Forensics Secure USB

Jonathan Chiu Zheng Hao, Sng Ray Shiang Nicholas, Karen Goh, Muhamad Fiqri Adam, Xande Koh Yong en  
*Infocomm Technology Cluster*  
*Singapore Institute of Technology*Singapore  
{2203187, 2203197, 2203430, 2202878, 2203203}@sit.singaporetech.edu.sg

*Abstract*— Maintaining the chain of custody for digital evidence is a critical challenge in forensic investigations, particularly in ensuring the confidentiality, integrity, and availability of the evidence. This research aims to present a proof of concept demonstrating that a disk storage device can accurately track its usage while preserving the confidentiality of sensitive information, maintaining the integrity of the evidence, and ensuring its availability for analysis. The proposed solution seeks to strengthen the reliability and security of digital forensic processes.

# Introduction

Digital evidence is becoming increasingly relevant in today’s world, making it crucial to ensure the integrity and accountability of evidence handling. Disk storage evidence is often handled over between multiple personnel, during which they will physically sign a paper to signify that the evidence has been handed over. This approach presents challenges, as it does not guarantee that the evidence has not been viewed by unauthorized personnel, accessed outside of the authorized location, or if the chain of custody has been strictly maintained. These issues can undermine the admissibility of the evidence [1].

A Chain of Custody (CoC) is a critical process in the management of evidence and investigations. CoC is a term that refers to the process of preserving and documenting the chronological history of digital evidence [2]

# Problem Definition

In digital forensic investigations, tracking access to evidence disks remains a critical challenge. Existing CoC methods lack the ability to monitor user activity on a host system but instead rely on manual logs or physical sign-off procedures, which are prone to errors and tampering. These traditional methods are inadequate for determining key factors of who accessed the digital evidence, specific actions performed and location of the access.

# Background Research

## **Literature Review**

In forensic investigations, CoC ensures the integrity and authenticity of evidence from collection to court presentation, which is especially critical for digital evidence. As digital data and networked systems become more prevalent, traditional CoC procedures face new challenges, such as increased risks of tampering and unauthorized access. Emerging technologies like blockchain and secure logging systems offer solutions to enhance transparency and reliability in handling digital evidence, yet globally accepted guidelines remain lacking [3]. This section will elaborate about the evolution of CoC practices, the challenges in digital forensics, and potential technological solutions to strengthen the process.

**Framework for CoC:**

The maintenance of a secure and reliable chain of custody is crucial in digital forensics to preserve the integrity and authenticity of the evidence. Jácome-Castilla and Villamizar-Nuñez [4] presented a comprehensive forensic analysis technique to manage digital evidence, particularly in information system attacks. This method involves obtaining forensic images, calculating hash values and documenting each step to ensure the evidence is unaltered and traceable. The paper highlights both technical and legal requirements for handling digital evidence to ensure that processes meet the forensic standards for admissibility in court. Key critical steps include creating exact forensic copies, securing the evidence with hash values, and maintaining logs of each interaction with evidence to prevent unauthorized tampering.

**Chain of custody using Blockchain:**

The potential for blockchain technology for maintaining a secure chain of custody has grown in interest as it provides an immutable and transparent ledger. Lone and Mir [5] proposed a solution, Hyperledger Composer, a permissioned blockchain framework. The Forensic-chain model ensures that only authorized users can modify or view the evidence records, therefore maintaining a secure CoC. Due to the characteristics of blockchain such as it is distributed, decentralized architecture and cryptographic guarantees makes it ideal for maintaining integrity, transparency, and auditability of the digital evidence during its lifecycle. However, this solution comes with certain limitations such as overhead performance which can impact system efficiency as transaction volume grows and the complexity of integrating blockchain with existing forensic tools.

**Utilizing Embedded Systems for Digital Forensics:**

Embedded systems are increasingly being used in digital forensic due to their versatility and ability to operate in resource-constrained environments. Mansor et al. proposed a mechanism for collecting forensic data from embedded systems in automotive ECUs (Electronic Control Unit), which demonstrated how much systems could enhance digital evidence collection through secure, privacy-preserving techniques [6]. The primary context is ensuring the authenticity, integrity and privacy of forensic data collected from embedded devices.

**Intrusion Detection System to Enhance Network Security Using Raspberry PI:**

Additionally, to automate applications, embedded systems like Raspberry Pi are being adapted for various forensic use cases. For instance, Jeremiah explored the use of a Raspberry Pi honeypot to enhance security [7]. This paper demonstrates the Raspberry Pi’s ability to monitor network traffic and detect malicious activity. The key advantage of utilizing Raspberry Pi in such setups is its affordability, portability and customizability making it a valuable tool in forensic investigations that require real-time monitoring and logging.

**USB-based Attacks and OOBAVD:**

USB-based attacks are a significant threat in industrial control systems, accounting for over 52% of cybersecurity incidents in these environments [8]. Air-gapped systems that were once considered secure, have proven to be vulnerable to malware such as Stuxnet and BadUSB. These attacks exploit the need for data transfers via USB devices, creating a major security gap.

To address this, Yu et al. [8] propose Out-Of-Band Antivirus Detection (OOBAVD), which acts as an intermediary between USB devices and air-gapped systems, scanning for malware before any data is transferred. OOBAVD operates out-of-band, minimizing the risk of tampering, and can be wiped and reflashed to remove persistent threats. This approach offers a robust solution for protecting critical systems from USB-based malware. For forensic investigations, OOBAVD’s model could be adapted to enhance the chain of custody by monitoring and logging unauthorized USB access, ensuring real-time tracking and integrity of digital evidence.

USB devices are convenient for data transfer and system updates however it poses significant security risk particularly USB-based attacks. As highlighted in the paper by Nir Nissim [9], attacks such as BadUSB and HID-based payload injections have exposed vulnerabilities even in air-gapped systems. HID attacks occur when seemingly benign peripherals like keyboard or mice are exploited to inject malicious code or gain unauthorized access to systems. It is relevant to the team’s proposed solution as the project leverages HID methods to track and log access to USB devices, capturing detailed information such as user identity, system activity and geolocation. By doing so, we aim to address these vulnerabilities and ensure that all interactions with USB devices are securely logged, enhancing both security and the chain of custody for digital evidence.

## **Existing Tools and Solutions**

In digital forensics, maintaining a reliable chain of custody is essential to ensuring the integrity and admissibility of digital evidence. Traditional paper record method is increasingly inadequate for tracking interactions that occurred within digital evidence. The need for automation and enhanced control over the evidence lifecycle has thus driven the need for automation and integrity of digital evidence. Tools such as P4wnP1 and Bash Bunny are known for the ability to execute Human Interface Device (HID) attacks, while useful for offensive security, are not designed for forensics use. They are tools used to simulate a trusted peripheral which allow an attacker to inject payloads and retrieve sensitive data. Although these tools were designed for penetration testing, their features can be adapted to use for accountability and provenance in forensics investigations. P4wnP1 is designed to emulate HID to allow execution of a variety of scripted payloads when plugged in Python or Bash. Similar to P4wnP1, Bash Bunny has a more user-friendly device that provides payload selector switch and plug and play functionality.

## Comparative Analysis of Existing Tools and Solutions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tool/Device | Type | Key Features | Advantaages | Limitations |
| P4wnP1 | Device Tool | - HID attack platform  - can log payload execution  - Can emulate trusted peripherals | -Customizable bash-based payload scripts  - Real-time interaction logging | - Hackers can exploit network weaknesses by executing attacks through USB mechanics |
| BashBunny | Device Tool | - tricks computers into divulging data, exfiltrating documents, installing backdoors | - user-friendly interface  - logs user actions when used in payloads | - effectiveness depends on target system’s configuration  - lacks networking features  - complex to maintain |
| USBDeview | Software Tool | - monitor USB devices that are currently or previously connected- allows to uninstall USB devices that were used | - simple and easy to use | - lacks real-time monitoring and features like geolocation |
| USB WriteBlocker | Hardware Device Tool | - permits read-only access to data storage devices without compromising integrity of data | - prevents any modification to the data on storage device  - offer plug-and play service | - slow down data transfer speeds  - physical device limitations |

## Comparative Analysis of Techniques for CoC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Technique | Type | Key Features | Advantages | Limitations |
| Blockchain | Decentralized, Digital | - Immutable ledger  - Tracks ever interaction  - Transparent and verifiable | - Provides tamper-proof records  - Allows decentralized verification  - Ensures integrity and authenticity | - Requires extensive computation resources |
| Manual CoC | Traditional physical record | - physical documentation of evidence handovers  - requires signatures | - simple to implement  - low cost | - susceptible to human error, fraud, and tampering  - difficult to track al interactions accurately |
| Decon (Decentralized consensus) | Distributed, consensus-based | - Uses many nodes(independent computers/systems) to verify evidence transactions  - fault-tolerant | - resistant to single-point failures  - distributes trust  - no central authority needed | - needs coordination between nodes  - more complex than traditional CoC Solutions |

The technique of Chain of Custody (CoC) the team is implementing is a manual chain of custody approach with digital automation. This method enhances the traditional manual CoC by introducing real-time data logging, which includes capturing details such as geolocation, user identity, PC name and hardware information. The data will then be stored in the logs folder, ensuring the integrity and authenticity of evidence by automated chain of custody.

# Proposed Solution

Given the limitations of traditional solutions in securing the integrity of the chain of custody, this project approach will be using a Raspberry Pi Zero 2W to fingerprint host computer information whenever plugged in.

The Raspberry Pi will act as a mass storage device, where an image file containing the evidence will be provided for the host pc to mount. In the Raspberry Pi OS, there will be a log folder for the device to store the information gathered. The device will only expose the evidence file to the host computer, allowing the host to interact with the evidence file for forensics investigations, while any fingerprinting information will be stored into the logs.

Additionally, the fingerprinting process will be gathered from the host computer via a HID attack. A HID attack script will be executed whenever a computer is plugged in or out. Keystrokes will be sent to the host computer to gather information, which will be sent over to the Raspberry Pi to be stored as logs. Since the logs cannot be accessed by the host computer, this ensures that the logs stored in the partition will be secure in ensuring evidence integrity and traces of the peripheral’s interaction with any other devices.

## **Process Flow**

A diagram of a software script

Description automatically generated*Fig 2. Process Flow*

### Initial Communication upon Connection

Upon connection of Raspberry Pi to the host computer, a HID script is automatically executed. This script facilitates the downloading and execution of the fingerprinting script from a server, which is important for gathering detailed system information from host computers such as the current user, operating system, BIOS version, hardware configuration, current time zone, public IP, and MAC address. These are essential to identify the users

### Secure Data Transmission using WebSocket Server

### The fingerprint data gathered by the Raspberry Pi is securely transmitted back to the Pi via a WebSocket Secure server. This server not only ensures encrypted communication but also handles bidirectional data flow between the Pi and the host, ensuring that all the transmitted data remains secure, preventing any potential data breaches or man-in-the-middle attacks. Real Time Monitoring with FileSystemWtacher

The FileSystemWatcher on the host computer monitors for any changes made to evidence files in real-time. This is to detect any unauthorized file access or modifications. Upon detecting any changes to the file, the FileSystemWatcher will trigger the Raspberry Pi to log these events. The logs are crucial as they provide a timestamped record of all the interactions with the evidence.

### Continuous Logging and Evidence File Management

All interactions and changes detected by FileSystemWatcher are logged in the Raspberry Pi’s local storage under the logs/ directory. This continuous logging is so that there is a detailed and unalterable record of all actions performed on the evidence.

A screenshot of a computer

Description automatically generated

*Fig 3. Example of logs stored in the tool*

### Secure Cleanup

Lastly, Raspberry Pi performs a secure cleanup to remove any scripts, logs, and temporary files that were created during the investigation. This step is crucial for ensuring that no data is left that could be exploited in the future. Thus, this maintains the integrity of both the host system and the Raspberry Pi.

## **Key Features**

### Composite Device

The Pi is configured as a composite USB device, combining the functionalities of HID and a mass storage device, allowing the forensic tool to perform multiple roles simultaneously.

*Human Interface Device (HID)*

As an HID, the host computer will recognize the Pi as a USB keyboard. This allowed the Pi to send automated keystrokes to the connected host, which is used to execute commands to retrieve and send information to and from the Pi.

*Mass Storage Device*

The pi is also emulated as a mass storage device to expose an img file containing the evidence files to the host computer, allowing investigators to interact with the evidence. The SD card will contain 1 file and a folder

* Evidence file: An img file which is made accessible to the user that plugs in the pi to interact with the evidence.
* Logs Folder: The logs folder is hidden and inaccessible from the user. Contains interaction and fingerprinting logs.

This functionality ensures that the evidence file remains isolated while providing the investigators an interface to interact with.

### WebSocket Secure Server

The device is equipped with a WebSocket Secure server, which functions as the bridge between the host computer and pi. The server's primary function is to send the fingerprinting script to the host computer and retrieve logs. By enabling real-time bidirectional communication, the server ensures that data is accountable during forensics investigations. Further the server employs SSL/TLS encryption to secure its communication channel. This ensures that the traffic between the pi and connected hosts is not susceptible to Man in the Middle (MiTM) attacks. The use of certificate-based encryption also helps to verify the authenticity of the server, preventing impersonation attacks.

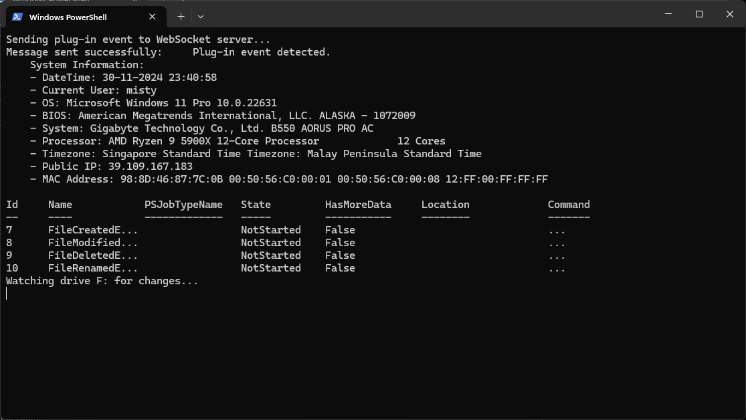
A screenshot of a computer

Description automatically generated*Fig 4. Data transmission encrypted with TLS/SSL*

### Fingerprinting Host Device

After retrieving the client script from the pi, and executing it on the host device, the script employs host fingerprinting techniques to collect detailed information of the host device, including the retrieval of key system details such as the current user, operating system, BIOS version, hardware configuration, current time zone, public IP, and MAC address. The script sends keystrokes via the HID interface to mimic user interaction, by utilizing a predefined character map for HID codes. These keystrokes access the command prompt, retrieve and execute client script on the host device.

The fingerprinted information is appended to each log message, assisting in easier identification of which host computer the pi is currently plugged into. This helps to provide comprehensive profiling of the host system, enable investigators to trace the interaction of the evidence file. The logged details would provide clear insights into the users that have interacted with the device and enable investigators to create a timeline of who has access the evidence.

*Fig 5. Fingerprinted information*

### Evidence File Integrity Check

Ensuring the integrity of evidence files is critical in their admissibility in legal proceedings. After fingerprinting the host device, the client script implements event monitoring on the host by using FileSystemWatcher. It captures real-time file events including the creation, modification, deletion, or renaming of the file. This functionality monitors the evidence file(s) in real-time, and it is logged and sent over to the pi over the WebSocket server, ensuring that any unauthorized changes to the evidence files are immediately detected and recorded.

### Cross-Platform Fingerprinting

The system also supports host fingerprinting on Linux, ensuring compatibility across diverse forensic investigation environments. Likewise, to windowskey information such as hostname, user details, operating system, BIOS version, and network configurations are captured on the Linux platform. Additionally, Linux is a preferred operating system for many forensic analysts and organizations due to its reliability, security, and open-source nature. Being able to extract key system information on Linux enhances the accuracy and comprehensiveness of the forensic process.

A screenshot of a computer

Description automatically generated  
*Fig 6. Fingerprinting on Linux Platform*

## **Evaluation**

## Comparing the team’s tool to existing tools, our solution surpasses existing methods in the following points. Unlike traditional tools, it ensures evidence integrity, reduces manual effort, and enhances traceability, making it a more reliable and efficient solution for digital forensics.

### Real-time Monitoring and Alert system

The Pi can collect detailed host system information via HID keystrokes. This ensures that investigators can trace interactions with the device back to specific users, a capability lacking in the other solutions mentioned above.

### Tamper-Proof Logging

Logs are stored in an inaccessible partition by the host on the Pi. Combined with the secure transmission of logs to the WebSocket server on the Pi using SSL/TLS encryption, this ensures tamper protection of evidence logs on the Pi.

### Automation and Ease of Use

## As compared to WriteBlocker which is frequently used in the forensics field, the Pi automates the CoC forensics process using HID scripting and file monitoring, reducing manual effort to document the details of CoC.

# Future Work

### Cross Platform Compatibility

The solution can be better dynamically optimized to be able to detect the current operating system of the host system. Based on the host’s operating system, the Pi will decide what HID script to send to the host as different operating systems require different HID processes.

### Anomaly Detection

Implement machine learning algorithms to analyze all logs in real-time to detect any patterns indicative of tampering or unauthorized access.

### Executable Program

Transforming the HID script into an executable malware program would significantly enhance the speed and efficiency of the key features such as fingerprinting. The executable program can directly interact with the host system to extract the necessary information, bypassing the need for simulated user inputs.

### Firmware and device

With this prototype, An unauthorized user will still be able to SSH into the device and to remove the microSD card. By converting this program into a firmware and with the proper housing, the risk of a user to unplug the microSD card and modify the evidence or formatting of the device would be greatly reduced.

# Conclusion

In conclusion, the team’s solution addresses the limitations of traditional chain of custody (CoC) methods in digital forensics, which are prone to tampering, unauthorizes access, and inefficiencies. Leveraging a Raspberry Pi Zero 2 W, the solution integrates features such as HID scripting, real-time monitoring, secure logging, and encrypted communication to automate and enhance CoC processes. Thus, this solution redefines CoC standards, ensuring reliable and secure handling of digital evidence.

##### References

1. G. Giova, “Improving chain of custody in Forensic Investigation of Electronic Digital Systems,” Jan. 26, 2011. [Online]. Available: <https://www.researchgate.net/publication/267400650_Improving_Chain_of_Custody_in_Forensic_Investigation_of_Electronic_Digital_Systems>
2. A. H. Lone and R. N. Mir, “Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer,” Digital Investigation, vol. 28, pp. 44–55, 2019. doi: [10.1016/j.diin.2019.01.002](https://doi.org/10.1016/j.diin.2019.01.002)
3. T. D’Anna, M. Puntarello, G. Cannella, G. Scalzo, R. Buscemi, S. Zerbo, and A. Argo, “The Chain of Custody in the Era of Modern Forensics: From the Classic Procedures for Gathering Evidence to the New Challenges Related to Digital Data,” Healthcare, vol. 11, no. 634, 2023. doi: [10.3390/healthcare11050634](https://doi.org/10.3390/healthcare11050634)
4. N. Jácome-Castilla and C. Villamizar-Nuñez, “Computer technique for the suitability of digital evidence in attacking an information system: Forensic analysis for the digital custody chain,” Journal of Physics. Conference Series, vol. 1388, no. 1, pp. 12027, 2019. doi: 10.1088/1742-6596/1388/1/012027
5. A. H. Lone and R. N. Mir, “Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer,” Digital Investigation, vol. 28, pp. 44–55, 2019. doi: [10.1016/j.diin.2019.01.002](https://doi.org/10.1016/j.diin.2019.01.002)
6. H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian, “Log Your Car: The Non-invasive Vehicle Forensics,” in 2016 IEEE Trustcom/BigDataSE/ISPA, IEEE, 2016, pp. 974–982. doi: 10.1109/TrustCom.2016.0164
7. J. Jeremiah, “Intrusion Detection System to Enhance Network Security Using Raspberry PI Honeypot in Kali Linux,” in 2019 International Conference on Cybersecurity (ICoCSec), IEEE, 2019, pp. 91–95. doi: 10.1109/ICoCSec47621.2019.8971117
8. “Out-Of-Band Anti Virus Dock (OOBAVD) - A Hardware & Artificial Intelligence Based Anti Virus Solution,” Black Hat, n.d. [Online]. Available: <https://www.blackhat.com/us-23/arsenal/schedule/#out-of-band-anti-virus-dock-oobavd---a-hardware-38-artificial-intelligence-based-anti-virus-solution-31923>
9. N. Nissim, R. Yahalom, and Y. Elovici, “USB-based attacks,” Computers & Security, vol. 70, pp. 675–688, 2017. doi: 10.1016/j.cose.2017.08.002
10. Swiftb0y, “P4WNP1,” P4wnP1 Official Wiki, n.d. [Online]. Available: <https://p4wnp1.readthedocs.io/en/latest/>