APTIKOM Journal on Computer Science and Information Technologies
Vol. 4, No. 2, 2019, pp. 57~60
ISSN: 2528-2417
DOI: 10.11591/APTIKOM.J.CSIT.34

A secure and reliable method to protect usb data

Alycia Sebastian*[1], K. Siva Sankar[2]
[1]Research Scholar, Department of Computer Science & Engineering, Noorul Islam University
Thuckalay, TamilNadu, India
[2]Department of Information Technology, Noorul Islam University
Thuckalay, TamilNadu, India
*Corresponding author, e-mail: alycia.sebastian@gmail.com[1]

Abstract

With technology advancement people have started using different type of memory devices for storing data and keeping it secure has become concern in today's world. Universal Serial Bus (USB) flash drives are leading portable storage device for storage and easy transfer of data from one computer to another. The usage of USB has grown exponentially and without security the data on the disk is at risk. Nowadays USB manufacturers offer password protection and fingerprint authentication to secure the USB data. In this paper, USB is devised as a highly secured portable boot medium with fingerprint authentication to secure the data.

Keywords*: boot loader, fingerprint authentication, operating system, security universal serial bus (usb)*

1. Introduction

USB is a small, portable storage medium whose size ranges from 2GB and up to 2TB are available for end users for data storage. Users can freely move data and securing the data from unwanted access is the key concern. To secure data, USB offers software based protection and hardware based protection. Biometric USB drive with in-built fingerprint scanner and password protection provides the maximum security for the user's information. So many solutions have been provided for securing USB data against unauthorized access. But, there is a possibility to access information when used with computer system by evading the fingerprint authentication thereby providing a serious threat to the user information in USB. So there is a need to do authentication before loading the OS.

Boot Loader is responsible for initializing and loading OS into RAM. In single stage, boot loader executes the kernel directly from the boot sector. In multi stage, boot loader follows the below steps.

Stage 1: boot. img

- Code is stored in the first 466 bytes of MBR.
- BIOS loads stage 1 to the address 0x7C00.
- Uses LBA48 addressing to point to stage 1.5.
- Loads stage 1.5 in to the memory at address 0x2200.
- Jump to entry point of Stage1.5.

Stage 1.5: core. img

- Stored in the 63 bytes after MBR.
- Contains file system specific code to find the operating system image on the "boot" file system.
- Since it contains file system drivers it can load stage 2 by directly specifying file path and file name.
- Loads the stage 2 into the memory at address 0x8000.

Stage 2:

- Initializes the grub shell and displays the menu and command environment.
- Appropriate selected kernel image is loaded into memory.
- Boot kernel.

Received May 9, 2019; Revised June 3, 2019; Accepted July 10, 2019

Security of the system starts from the boot process. In this paper, USB is developed as a boot medium and the fingerprint authentication is done upon the boot process. On successful authorization, the operating system is loaded into the memory from the USB. This method provides a secure, reliable and portable device which can be used in any system with much required protection for the USB data.

Dynamic boot loader in normal boot loaders to boot from USB, user has to change BIOS settings to USB boot which requires user knowledge of BIOS settings. These types of boot loaders are static and are dependent on BIOS. The concept of dynamic boot loader is that it automatically finds the USB boot medium and load OS without user intervention [1]. This eliminates BIOS dependency, thus providing a user friendly system and significantly reducing the time it takes in changing the BIOS settings. The proposed design on fingerprint security for USB data can be integrated with dynamic boot loader. The integration provides a user friendly, secure and highly portable USB system.

2. Related Work

Much research work has been successfully done in safeguarding USB information from unauthorized access but setting a safe environment is more important in protecting the data. Biometric authentication has since become popular because of successful attacks on password and PIN's. In USB token fingerprint authentication system, the patterns are stored in USB token but recognition is done in the computer system which makes the entire authentication system vulnerable to attack. This disadvantage can be overcome by performing fingerprint matching inside the USB token system [2].

USB with Fingerprint authentication has an integrated fingerprint scanner and a private drive. User can enroll the fingerprint using the host system. On plug in, it runs the fingerprint program and on successful authorization, it opens the private partition where the user can store data. In paper [3], the authors successfully bypassed the fingerprint authentication by making binary code modification in .dll file. Also they developed a program that retrieves the fingerprint reference templates from the drive which poses serious security threat to user data in the USB.

The Universal Serial Bus (USB) is a mass storage device which is vulnerable to attacks. In paper [4], the authors have proposed a control algorithm for mutual authentication between host and USB to protect USB documents. User has to authenticate himself with user name and password and a session key is generated based on the username and ID of the USB device. The generated key is then used to encrypt the files being stored in the USB device.

USB is the most popular portable storage device. Once it is lost, the information in the USB is prone to theft. In paper [5], the authors have discussed the various ways of securing USB memories while analyzing its vulnerabilities [6]. The approaches to secure USB can be categorized under software only approach, hardware supported partitioning approach and Hardware based encryption approach.

In general, to secure data in USB, it has to be protected either by software or hardware or by both. User authentication method like password, fingerprint are for access control to the data stored in USB and encryption and decryption are for protecting data transfer between host and USB device.

3. Proposed Design

Different approaches are in use to secure USB device from unauthorized access. The security on stored data becomes meaningless when secure USB is used with a hostile host system. The attackers can bypass the fingerprint authentication system and access the stored data. To eliminate this type of attack, the paper discusses a plug and play Live USB where fingerprint authentication is done before the OS is loaded from USB to RAM. This method guarantees a safe, secure and reliable portable system for user data. So now the user can store highly confidential information on his live USB and use in any host system which supports USB booting.

The section 3.1 discusses the booting process of the proposed method and section 3.2 shows the fingerprint secured USB architecture.

3.1. Booting Process

a. After power on, the user changes the BIOS setting to boot from USB. Here the dynamic boot loader concept can be used to automatically identify the Live USB.
b. The boot loader partition is made read only so that BIOs can detect it at boot time.
c. The boot loader in USB is modified to call the fingerprint authentication program at a specified address using LBA48 addressing.

d. The fingerprint software prompts the user to scan the fingerprint. The fingerprint of the user are scanned and stored before the booting process using the host system. The software matches the fingerprint and gives access to the secured partition if the authorization is successful.

The Figure 1 shows diagram depicts the functional design of the booting process of the proposed design.

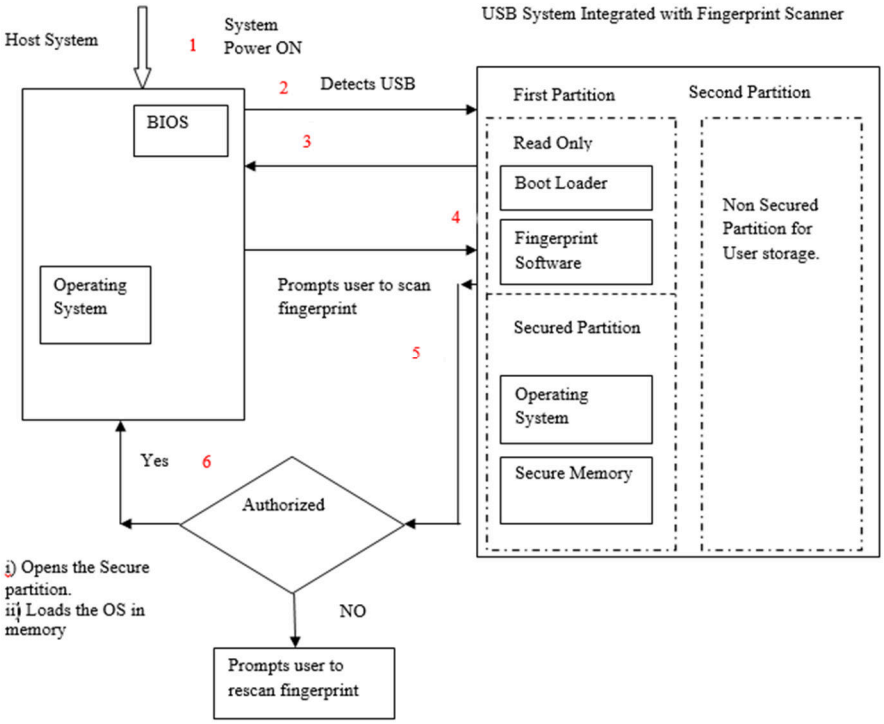

Figure 1. Booting from USB with fingerprint authentication

3.2. USB Architecture

The bootable USB used comes with fingerprint scanner integrated along with it. A 32 GB USB is used. The USB is divided in to two partitions using the repartition tool. The first partition contains the boot loader, the fingerprint software and the operating system. The operating system and remaining storage is secured using fingerprint. Initially only boot loader and fingerprint software are visible and is read only for BIOS to detect the USB at boot time. The second partition is public partition for the user storage.

The Figure 2 shows diagram the architecture design of USB.

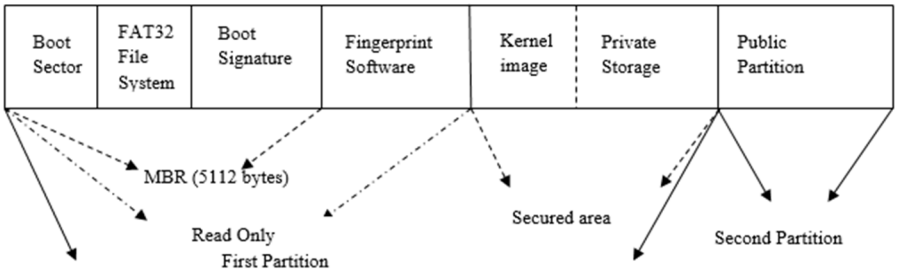

Figure 2. USB architecture

The USB first partition contains the Master boot record (MBR) which has the boot loader, FAT32 file system, partition table and Disk signature [7] and finger print software and the remaining sectors are made private. The FAT32 file system is used so that the USB can be used in almost all system that supports USB. The USB is hardware encrypted. The OS and the private sectors are protected with fingerprint.

The USB can be made bootable for Linux based system or windows based system. For Linux based system, Unetbootin can be used to make the USB bootable with Linux distributions [8]. The USB is formatted with FAT32 file system for Linux bootable.

For Windows based system, the USB is formatted with NTFS file system. The USB is made bootable using the tool wintoUSB. This tool allows installing the windows onto a USB drive [9]. For windows installation requires higher capacity USB drive and takes longer time for installation. The second partition is public partition which is available for normal data storage. The user can use the USB as normal data storage medium.

The USB can hold up to 10 fingerprint images. Once the finger print program starts running, the user is prompted to swipe his finger across the finger print scanner. The authorization process matches the fingerprint pattern against each of the stored pattern. Once the fingerprint matches, then the secured partition is opened and the kernel image is loaded into RAM of the host machine. Once authentication is done the first partition is now available for the user to access the files from the protected partition. The authentication is done before the OS loading and hence the OS is not involved in the fingerprint verification process. So the coding is done in Assembly language.

All the fingerprint operation is done inside the USB without involving the host system. This eliminates the need to transfer fingerprint images between host and USB, thus securing the pattern. The USB can be used in any machine irrespective of the OS installed in that system.

3.3. Performance Analysis:

Most of the solutions for securing the USB data discussed of how to secure the data in stored format. But when the protected USB drive is used in a host environment, the transfer of data between host system and USB makes the data vulnerable for attacks. Since the OS runs from the USB and not dependent on the operating system of the machine, the proposed design achieves the security for the data stored as well as make the environment secure. The kernel image can be customized as per the requirement of the user. This will greatly improve the loading time of the OS to the host machine.

4. Conclusion

The USB devised as bootable medium with integrated fingerprint authentication provides a highly portable and secure USB system which can guarantee confidentiality for the information stored on the USB. Since the OS can be customized as per the need of the user, the loading time is much reduced.

Integrating the concept of dynamic boot loader which automatically identifies the Live USB, with fingerprint secured USB will provide a user friendly environment to the user thereby both securing the data and giving a much needed secure and user friendly environment to the user.

References

[1] Alycia Sebastian, Dr. K. Siva Sankar. Design of a Boot Loader for Operating System. *Australian Journal of Basic and Applied Sciences*. 2015; Vol. 9(2): 368-374.
[2] Daesung Moon, Youn hee Gil, Dosung Ahn, Sung Bum Pan, Yongwha Chung and Chee Hamg Park. *Fingerprint Based Authentication for USB Token Systems*. Information security Applications. Volume 2908 of the series Lecture Notes in Computer Science, 2003, page: 355-364.
[3] Benjamin Rodes, Xunhua Wang, *Proceedings of the 26th Annual Computer Security Applications Conference*. 2010: pp 89-96.
[4] Mr. A. N. Magdum, Dr. Y. M. Patil, A. Secure Data Transfer Algorithm for USB Mass Storage Devices to Protect Documents. *International Journal of Emerging Engineering Research and Technology*. 2014; Vol. 2(4): 78-84.
[5] Kyungroul Lee, Hyeungjun Yeuk, Youngtae Choi, Sitha Pho, Ilsun You, Kangbin Yim, Safe Authentication Protocol for Secure USB Memories. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 2010; Vol. 1(1): 46-55.
[6] Jaein Kim, Youngjun Lee, Kyungroul Lee, Taeyoung Jung, Dmitry Volokhov, and Kangbin Yim, Vulnerability to Flash Controller for Secure USB Drives. *Journal of Internet Services and Information Security (JISIS)*, Vol. 3(3/4): 136-145.
[7] Jan Axelson, *USB Mass Storage: Designing and Programming Devices and Embedded Hosts*. Lakeview Research LLC, 2006.
[8] https://unetbootin.github.io/.
[9] http://www.easyuefi.com/wintousb/.