Journal of Engineering and Applied Sciences 14 (19): 7098-7104, 2019
ISSN: 1816-949X

Design and Implementation of Input/Output Port Blocker System to Thwart Input/Output Attacks

Marwah Q. Abbas and Alaa M. Abdul-Hadi
Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq
marwah.qassim@yahoo.com

Abstract: Computer systems may store temporarily or permanently vital information to individuals or organizations, so, it can be subjected to different types of attacks. Attackers can exploit existing vulnerabilities to reach and extract targeted information via. one of the attack points represented by the Input/Output (I/O) ports, such as the Universal Serial Bus (USB). This study proposes a security solution to thwart malicious I/O devices from launching I/O attacks against the computer using hardware/software system called Input/Output Port Blocker (IOPB) that is based on a two-factor user authentication technique. The main idea of protecting the computer system from I/O attacks is to add a security layer to isolate the I/O ports from the computer system. In this layer, the required security policy can be applied using different security controls such as user authentication, access control, anti-malware and encryption. This research applies to all I/O ports but as a case study, protecting computers from I/O attacks arising from the USB ports are considered. It also describes the proposed IOPB system design and provides a detailed implementation using C# programming language, Raspberry Pi3 Model B, a relay switch and fingerprint reader U.are.U 4500. Finally, it studies the effect of activating the IOPB on data transmission speed between the USB-based storage device and host considering image, voice and video. The obtained results from the transmission of multimedia to the computer system show that the effect of adding the IOPB increases the delay approximately by 2.5, 2 and 3.5%, respectively. This effect can be considered negligible and has no significant effect on the system performance compared to the protection offered by the IOPB.

Key words: I/O attacks, USB protection, Raspberry Pi, user authentication, arising, security

INTRODUCTION

It's known for a long time that an unknown USB drive should never be plugged into a computer system as it may be loaded with malicious software (Nissim *et al*., 2017). The main goal of this study is to protect the computer system from the attacks coming from the USB port using a median security subsystem. I/O attacks can compromise the security of any computer in several ways by exploiting the confidentiality, integrity and availability of the data inside the system (Patric, 2014). USB is a popular option of interfacing with computer peripherals, this great convenience comes with a danger that can permit a device to achieve arbitrary actions at any time while it is plugged into it, since, the USB is for plug-and-play devices (Myung, 2015).

This undetectable threat differs from a traditional virus that spreads via. USB devices, due to its location and the way it works, so, it is unacceptable for antivirus to find it (Myung, 2015). Many researchers solved I/O attacks in several ways like using hardware or software to encrypt the memory content and so on. Existing methods for protecting computer systems from I/O attacks can hardly be adopted due to complex hardware use like Field Programmable Gate Array (FPGA) and high cost to encrypt the main memory (Henson, 2014). And other solution like Input Output Memory Management Unit (IOMMU) from the company does not support every motherboard (Markuze *et al*., 2016).

This research paper discusses the design and implementation of a security system called IOPB. It's hardware consists of Raspberry Pi3 Model B that acts as a controller between the USB port and the computer system, a user authentication subsystem and a relay switch to make all the USB ports off, until the user successfully passes the authentication process. The authentication occurs in two-levels first, the user enters the PIN, then the fingerprint reader scan his fingerprint, if both are correct, then the Raspberry Pi sends a signal to the relay switch to make the required port on. For user authentication the fingerprint reader from U.are.U 4500 is used.

Corresponding Author: Marwah Q. Abbas, Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq, marwah.qassim@yahoo.com

Literature review: The following is a brief description and features of I/O attacks and protection methods achieved in previous years. Boileau (2006), described the attack of DMA under various operating systems. To mitigate this type of attack, he suggested to turn off the firewire port or just enable the password on G5 Macs or makes the OS OHCI controller drivers disabled. Loic, described the vulnerabilities of networks, they discovered that if the attacker controls the network card then he can do any type of attacks like stop processing packets, drop some packets ARP/DNS cache poisoning, implement SSLstrip-like attacks, attack hosts on the LAN replace the firmware, attack the host (read/write access to the main memory).

Patric (2014), studied the hidden attacks on a computer system and implemented a system called Dagger to protect against these attacks. He discovered that DMA attacks are very effective and cannot be found by anti-virus programs, thus, need more security mechanisms to protect from different types of attacks.

Myung (2015), proposed a new hardware/software solution to protect the USB port in the computer system, the system developed called USB wall. They develop a middle device to make the user to find the risk in the computer system. To test the programmed device they make simulation between USB device and malicious device to find the risk.

Liao *et al.* (2018), proposed a software solution to design and implement a secure USB flash drive. They use a chaos authentication with non-linear encryption and decryption function and use a smartphone as master (client) and micro-controller as a slave (server) and use the Bluetooth module to connect between client and server. For safely access data to flash the user activated the app in a smartphone (client) then the micro-controller (server) allows the computer to access the data on the USB flash.

Summing up, researchers were focusing on solutions to protect the computer system from I/O attacks exploiting specific I/O ports. This way called input/output port blocker that blocks the USB port in the computer and when the user authentication occurs the USB port turn on. IOPB that focus on hardware/software solutions with user authentication technique.

MATERIALS AND METHODS

The architectural design of IOPB: The input-output port blocker consists of three blocks connected as shown in Fig. 1. These blocks are the control system which acts as a proxy that receives a signal from the computer system and send a signal to the switch. The other block is the

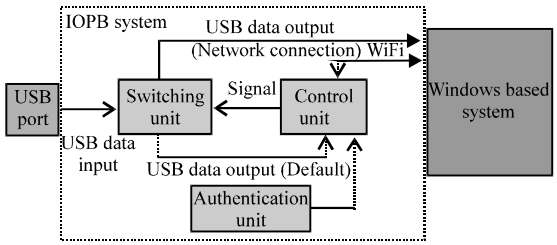

Fig. 1: The IOPB block diagram

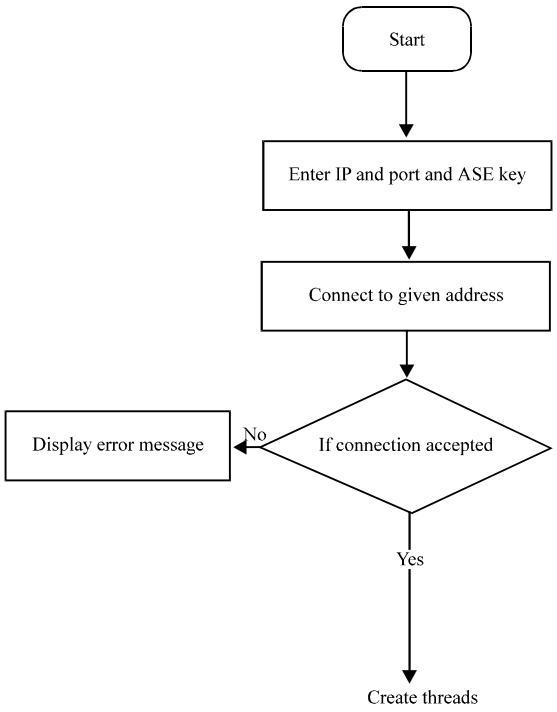

Fig. 2: Basic functionality of IOPB

(Fig. 2 and 3) switch that receives a signal from the control system to turn the USB port ON. The last block is the authentication system when the authentication occurs, the control system can send a signal to switch to turn the USB port ON.

Basic IOPB functionality: Initially when the user switch on his computer system with windows operating system all USB ports are OFF and any USB flash inserted will not be activated. The user login to the computer using his own password. Then the IOPB will operate automatically and the user must enter its fingerprint using a fingerprint scanner (U. are. U 4500) and PIN for more security, since, IOPB is designed using two levels of authentication. If the fingerprint matches the one stored in the database the system will be connected to the main system controller which is the Raspberry Pi (server) and wait for user input.

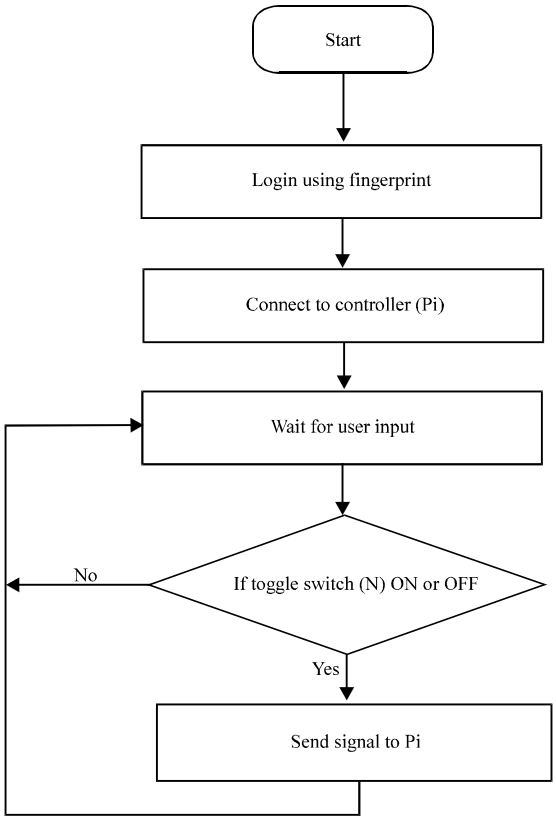

Fig. 3: Data flow of the connection

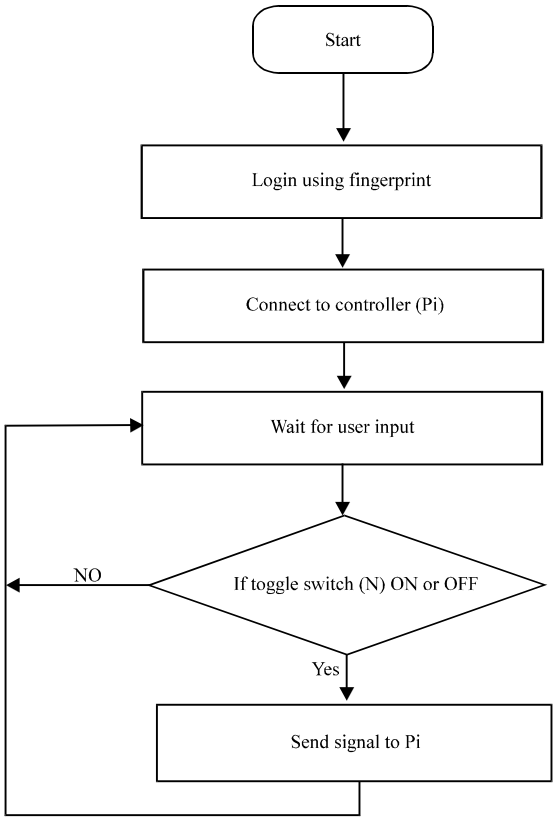

Fig. 4: Send signal thread

If the user inserts any device ports, then the controller sends a signal to the relay switch to make the required USB port ON. If the port is not used (idle) for a few minutes (5 min in the program) the port will transit to the OFF state again. Figure 4 shows the flowchart of the basic functionality of IOPB. The following devices are used instead of the block diagram in Fig. 1. Figure 2 shows the basic functionality of the IOPB system:

- Raspberry Pi 3 Model B act as a control system
- Relay switch as a switch
- Fingerprint reader (U. are. U 4500) as an authentication system

The connection among IOPB components: The connection between the controller (Raspberry Pi) and the main computer is generated using team viewer (Fig. 3). The administrator enters the IP address, port number and encryption key to establish a connection between the client (PC) and the server (Raspberry Pi). If the connection to the given address is not accepted, the program displays an error message and if it is accepted the program generates three actions (threads): send signal thread, receive signal thread and check threa d. Figure 3 shows the data flow for the connection. The remaining three threads are shown in the next three flowcharts Fig. 4- 6.

IOPB components: Before going to the implementation of IOPB and other details, it is necessary to show the selection items required to implement the system. The system consists of the following hardware and software components:

Hardware components:

- Raspberry Pi3 Model B
- Relay switch
- GPIO interface sockets
- Fingerprint reader
- Vero board
- External USB ports

Software components:

- Ubuntu MATE Linux operating system for Raspberry Pi
- Team viewer
- Visual studio.NET

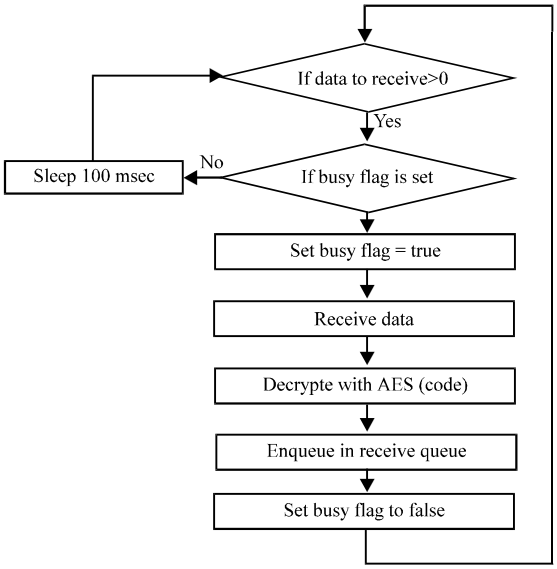

Fig. 5: Receive signal thread

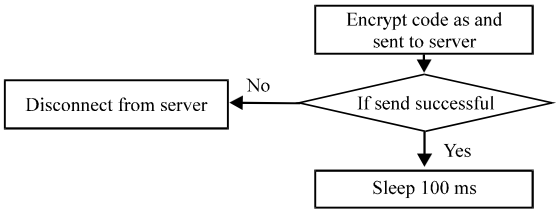

Fig. 6: Checks signal thread

The reason for choosing these components and the benefits of using Raspberry Pi and GPIO socket. Raspberry Pi is a stand-alone controlling system and has a low cost compared with the other controlling card. In addition, Raspberry Pi has many ports to add any external hardware in another world, it is expendable (Vujovic and Maksimovic, 2015):

- Raspberry Pi is portable in other worlds, it has a small size, so, it is easy to setup
- Raspberry Pi is a full minicomputer system (Vujovic and Maksimovic, 2015) and in the IOPB used as a server
- Raspberry Pi has an open source Ubuntu MATE (Teamviewer)
- Raspberry Pi 3 has many cores (quad-core CPU) (Anonymous, 2018), so, it can run multi threads without error and with full performance

C sharp programing language is chosen because it is cross-platform, easy to use and object-oriented. Ubuntu MATE is an open source that depends on Debian

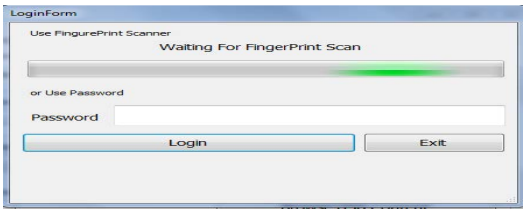

Fig. 7: Windows for user authentication (fingerprint and PIN)

architecture (Anonymous, 2018). Fingerprint reader it's a USB peripheral for the different desktop and it extracts excellent image quality for the finger and stored in the database. When the user puts his finger on the reader window. The reader automatically captures the image of the finger and encrypts the fingerprint image before sending. Compared with other type U.are.U is easy to use and have security on board (Anonymous, 2016, 2017) (Fig. 7).

IOPB system implementation: The implementation of IOPB consists of two sides, client-side represented by the computer system and server side represented by Raspberry Pi. The following sub-section will describe each of them.

Server side: Server side is implemented using Raspberry Pi as hardware. Raspberry Pi connecting with the computer using a wireless connection and hotspot network. This connection is the most dangerous because the attacker can cut it to steal the information transmitted from the USB flash to the computer, so, it is encrypted using Advanced Encryption Standard (AES) algorithm. AES is the most common algorithm used because it is difficulty breaking and also it is a symmetric block cipher (Ako, 2017). When the user is authenticated Raspberry Pi send signal to relay switch, making the USB port ON then the user can use the USB port as any normal port. Server side programmed in teamviewer using open source Ubuntu MATE. Teamviewer acts a virtual computer system. To run the main program the following steps are followed:

- Right click on the desktop in teamviewer
- From the list, open in terminal is selected

After pressing the (open in terminal) another window will appear, sudo monodevelop is typed and the connection password (1234) is entered to open the server program. When the program is run, the window in Fig. 8 will appear that means it waits for the client to make the connection.

Table 1:Main form fields description

Fields	Descriptions
AES encryption key	The encryption key for the AES encryption algorithm
IP Address	The IP address for network connection. The default is 192.168.43.6
Sleep timer	When this time reaches zero the system will turn off automatically and the lock screen for the computer system appears
Connect to server	This button is pressed to connect the client to server
Disconnect	This button is pressed to disconnect the client from the server
Lock	This button is used to turn off the IOPB and request for fingerprint and PIN again
Refresh (USBN)	Used to refresh the contents of USB drive
Type	There are two types of USB connection based peripherals, some of which stores data while others don't such as mouse keyboard,etc. So, if the peripheral is non-storage then the browse button will be disabled and if the peripheral is a storage device the browse button will enabled to open the USB device contents
Browse USB content	To open the USB flash and read its contents
Log	Messages appear when IOPB system is activated
Raspberry Pi to PC	This button switch between Raspberry Pi and computer system

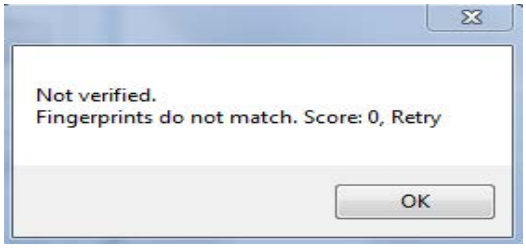

Fig. 8: Finger print is not match

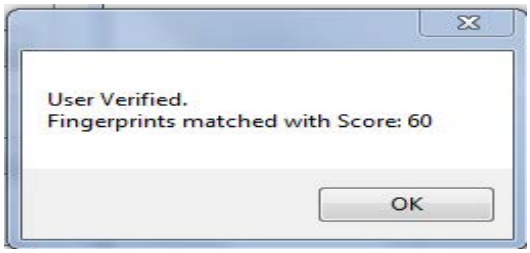

Fig. 9: Finger print is match

Client side: The client side is programmed using visual studio and C sharp language. The user can control the USB port means that can put it ON or OFF at any time but the default case all the USB port is OFF and when the user wants to use any port can use the authentication system (IOPB) to make the port on. At first when the computer started all the USB port is OFF and the security system (IOPB) will resaerch and ask for user authentication (PIN and fingerprint) to make the ports ON. Figure 7 after entering the PIN and fingerprint there are two cases:

- The fingerprint and/or PIN don't match Fig. 8
- The fingerprint and PIN are matched Fig. 9

When the user presses the ok button the system work and can control and modify the USB port case between ON and OFF state. The main window is shown in Fig. 10. The main form has few fields and buttons, each one of them is describe in Table 1. When the USB based

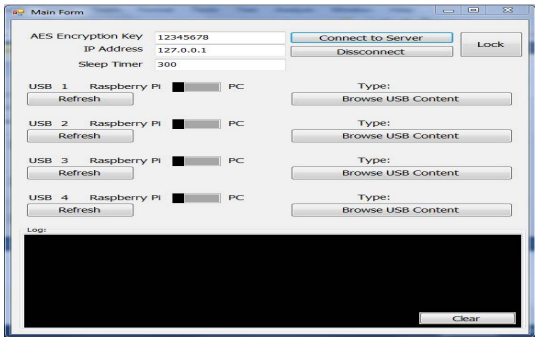

Fig. 10: Main from of the IOPB system

Fig. 11: USB device opened in Raspberry Pi

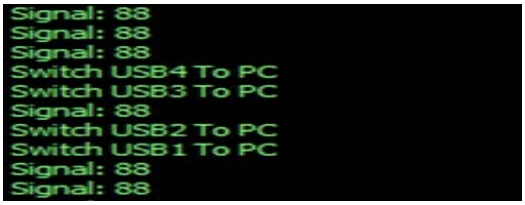

Fig. 12: USB device opened in PC for download or deleting items

device non-storage like mouse or keyboard is connected, the browse button will be disabled, since, it has no contents but when the system discovers a storage device, then the browse button will be enabled and the contents of the USB flash are displayed. Figure 11 shows the notification message when the USB drive opens in Raspberry Pi and Fig. 12 shows a notification when the USB opens in the computer system ready for download or deleting files.

RESULTS AND DISCUSSION

Applying security controls to a system has its drawbacks, so, if the security level applied to any computer system rises, the performance will decrease. Accordingly, the time required exchanging data between a peripherals device and a computer system increases. Our tests include transmitting image, voice and video of various sizes stored in a USB flash and downloaded to the computer system.

From figure above it can be seen that the time measurements required for downloading image, voice and video files from USB flash to the computer system when the IOPB is not activated are:

- For a minimum selected file size (100 kB): 0.92, 1.018, 1.211 sec, respectively
- For a maximum selected file size (10 MB): 2.932, 5.069, 5.398 sec, respectively

In other hand, the time required to transmit image, voice and video from USB flash to the computer system when the IOPB is activated are:

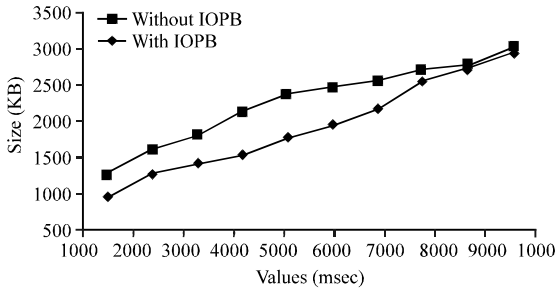

Fig. 13: Results obtained from the transmission of image; Comparison between image downloads with and without IOPB

- For a minimum selected file size (100 kB): 1.249, 1.32, 1.454 sec
- For a maximum selected file size (10 MB): 3.008, 5.076, 5.478 sec

Form the curves in the figures above, we can see that when the file size increases, the time required for downloading files decreases (Fig. 13-16). This happens

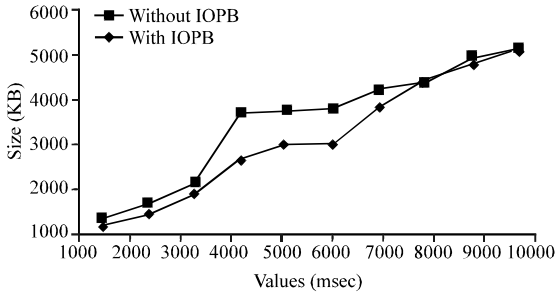

Fig. 14: Results obtained from the transmission of vioce; Comparison between voice downloads with and without IOPB

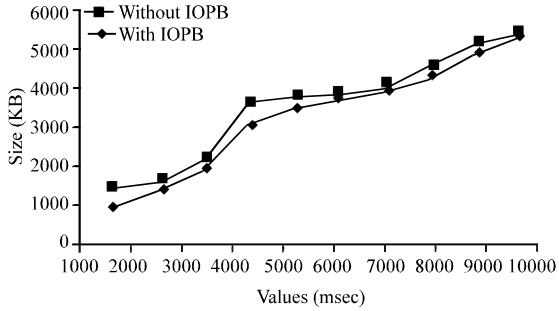

Fig. 15: Results obtained from the transmission of video; Comparison between video downloads with and without IOPB

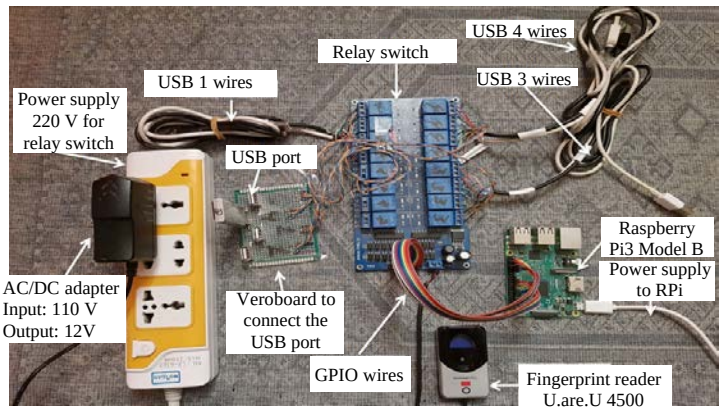

Fig. 16: IOPB prototype

when the IOPB is used as sockets are created to exchange data between the Raspberry Pi and computer system.

The IOPB prototyping: Hardware components of the IOPB system prototype are connected together as shown in Fig. 16. As can be seen, the external USB ports in the Veroboard are connected to the relay switch which in turn connected to the windows based system and Raspberry Pi3 via. GPIO and USB wires. From the other hand, the Raspberry Pi3 is connected to the fingerprint reader and windows based system.

CONCLUSION

In this research, we presented the design and implementation of a security system called IOPB. It is used to isolate the USB port from the computer system, thus, protecting the computer from different types of USB-based attacks. This research focuses on protection from attacks utilizing the authorized keyboard attachment, since, the attacker may use keystrokes attacks to extract important information from the computer system. The implementation is a hardware/software solution against USB attacks. The obtained results from the transmission of multimedia to the computer system show that the effect of adding the IOPB increases the delay approximately by 2.5, 2 and 3.5%, respectively. Thus, we conclude that the performance of the system supported by activating the IOPB is acceptable and the IOPB slightly affects the time required to transfer image, voice and video from the USB flash to the computer system. This research can be extended in the future to include other I/O ports and analyze the contents of the storage type I/O based peripherals to identify malicious code and adopt countermeasures accordingly.

REFERENCES

Ako, M.A., 2017. Advanced Encryption Standard (AES) algorithm to encrypt and decrypt data. Cryptography Network Secur., 1: 1-12.

Anonymous, 2016. U.are.U® 4500 reader optical USB fingerprint reader. D igitalPersona, I nc., Palm Beach Gardens, Florida, USA.https://www. crossmatch.com/wp-content/uploads/2017/05/2016 0122-DS-En-U.are_.U-4500-Reader.pdf

Anonymous, 2017. The ultimate guide to Raspbian and other raspberry Pi software. Eltechs, Inc., Moscow, Russia. https ://eltechs. com/raspbian-and-other -raspberry-pi-software/

Anonymous, 2018. Raspberry Pi 3 model B. Raspberry Pi Foundation, ?Cambridge?, UK. https: //www. raspberrypi.org/products/raspberry-pi-3-model-b/

Boileau, A., 2006. Hit by a bus: Physical access attacks with firewire security. Syst. Rev., 42: 93-98.

Henson, M.J., 2014. Attack mitigation through memory encryption. Ph.D Thesis, Dartmouth College, Hanover, New Hampshire.

Liao, T.L., P.Y. Wan, P.C. Chien, Y.C. Liao and L.K. Wang *et al*., 2018. Design of high-security USB flash drives based on chaos authentication. Electron., Vol. 7, 10.3390/electronics7060082

Markuze, A., A. Morrison and D. Tsafrir, 2016. True IOMMU protection from DMA attacks: When copy is faster than zero copy. ACM. SIGARCH. Comput. Archit. News, 44: 249-262.

Myung, K., 2015. USB wall: A novel security mechanism to protect against maliciously reprogrammed USB devices. MS Thesis, University of Kansas, Lawrence, Kansas.

Nissim, N., R. B Yahalom and Y. Elovici, 2017. USB-based attacks. Comput. Secur., 70: 675-688.

Patric, S., 2014. Detecting Peripheral-Based Attacks on the Host Memory. Springer, Berlin, Germany, ISBN:978-3-319-13514-4, Pages: 107.

Stewin, P. and I. Bystrov, 2012. Understanding DMA malware. Proceedings of the International Conference on Detection of Intrusions and Malware and Vulnerability Assessment, July 26-27, 2012, Springer, Berlin, Heidelberg, Germany, ISBN:978-3-642-37299-5, pp: 21-41.

Vujovic, V. and M. Maksimovic, 2015. Raspberry Pi as a sensor web node for home automation. Comput. Electr. Eng., 44: 153-171.