International Journal of Engineering and Advanced Research Technology (IJEART)

ISSN: 2454-9290, Volume-6, Issue-7, July 2020

An Overview on the Threats and Challenges to Hardware Security

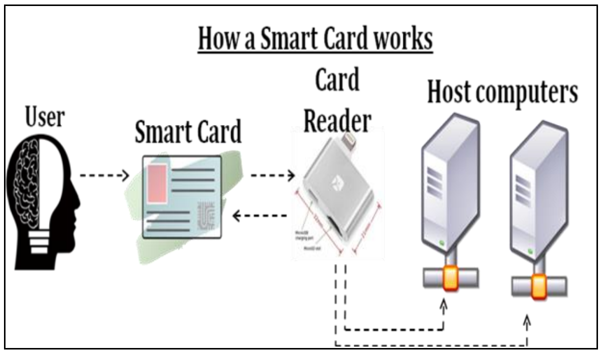
Roopali Jamwal, Talat Khan, Rachna Bharti, Vibhakar Mansotra

Abstract
Security should be intertwined with every part of system; the hardware is no exception. The interaction between hardware and software must be carefully planned. In doing so, the security of the entire system is strengthened. Hardware security is the protection of personnel, hardware, programs, networks, and data from physical circumstances and events that could cause serious losses or damage to an enterprise, agency, or institution. This includes protection from fire, natural disasters, burglary, theft, vandalism and terrorism.

I. INTRODUCTION

Hardware is the root of computation and communication. It is the enabler of any software, algorithm, or communication protocols. All the computation will eventually be carried out by hardware, namely the processor or the circuits. Nowadays, hardware becomes the enforcer for secure systems because it is used to ensure that only the authenticated user and software can access the processor. However, current hardware design flow does not have security as a key design objective. Biometric systems and smart cards are the only new hardware technologies that are widely impacting security. The most obvious use of biometrics for network security is for secure workstation logins for a workstation connected to a network. The cost of hardware devices is one thing that may lead to the widespread use of voice biometric security identification especially among companies and organizations on a low budget. . The main use of Biometric network security will be to replace the current password system.

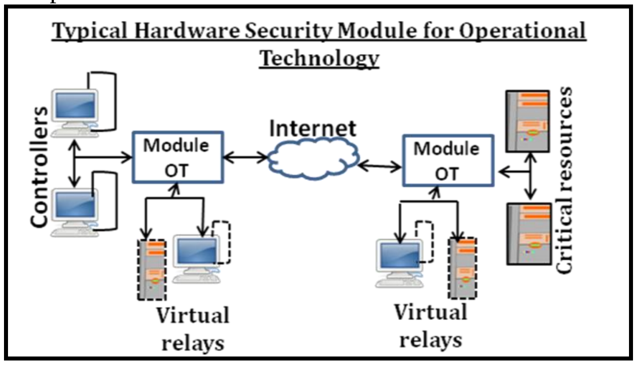
Maintaining password security can be a major task for even a small organization. Passwords have to be changed every few months and people forget their password or lock themselves out of the system by incorrectly entering their password repeatedly. Very often people write their password down and keep it near their computer. This of course completely undermines any effort at network security. Biometrics can replace this security identification method. The use of biometric identification stops this problem and while it may be expensive to set up at first, these devices save on administration and user assistance costs. Smart cards are usually a credit card sized digital electronic media. The card itself is designed to store encryption keys and other information used in authentication and other identification processes. The main idea behind smart cards is to provide undeniable proof of a user’s identity.



Smart cards can be used for everything from logging in to the network to providing secure Web communications and secure e-mail transactions. It may seem that smart cards are nothing more than a repository for storing passwords. Obviously, someone can easily steal a smart card from someone else. Fortunately, there are safety features built into smart cards to prevent someone from using a stolen card. Smart cards require anyone who is using them to enter a personal identification number (PIN) before they’ll be granted any level of access into the system. The PIN is similar to the PIN used by ATM machines. When a user inserts the smart card into the card reader, the smart card prompts the user for a PIN. This PIN was assigned to the user by the administrator at the time the administrator issued the card to the user. Because the PIN is short and purely numeric, the user should have no trouble remembering it and therefore would be unlikely to write the PIN down. But the interesting thing is what happens when the user inputs the PIN. The PIN is verified from inside the smart card. Because the PIN is never transmitted across the network, there’s absolutely no danger of it being intercepted. The main benefit, though, is that the PIN is useless without the smart card, and the smart card is useless without the PIN. There are other security issues of the smart card. The smart card is cost effective but not as secure as the biometric identification devices.

II. HARDWARE SECURITY MODULES & ABSTRACT KEY TYPES

Hardware security modules are needed for protecting cryptographic keys and security services from threats of information leakage. For that protection, the modules normally have the methods of password-based access control and message encryption. In several cases storing the long term cryptographic keys in a hard disk or even in memory poses a significant risk. Once the system they are stored is compromised the keys must be replaced as the secrecy of future sessions is no longer guaranteed. Moreover, past sessions that were not protected by a perfect forward secrecy offering ciphersuite are also to be assumed compromised.

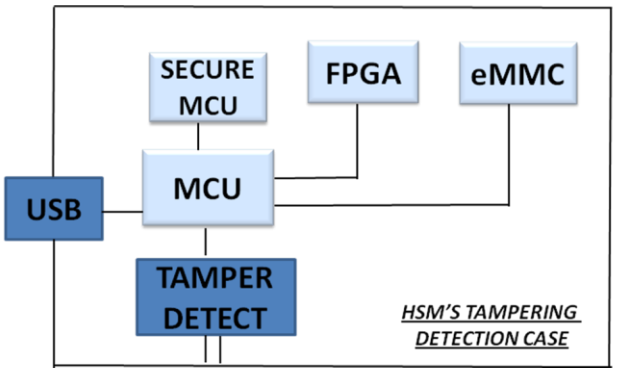


If such threats need to be addressed, then it may be wise storing the keys in a security module such as a smart card, an HSM or the TPM chip. Those modules ensure the protection of the cryptographic keys by only allowing operations on them and preventing their extraction. The purpose of the abstract key API is to provide an API that will allow the handle of keys in memory and files, as well as keys stored in such modules.

In GnuTLS the approach is to handle all keys transparently by the high level API, e.g., the API that loads a key or certificate from a file. The high-level API will accept URIs in addition to files that specify keys on an HSM or in TPM, and a callback function will be used to obtain any required keys. The URI format is defined in [*TPMURI*] and the standardized [*PKCS11URI*].

III. HARDWARE ATTACKS

- Manufacturing backdoors, for malware or other penetrative purposes; backdoors aren't limited to software and hardware, but they also affect embedded radio-frequency identification (RFID) chips and memory
- Eavesdropping by gaining access to protected memory without opening other hardware
- Inducing faults, causing the interruption of normal behavior
- Hardware modification tampering with invasive operations; hardware or jail broken software
- Backdoor creation; the presence of hidden methods for bypassing normal computer authentication systems
- Counterfeiting product assets that can produce extraordinary operations, and those made to gain malicious access to systems



Hardware attacks pertain to the following devices:

- Access control systems such as authentication tokens
- Network appliances
- Industrial control systems
- Surveillance systems
- Components of communication infrastructure

Attackers could also act at lower levels to affect the work of microcircuits, fundamental components of any electronic device. Recently researchers have explored the possibility of modifying hardware behavior by managing the concentration of dopant in electronic components or altering its polarity.

Scientists provided further ideas of types of hardware backdoors:

- **<u>Ticking time bombs</u>–** An attacker could program a time bomb backdoor into HDL code that automatically triggers backdoors after a pre-determined fixed amount of time after the power-on of a device. A device could be forced to crash or operate maliciously after a determined number of clock cycles. It's clear that this type of attack could be very dangerous. An attacker could design a kill switch function that could be undetectable by any validation methods.
- **<u>Cheat codes</u>**– An attacker could program backdoor triggers based on specific input data, otherwise known as "cheat codes." A "cheat code" is secret data that the attacker uses to identify themselves to hardware backdoor logic. It'll then initiate a malicious operation mode. Of course, the code must be unique to avoid being accidentally provided during validation tests. As opposed to time bombs, this kind of backdoor needs a second attack vector, the "cheat code." The attacker could provide "cheat codes" which send a single data value containing the entire code (single-shot "cheat codes") or a large cheat code in multiple pieces (sequential "cheat codes.")

IV. PREVENTION

The best way to prevent the insertion of hardware backdoors is to tightly control the entire production process. Use a trusted design team, use component design that's free of backdoors and release it to a trusted foundry. Trusted people, clean production environments and self made tools provide assurance that products are free of backdoors.

Prevention could be implemented in different phases of production:

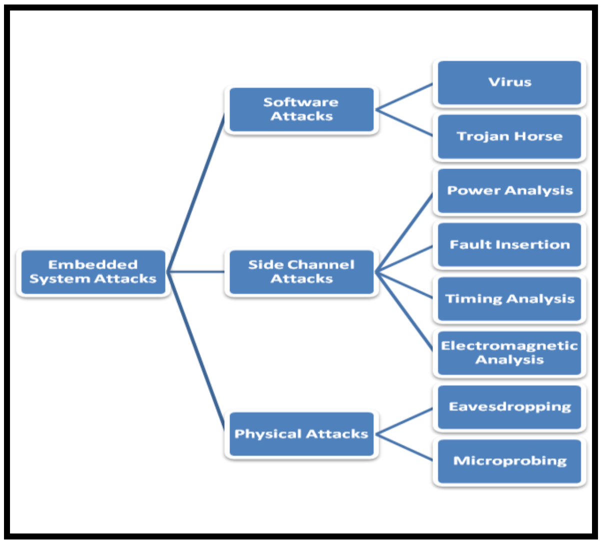
- The design level- the ability to create trusted circuits using unreliable EDA tools is the primary goal for detection at this stage. Principal solutions fully account for the use of all hardware resources, leaving no time frame for the execution of malicious features.
- The fabrication level provides both hardware specifications and a list of "security-related properties." Customers and manufactures must agree how to turn these concepts into a formal mathematical codification procedure. The IP producer writes the Hardware Description Language (HDL), they also produce evidence that the specified

hardware fulfills all requirements. That can then be checked by a theorem, proven when the IP is delivered to the consumer.

- The post-fabrication level– to cut down the attacker's window of opportunity, reconfigurable logic could be placed between the output of some ICs and the input of other ICs, disguising some of the design from an attacker who has access to the Register Transfer Level.

V. ATTACK ON EMBEDDED SYSTEMS AND PORTABLE HARDWARE

While there are many kinds of computer hardware that use cryptographic processing, embedded systems and portable hardware pose some unique challenges.



Consider the following two examples of hardware which have been the targets of implementation attacks:

Smartcards. Thin credit card-like cards with embedded ICs. The cards do not carry their own power source, as the contacts on the card allow the card readers (ATMs, pay telephones, Points of Sale) to both power and communicate with the card. The cards typically have sensitive information such as private keys in non-volatile storage, and communicate with a card reader using standard protocols to encrypt and authenticate.

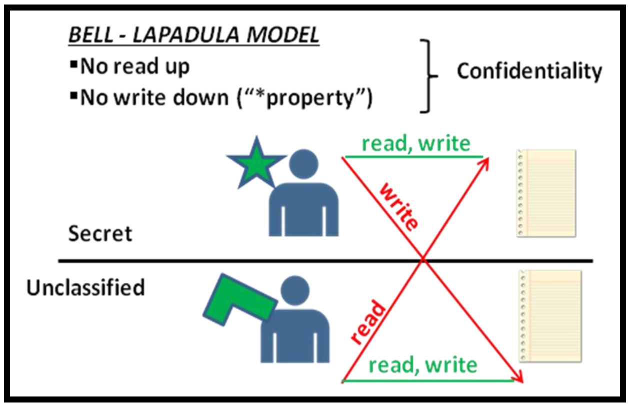
Cell phones and PDAs. These devices have more computational power and wireless communications capabilities. In order to obtain network service, they must authenticate securely over an insecure and easily manipulated channel. The devices often store and communicate private information belonging to consumers, service providers and manufacturers.

The noteworthy aspect of security as it relates to embedded systems and portable hardware is the extremely hostile environment in which the hardware is used.

Security Models

Security for an information system is typically defined in terms of a security model, which is both an abstraction of a class of information systems and a characterization of what ''security'' means for that class of systems. The model may be very abstract (e.g., the machine model used in the definition of non interference) or be fairly concrete and include specific security control mechanisms (e.g., the Bell and LaPadula model). At base, a security model is simply a specification of an information system (or class of systems) and its security properties. What we are calling models here are no different than what are called specifications in the conventional software engineering process—and this is an important point: security models are nothing more than (usually formal) requirements statements and specifications for the security properties of a system design.

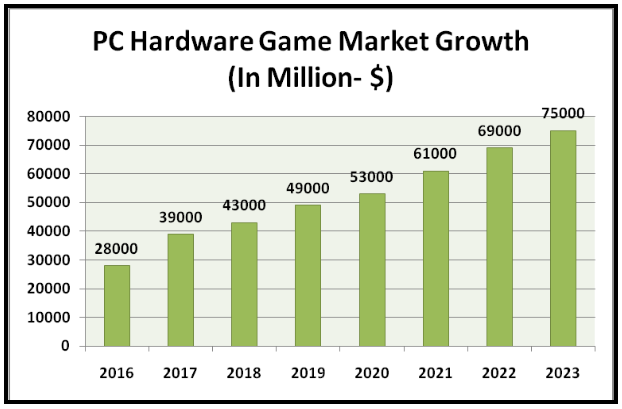
Types of Security Models



The *Bell and LaPadula* model is an access control model that incorporates a number of rules designed to restrict the flow of information. Among these rules are the simple security property often summarized, respectively, as follows: a subject may only read an object at a level at or below his own level (in the security lattice), and a subject may only write to an object at or above his own level. Variants of the Bell and LaPadula model have been widely used in secure system development despite the fact that the model does not preclude some types of intuitively un-secure behavior (covert channels). Information flow models tend to be more abstract than access control models. Rather than define security in terms of the permissible accesses of subjects, they define security in terms of the absence of certain prohibited information flows, without regard for the particular mechanism through which these flows may occur. Thus, un-secure information flow is proscribed, even if no action of any subject violates the access control rules of the system. Several of the most widely studied security models are noninterference and non-deducibility.

Noninterference is an information flow model. Subject **a** is non-interfering with subject **b** if no action of **a** can have any effect on subsequent actions of **b**. Any specific noninterference policy specifies pairs of subjects that must be non-interfering. A noninterference policy tends to be stronger than an access control policy since it precludes information flowing through mechanisms that are not easily captured in the framework of subject-object access. Noninterference is limited to deterministic systems. It has been used as the security model for at least one large system development effort.

Non-deducibility is a strengthening of noninterference suited to non-deterministic systems. The key idea is the following. Assume that subject **a** is prohibited by the policy from passing information to subject **b**. The system is non-deducibility secure if any possible set of observations of the system by **b** is consistent with any possible set of actions of **a**. Daryl McCullough has proposed a variant called restrictiveness that has the so-called ‘‘hook-up’’ property—two restrictive systems can be combined to yield a restrictive system. This model of security is used in the Ulysses (later called Romulus) system of Odyssey Research Associates.



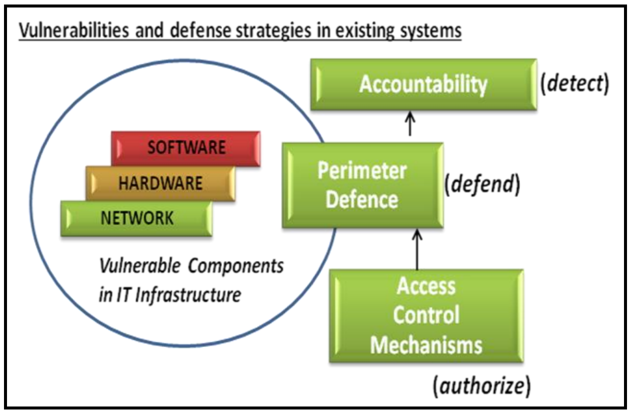
How Hardware Facilitates Security Attacks

The entire PC platform operation relies on the orchestration of multiple hardware elements in order to achieve the platform operational goals. Each piece of hardware brings something different to the party, and the security relevance of that piece of hardware depends upon its overall native role in the PC platform. For example:

- The hardware may have direct capabilities to affect a critical system resource (e.g. DMA to system/software memory)
- The hardware may have indirect sideband access to a resource (e.g. PCI cards typically have access to an SMBus segment)
- The hardware may store arbitrary software executable code that can be automatically invoked (e.g. HDD or USB drive; PCI device option ROM)
- The hardware may proxy data from an untrusted external source (e.g. NICs , Wifi radios)

VI. OBTAINING HARDWARE ACCESS

Since we are looking at hardware-involved security attacks on software, we need to characterize how an attacker will first obtain necessary access to the hardware to achieve the intermediary step of the attack. Hardware access can be realized in a number of ways:

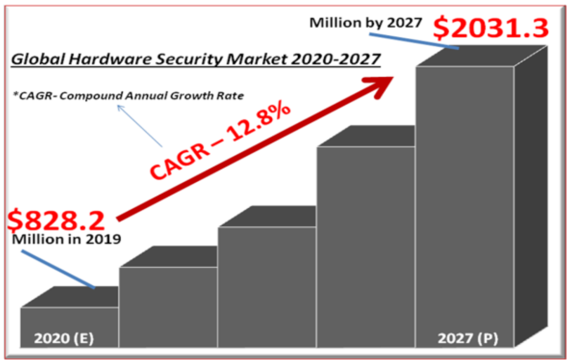

Vulnerabilities and defense strategies in existing systems
SOFTWARE
HARDWARE
NETWORK
Vulnerable Components in IT Infrastructure
Accountability
(detect)
Perimeter Defence
(defend)
Access Control Mechanisms
(authorize)

1. **Mistakenly passed through by a higher privilege software layer**.
A higher-privileged software layer may attempt to provide a controlled or limited access to hardware, but wind up being overly-permissive; or the hardware it allowed access to has additional, unrecognized functionality. In other situations, the hardware access functionality provided by the privileged software layer may be a remnant of non-production debugging needs, etc
2. **Explicitly passed through by a higher privilege software layer**.
Many hardware devices exist with intent they be accessible and utilized by the local user, who typically is running in the application privilege layer. Graphics is a great example: GPGPU workloads, DirectX shaders, and OpenCL kernels originate at the app layers and are passed through to the graphics hardware for GPGPU interpretation & execution. Elsewhere, user-mode driver frameworks, particular in the USB device arena, allow flexibility by the OS to offload select arbitrary device handling to user-mode applications. In all of these situations, the OS layer is allowing access to a select portion of hardware to facilitate the management & use of that hardware by the application layer.
3. **Explicitly provided by hardware architectural intent.**
Hardware assisted virtualization technologies like Intel VT-x, AMD-V, EPT, VT-d /AMDV/IOMMU, and SR-IOV facilitate direct hardware access (e.g. access by VM guests in a virtualized environment, to the benefit of the hypervisor). These technologies get leveraged in VMM product features such as Xen PCI passthrough, Xen VGA passthrough, and VMWare VMDirectPath I/O . Similarly but on a more conceptual level, application (ring 3) software is allowed to directly execute many instructions on

the CPU without OS (ring 0) involvement. Overall a VMM or OS may utilize hardware architecture to provide bounded access to less privileged software in a controlled manner; the assumption is that it is safe to do so.

4. **The attacker is already deemed to have access.**
 Particular to SMM/BIOS software layer, SMM is simply not in a strong architectural position to gate all hardware access by the software layer above it; therefore it must always operate with consideration that system resources are shared with a potentially unreliable or compromised OS/hypervisor layer. Hardware assisted virtualization technologies also provide hypervisors with the ability to only intercept a subset of hardware access and CPU instructions originating from a VM guest; certain VM guest hardware operations simply do not have a corresponding VMM trap/exit available for the hypervisor to leverage.

5. **The attacker is physically proximate to the system.**
Physical possession or access to a PC system allows for various hardware tampering attacks (e.g. Evil Maid, cold boot, hardware keylogger) and use of externally exposed hardware capabilities (e.g. Firewire DMA). Physical proximity is sufficient for attacks using radio hardware (e.g. Wifi, LTE/Wimax, Bluetooth, GSM/cellular) as the entry point into the system. Looking beyond PCs for a moment, the community already has seen many instances of embedded system "jailbreaking" and game console hacks where physical access to the device was leveraged to achieve a software advantage of some sort.



VII. CONCLUSION

Hardware security has become a topic of deep interest for the people associated with this domain. However, the scope of research on hardware security has never been defined clearly. In this paper various key topics pertaining hardware security has been dealt in detail. The research efforts are also reflective of future trends in hardware security. We just hope that more researchers come forward to develop ideas and solutions in order to curb the threats and issues related to hardware security.

REFERENCES

http://resources.infosecinstitute.com/hardware-attacks-backdoors-and-electronic-component-qualification
http://www.berkes.ca/archive/berkes_hardware_attacks.pdf
http://forristal.com/material/Forristal_Hardware_Involved_Software_Attacks.pdf

Shailja Pandey, "Modern Network Security: issues and challenges" , ISSN : 0975-5462 vol. 3 no. 5 may 2011

Atul Kahate, "Cryptography and Network Security", ISBN-13: 978-0-07-064823-9
Joseph Migga Kizza, "A guide to computer network security", ISBN: 978-1-84800-916-5 (Print)

Cisco 2014 - Annual Security Report available at "www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf"

Christopher Leidigh, "Fundamental Principles of Network Security" available at http://www.apcdistributors.com/white-papers/Management%20Systems/WP-101%20Fundamental%20Principles%20of%20Network%20Security.pdf

Siddharth Ghansela, "Network Security: Attacks, Tools and Techniques", Volume 3, Issue 6, June 2013 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

Advancing Towards a Securer Cyberspace by DSCI (NASCOM)

Roopali Jamwal, *Talat Khan, **Rachna Bharti, **, ***Vibhakar Mansotra
Department of Computer Application
GCW Parade (J&K), *Govt. Degree College, Nagrota, **Govt. Degree College, Udhampur, **, ***University of Jammu.