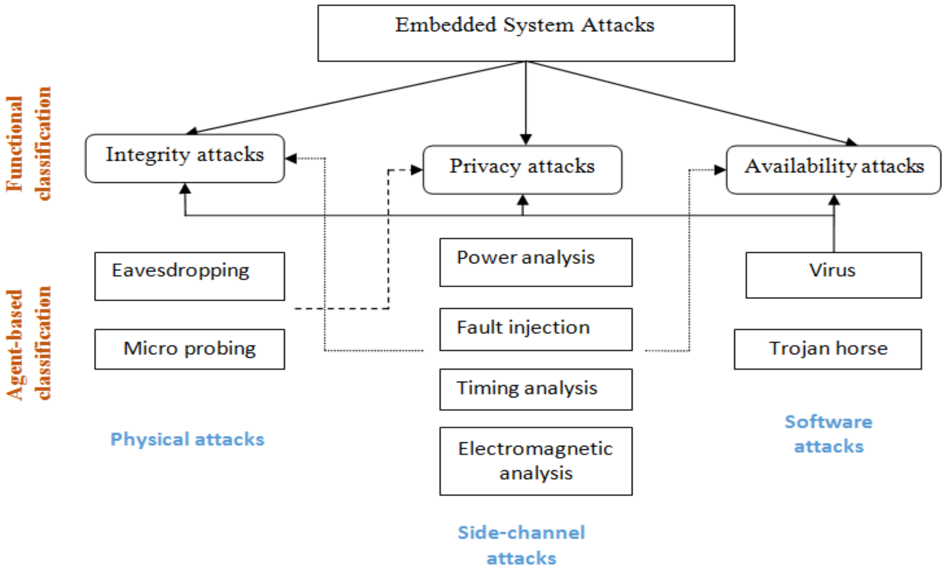

Figure 2. Attacks on Embedded Systems

1. Trojan, Worm, and Virus: These attacks are carried out by malicious operators such as Trojan, Worm, and Virus.
2. 2. Susceptibility Exploitation: A presentation is a sector area at which an attacker may attempt to gain access indirectly, but a weakness allows the attacker to increase direct access to the end-framework.
3. Buffer Overflow: When a support fails to store data, more data is added to it than it was designed to handle. This is known as a buffer flood. This flaw could be used by a gatecrasher to take over a framework. When support is used with helpless limit checks, this situation arises.. Support limits might be abused due to off base circle limits, design string assaults, and so forth Support flood impacts can incorporate overwriting stack memory, stacks, and capacity pointers [10].

4. STEPS TO INTEGRATING SECURITY IN EMBEDDED SYSTEMS

Embedded system design should have the following steps while addressing security [11].

Step 1: Perform an End-to-End Risk Assessment

Improving the security of an implanted gadget begins with identifying the expected dangers. These dangers must be assessed in the setting of the gadget maker, administrators (if the gadget is provisioned in such a manner), and end clients, including their usage environment. Dangers are depicted as far as an assault vector (how the assault is executed on the gadget) and the weakness it abuses (the shortcoming or flaw in the equipment or programming that permits the assault to misuse the gadget). Instances of assault vectors remember a wired Ethernet association for the gadget utilized for communication, and regular administrations, for example, web (HTTP), FTP, SSH, what's more, investigate operators.

A total item life cycle investigation should be performed: This examination must incorporate designers, manufacturers, administrators, merchants, retailers, and end purchasers to catch the all-out use sway on gadget security.

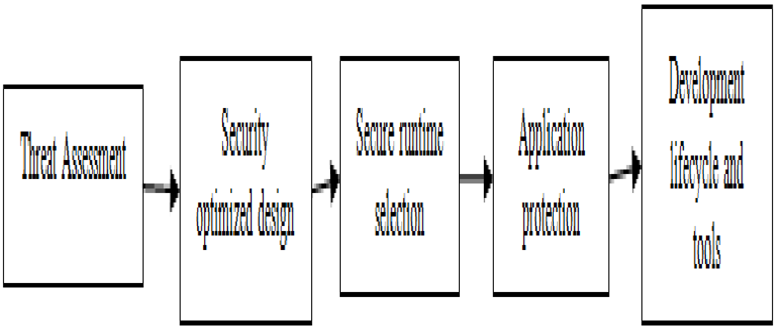

Figure 3**.** Steps to Integrating Security

Stage 2: Leverage Existing Advanced Security Designs

Various advancements and plan approaches have developed to address the consistently expanding dangers too associated with gadgets. An inexorably significant worldview for cutting edge security plans is utilizing demonstrated business off-the-rack (COTS) framework segments that can improve gadget security while controlling costs. Instances of such security segments incorporate inserted virtualization, working framework apportioning, and middleware, and apportioning these parts into virtual run-time conditions for expanded partition and deliberation. Virtualization is picking up prevalence in inserted frameworks because it empowers different working frameworks to run on a common hard-product stage. This gives adaptability in the framework plans and permits creators to get more out of their equipment than with single-OS frameworks. What's more, it can give an establishment to apportioning a gadget's activities into virtual execution conditions, which can empower the division of concerns and encourages conveying higher criticality parts with less delicate code on a mutual stage.

Stage 3: Select an Appropriate Run-Time Platform

Determination of a fitting business run-time stage for an implanted framework is a key thought. Executing a framework with segments that have COTS security proof can increment the security and diminish the expense of improvement of the general stage. There can be extra advantages of utilizing COTS programming segments rather than roll-your-own (RYO) code or self-ported furthermore, kept up open-source code. Hardware uphold layer: Run-time stages, for example, hypervisors and working frameworks require low-level equipment support layers that contain gadget drivers for the particular equipment gadgets. Depending on business equipment uphold is a basic initial phase in utilizing off-the-rack segments, yet these ought to be obtained through known and confided in providers. Business contributions, for example, board uphold bundles (BSPs), are advanced for the objective equipment, accompanied specialized support and upkeep, and sometimes have accreditation proof to help inclusion into wellbeing and security-basic conditions.

- Embedded virtualization: To give extra framework honesty, an inserted hypervisor can give virtualization to empower advanced dividing plans, multi-OS capabilities, and backing of multicenter and other processor architectures. Running on the head of the business equipment uphold layer, an installed virtualization part with business certification proof can quicken and improve the security of gadgets with blended criticality segments.
- Real-time working framework: Many inserted frameworks require little impression, severe planning imperatives, or well-being/security certification.

Stage 4: Secure the Applications

Currently, installed frameworks are significantly more than the conventional devoted gadgets with a solitary method of activity. Installed frameworks currently regularly have various applications and normally have their capacities increased through programming and equipment updates and overhauls over the life of the gadget. Similarly as with work area what's more, worker applications, it becomes basic that installed delicate product applications have secure capacities since they are likely to be the objective of malignant code or information penetrates. Updatable or upgradable implanted gadgets can utilize a technique called "whitelisting" to improve security. Whitelisting permits gadgets to just acknowledge applications for download that are known to be ok for execution. Any product not on the whitelist will not be introduced and will be dismissed by the framework. Boycotting, a connection method that gives a rundown of known malware and infections is utilized by frameworks to dismiss downloading or introducing any programming that is on the boycott. Since boycotts are a lot bigger than whitelists and change substantially more quickly, inserted gadgets normally need adequate processing assets to deal with on-the-fly boycotting and likely can't uphold the regular updates, information capacity, and organization network required. This makes whitelisting an alluring and compelling strategy for implanted frameworks.

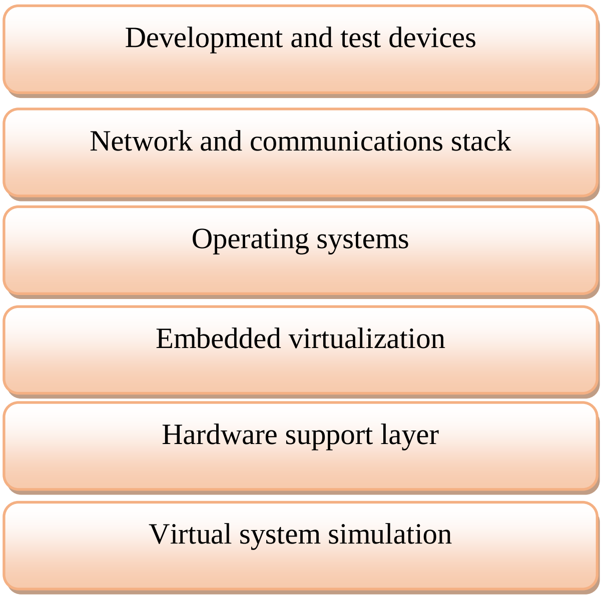
Development and test devices
Network and communications stack
Operating systems
Embedded virtualization
Hardware support layer
Virtual system simulation

Figure 4. Embedded Improvement and Run-Time Arrangement Stack

Stage 5: Adopt Comprehensive Life Cycle Support

Security continually advances as dangers change after some time. As a gadget gets mainstream (Stuxnet focused on a famous PLC gadget) or then again exists in the market longer, it can turn out to be more defenseless to assault. Numerous gadgets in the past were not intended to be field programmable or to acknowledge refreshes without critical modifications. Those days are no more. Gadgets today should be field update capable not exclusively to change and improve usefulness yet to determine future security issues.

4.1 Security Metric

A broadly accepted management principle in an embedded system is that an action cannot be directed if it cannot be identified. Metrics can be an efficient mechanism for security managers to separate the efficiency of different parts of their security programs, the security of a particular system, product, or services, and the capacity of departments within an enterprise to identify security problems for which they are in charge [16].

For instance, the security point of the secure communication set can be identified by utilizing equation 1,

$$L_{sc} = W_{ska} \ast L_{ska} + W_{pka} \,.\, L_{pka} + W_{ma} \ast L_{ma} \quad (1)$$

Where

$L_{ska,}$ L_{pka}, L_{ma}- security levels

W_{pka}, W_{ma}- are the weights

5. COUNTERMEASURE RESULT

Countermeasures must ensure that it ensures classification, uprightness, and accessibility at cost. The obstacle in the execution of the countermeasure is because of the imperatives attached to the implanted frameworks.

Those limitations are hitter, preparing force, and memory. As the installed frameworks are required to function as an independent gadget with explicit purposes, the requirement for high preparation isn't important. The expense and battery life obstruct the framework from executing very good quality crypto calculations and firewalls [12].

Thus, the assailants think that it's simple to misuse by beast power. Misconfiguration of a gadget may bargain with the trustworthiness framework. The firmware of the implanted framework must be refreshed and fixed appropriately. Conventions must actualize security all alone as opposed to needing to be constrained. Table I shows the countermeasures for every one of the assaults referenced previously [13]:

Table 1. Attacks and its Countermeasures

Type	Attack	Countermeasures
An attack based on software	Malware	Anti-malware application, Machine-learning-based Application [14]
	Brute force	Limiting the number of tries
	Buffer overflow	Hardware/Software Defender technique
	Web-based vulnerability	Sandboxing
An attack based on network	MITM	IPsec [15]
	DNS poisoning	DNSSEC
	Session hijacking	Encryption, disposable credits
	Signal jamming	Anti-jamming mechanism
An attack based on physical and side-channel	Power Analysis	Anti-jamming mechanism
	Timing Attacks	Data Masking technique
	Electromagnetic Analysis Attack	Shielding techniques

The greater part of the embedded systems is controlled by assets, for example, memory, preparing force, hitter, and cost. Thus, the execution of security becomes risky. Secrecy, honesty, and accessibility must be saved in the implanted framework when it is working. In late days, a considerable lot of the assailants focus on the implanted frameworks. For instance, the Mirai botnet significantly comprises of the implanted frameworks associated with the web.

6. CONCLUSION

Security is a significant necessity in developing embedded systems, particularly considering the availability that the vast majority of these frameworks give to framework organizations, private organizations, or the internet. Within the product and PC designing fields, the comprehension of security is basic to the effective development of present-day implanted frameworks. To effectively plan and develop present-day implanted frameworks, it is basic that PC and programming architects to comprehend security very well. In this paper, we have quickly analyzed security prerequisites and difficulties in the advancement of implanted frameworks. We have additionally analyzed how an inserted framework can be assaulted and their countermeasures are additionally analyzed.

REFERENCES

[1] Vivek Purohit, Garima Kothari, "A vital analysis on Integrating Security in Embedded Systems", International Journal of Engineering Research & Technology (IJERT), 172-174.

[2] Sztipanovits, J.; Karsai, G. Model-integrated computing, Computer Volume 30, Issue 4, April 1997Page(s):110 – 111

[3] Karsai, G., Sztipanovits, J., Ledeczi, A., Bapty, T.: "Model-Integrated Development of Embedded Software," Proceedings of the IEEE, Vol. 91, No.1., pp. 145-164, January 2003

[4] Ravi, Srivaths, et al. "Security in embedded systems: Design challenges." ACM Transactions on Embedded Computing Systems (TECS) 3.3 (2004): 461-491.

[5] Sri Parameswaran, Tilman Wolf, “Embedded systems security—an overview”, Des Autom Embed Syst (2008) 12: 173–183.

[6] Virus Information. Computer Security Resource Center, National Institute of Standards and Technology. Available at http://csrc.nist.gov/virus/Vulnerability notes database. CERT coordination center Available at http://www.kb.cert.org/vuls/

[7] M. Howard and D. LeBlanc. Writing Secure Code. Microsoft Press, 2001.

[8] https://www.iconlabs.com/prod/security-requirements-embedded-devices-%E2%80%93-what-really-needed

[9] Dacosta, Italo, et al. One-time cookies: Preventing session hijacking attacks with disposable credentials. Georgia Institute of Technology, 2011.

[10] Pajic, Miroslav, and Rahul Mangharam. "Anti-jamming for embedded wireless networks." 2009 International Conference on Information Processing in Sensor Networks. IEEE, 2009.

[11] Five Steps to Improving Security in Embedded Systems, WIND RIVER, 1-7.

[12] Gowthamaraj Rajendran, Ragul Nivash, “Security in the embedded system: Attacks and Countermeasures”, International Conference on Recent Trends in Computing, Communication and Networking Technologies (ICRTCCNT‟19), Kings Engineering College, October 18-19, 2019.

[13] Baheti, Radhakisan, and Helen Gill. "Cyber-physical systems." The impact of control technology 12.1 (2011): 161-166.

[14] Peikari, Cyrus. "Protection of embedded processing systems with a configurable, integrated, embedded firewall." U.S. Patent Application No. 10/346,956.

[15] “IPsec-Internet-Protocol-Security.” [Online]. Available:https://searchsecurity.techtarget.com/definition/IPsec-Internet-Protocol-Security

[16] Alberto Ferrante, Jelena Milosevic and Marija Janjuˇsevi´, “A Security-enhanced Design Methodology for Embedded Systems”, 39-50, Scitepress Science and Technology Publications, DOI: 10.5220/000450100039005