CHAPTER 1

INTRODUCTION

This chapter includes the following subtopics, namely: (1) Rationale; (2) Theoretical Framework; (3) Conceptual Framework/Paradigm; (4) Statement of the problem; (5) Hypothesis (Optional); and (7) Scope and Delimitation.

1.1 Rationale

Write-Blocker is a digital forensics device that protects the integrity of digital evidence from being altered [4]. The use of write-blockers is a practice that is accepted nearly everywhere worldwide and is more aligned across jurisdictions [5]. In other words, the use of a write blocker is the key to the admissibility of digital evidence in the court. Write-blockers also play a crucial role in preventing the contamination of digital evidence [6]. When forensic investigators plug an evidence NTFS drive into a PC [7] or preview a single image [6], they unintentionally cause evidence contamination. Without a write blocker, contamination is unstoppable in a second.

Write-blocker research has a lengthy timeline, beginning in the Pre-Automated Tools Era (1999-2001) [8] with investigations into Software Write Blocker (SWB), progressing to the Automated Tools Era (2001-2009) with the introduction of Hardware Write Blocker (HWB), and continuing to the present day. In comparison to SWB, HWB garners significant interest from the academic community for several reasons:

1. The ongoing advancement of storage interface technology and PC interface technology [2]
2. The growing accessibility of advanced embedded systems [2] the expanding capacity of hard disks
3. reliable, resistant to attacks, and enabling the distinction between first responder and forensic investigator responsibilities [8]

Fig. 1.1 depicted HWB setup during preservation of evidence.

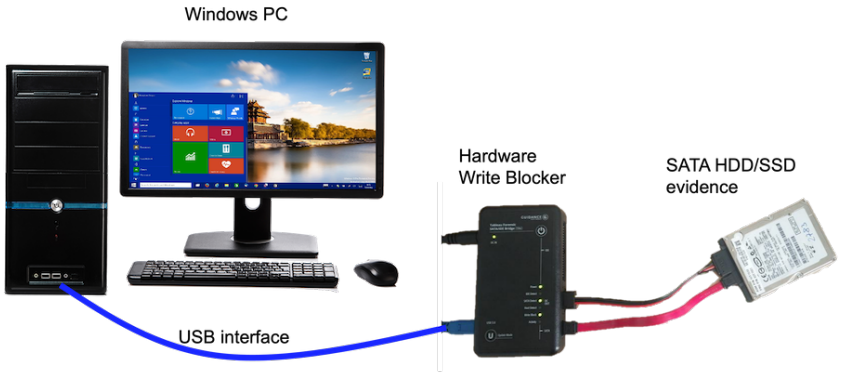

Figure 1.1: HWB setup with PC and evidence HDD

1.2 Theoretical Framework

Layer architecture has been implemented for interfaces and protocols such as TCP/IP, SATA, and SAM. According to Zenan [9], the modular nature of layered architecture promotes the reusability of components across different projects, which can lead to more efficient development cycles. In layer architecture, each layer should have specific functions; in terms of software architecture, it was called "high-cohesion." Each layer depends on each other in a "low-coupling" way as described by Pressman [10] and Fenton [11]. Either "high-cohesion" or "low-coupling" leads to portable layer architecture.

Figure 1.2 depicts the architecture of the SATA protocol and SCSI Architecture Modeling (SAM) in a single figure, as pointed out by Bruce [5]. Each architecture has four layers: application layer, transport layer, link layer, and physical layer. The application layer handles ATA and SCSI commands coming from the OS, handles them accordingly, and returns data for READ or INFO commands. Specific commands of ATA and SCSI are handled by this layer also. The SATA transport layer handles SATA Frame Information Structure (FIS) and the command queuing feature known as Native Command Queueing (NCQ). Whereas SCSI transport layer handles data transport protocol for Fibre, SAS, and USB Attached SCSI Protocol (UASP). The link layer handles packet data control before it is transmitted physically by the physical layer. This link layer is also called the SCSI Interconnects layer. The physical handle PHY signalling through cable.

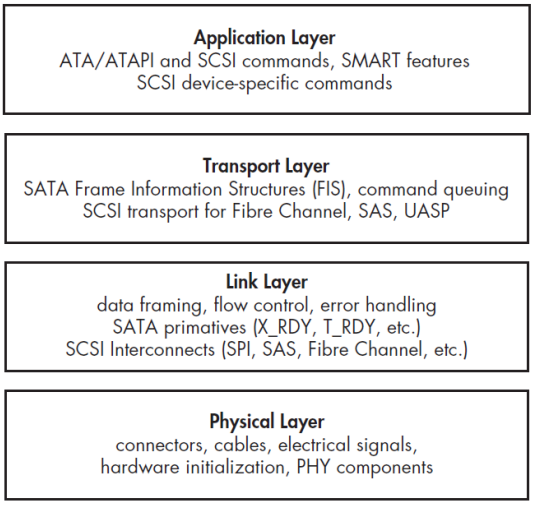

Figure 1.2: Layer Architecture of SATA and SAM

1.3 Conceptual Framework/Paradigm

HWB is physically attached as an interface between the host PC and digital evidence. HWB works by monitoring commands sent from the PC to the storage and prevents modifying requests while allowing read requests. Examples of modifying requests are write, format, and delete. To secure the digital from modification attempts, a cryptographic Secure Hash Algorithm (SHA) was used to verify the data integrity of digital evidence.

Besides data integrity, the National Institute of Science and Technology (NIST) launches the Computer Forensics Testing Tools (CFFT) project to ensure the reliability of forensic tools. The NIST CFTT has provided four mandatory HWB requirements [12], a comprehensive test scenario in the Assertion Test Plan (ATP) [3], and a ready-to-use Federated Testing Linux image (.iso). The ATP was a white box test during HWB development, whereas the Federated Testing is a black box test suite that adheres to the incidence response standard of International Standard Organization (ISO)/IEC27041 [13].

1.4 Statement of the Problem

The layered architecture is a popular paradigm for software architecture that encourages modularity, scalability, and maintainability while explicitly separating domains. Layered architecture in SATA protocol and SCSI Architecture Modelling (SAM) also encourages independency from a specific hardware platform. Prior work in HWB [14], [15] has implemented layered architecture, but a problem arises when the implemented hardware platform is discontinued [14]. The discontinued hardware makes prior work HWB obsolete. Based on this problem, the defined research question are

- RQ1: How to design an application layer that is hardware independent from the hardware layer?
- RQ2: What is required to validate the proposed design ?
- RQ3: What is required to validate the accuracy of the proposed design?
- RQ4: Does the proposed design affect the transfer speed?

1.5 Objective and Hypotheses

For this reason, there is an urgent need to avoid hardware dependency with a software portability design approach. Recent software design patterns from SOLID principles [16] introduce dependency inversion (DI) as a method to reduce hardware dependency. To be more specific, the main contributions of this research are

1. proposed portable architecture HWB of SATA protocol-based by low-coupling the application layer from the hardware-dependent layers.
2. Introduces a standardized Software Development Life Cycle (SDLC): portability requirement along with NIST HWB **requirements**, low coupling **design**, **implementing** a proof-of-concept (PoC) to a TUSB9261 USB-to-SATA bridge board, and ensuring functionality compliance with evaluation **metrics** and the NIST CFTT **test**.

1.6 Scope and Delimitation

Scope of Facility

1. TUSB9261DEMO USB-to-SATA bridge as validation board.
2. TSUB9261 RS232 Debugger with baud rate 115,200 bps as replacement of SATA protocol analyzer.
3. FTK Imager and Ubuntu DiskDump (dd) was tested under PC with i7-7700K for testing Windows 10 and Ubuntu environment. MacOS's DiskDump (dd) was tested under MacBook 2015 for testing MacOS environment.

Scope of Environment

1. Operating System: Windows 10, Windows 11, MacOS Monterey up and Ubuntu version 20 up.
2. Forensic imaging tool : FTKImager
3. C Compiler Tools from Texas Instrument

Scope of User

1. Forensics Examiner
2. First Responder
3. Law enforcement

Scope of Feature

Proposed USB-to-SATA HWB is likely to have similar performance (data transfer) , compatibility with commercial product while keep integrity intacted. In this research, proposed HWB performance will be compared with the popular yet easy-to-purchase commercial USB-to-SATA Tableu T35U Bridge (Figure 1.3).

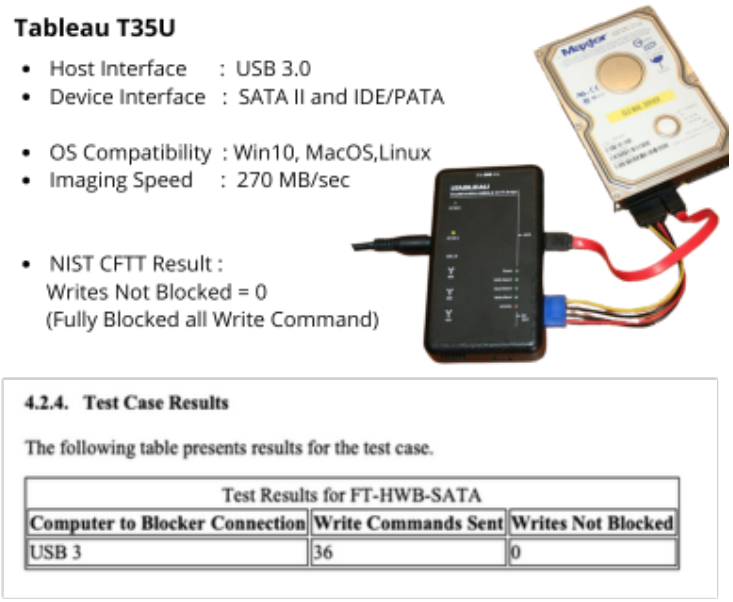

Figure 1.3: Tableau T35U specification

Scope of Usability

1. With this research , porting of write-block logic to other hardware platform will be easier and maintainable
2. User will gain the advantage of a hardware write blocker, which is plug-and-protect without necessary setup or kernel things.