CAUID: Chip Against Unintended Information Disclosure

Monday 27th May, 2024 - 20:33

Georgi Bozhkov
University of Luxembourg
Email: georgi.bozhkov.001@student.uni.lu

This report has been produced under the supervision of:
Bernard Steenis
University of Luxembourg
Email: bernard.steenis@uni.lu

***Abstract*—As technology continues to develop and become more accessible, becoming a victim of a cyberattack becomes increasingly probable. Advancements in the sphere of cybersecurity are continuously made, but in some instances, even such improvements are not enough, and rather than relying on software that works in conjunction with the internet to prevent malicious attempts at data theft, a more convenient and responsive method could be implemented against security penetration attempts. A computer chip named CAUID collaborates directly with the CPU in order to immediately stop any attacking attempts.**

1. Plagiarism Statement

I declare that I am aware of the following facts:

- I understand that in the following statement the term "person" represents a human or **ANY AUTOMATIC GENERATION SYTEM**.
- As a student at the University of Luxembourg I must respect the rules of intellectual honesty, in particular not to resort to plagiarism, fraud or any other method that is illegal or contrary to scientific integrity.
- My report will be checked for plagiarism and if the plagiarism check is positive, an internal procedure will be started by my tutor. I am advised to request a pre-check by my tutor to avoid any issue.
- As declared in the assessment procedure of the University of Luxembourg, plagiarism is committed whenever the source of information used in an assignment, research report, paper or otherwise published/circulated piece of work is not properly acknowledged. In other words, plagiarism is the passing off as one's own the words, ideas or work of another person, without attribution to the author. The omission of such proper acknowledgement amounts to claiming authorship for the work of another person. Plagiarism is committed regardless of the language of the original work used. Plagiarism can be deliberate or accidental. Instances of plagiarism include, but are not limited to:

  1) Not putting quotation marks around a quote from another person's work
  2) Pretending to paraphrase while in fact quoting
  3) Citing incorrectly or incompletely
  4) Failing to cite the source of a quoted or paraphrased work
  5) Copying/reproducing sections of another person's work without acknowledging the source
  6) Paraphrasing another person's work without acknowledging the source
  7) Having another person write/author a work for oneself and submitting/publishing it (with permission, with or without compensation) in one's own name ('ghost-writing')
  8) Using another person's unpublished work without attribution and permission ('stealing')
  9) Presenting a piece of work as one's own that contains a high proportion of quoted/copied or paraphrased text (images, graphs, etc.), even if adequately referenced

  Auto- or self-plagiarism, that is the reproduction of (portions of a) text previously written by the author without citing that text, i.e. passing previously authored text as new, may be regarded as fraud if deemed sufficiently severe.

2. Introduction

The story of antivirus systems goes back to 1971, when Ray Tomlinson, an American computer programmer, developed a program called "Reaper" with the intention of protecting the first wide-scale packet-switched network and the predecessor of the Internet, ARPANET, from a detected virus called "Creeper". Even though the created program

was essentially also a virus, made with the goal of finding files corrupted by "Creeper" and erasing them, this could be considered the first concept of an antivirus system. Since that point on, they have become a necessity for every device that is connected to the Internet.

A method of penetrating the security system of a computer, more commonly referred to as "hacking", is an act that allows the gaining of personal information without the consent of an individual. Despite numerous advancements made to various security systems throughout the years, people with technological expertise and malicious intent are still able to frequently manage to get past the barrier-like layers and access confidential information. Additionally, it is inevitable for the occurrence of such unethical events to rise due to the globalisation of technology over time.

Due to the aforementioned increase in danger on the Internet, people have been searching for the most reliable and secure antivirus systems for their devices in order to maximise their safety online. The Chip Against Unintended Information Disclosure, or CAUID for short, aims to be a dependable security system that minimally affects the performance of the device while reducing its chances of being compromised by malicious attacks targeting it.

The task of this BSP is to develop and proof the concept of a convenient and responsive antivirus system that does not interfere with the experience of the user but also enhances the security of the device it is implemented on. The task in the scientific deliverable section is to justify the implementation choice of CAUID, which is done by answering the scientific question, "Are hardware-based antivirus systems better than software-based ones?" In order to find an answer to the question, a number of features that the security systems acquire are assessed based on given and predefined criteria. Additionally, a table is created, containing the final results, which are then analyzed and provide assistance with effectively reaching a conclusion.

The goal of the technical deliverable of the BSP is to prove that the concept of a CAUID antivirus system is achievable, which is done in two parts. The first part contains a description of the implementation of the chip. It includes an explanation of all the components CAUID is comprised of and what their purpose is, as well as a comprehensive explanation of its working. For the second part of the proof, a Python program is developed with the intention of simulating how the chip functions under its intended circumstances. Important sections of the code are further interpreted.

3. Project description

3.1. Domains

3.1.1. Scientific. The domain of this scientific deliverable is centered around the knowledge sphere of cybersecurity. As the goal of the deliverable is to compare two types of antivirus systems, it is essential for this section to provide an elementary level of knowledge regarding their working processes. That includes various algorithms and monitoring methods utilized by them.

3.1.2. Technical . The technical deliverable of the BSP focuses on the hardware components of an ordinary computer. In order for CAUID to work as intended, it needs to be able to operate in conjunction with the CPU. For successful communication process, instructions and data buses are to be utilized

3.2. Targeted Deliverables

3.2.1. Scientific deliverables. The goal of the scientific deliverable is to find an answer to the question, "Are hardware-based antivirus systems better than software-based ones?" This is achieved by doing a comprehensive analysis of both types of security systems across various criteria. Afterwards, the comparisons are shown in a structured table format, showing a clear comparison between the two types of antivirus systems. Through an assessment of the results, a conclusion is reached.

3.2.2. Technical deliverables. The goal of the technical deliverable is to justify the existence of CAUID as a concept. This is done by providing a detailed explanation of the structure of CAUID, including its components and working procedures. For clarity, a scheme is applied to visualise the security chip and its connections with the CPU. Additionally, Python language is used to create a program that simulates the working process of CAUID.

4. Pre-requisites

4.1. Scientific pre-requisites

From a scientific standpoint, before beginning with the BSP, a solid understanding of the procedures of an antivirus system is required.

4.2. Technical pre-requisites

From a technical standpoint, before beginning with the BSP, one must have an understanding of how a computer is structured on a physical basis and knowledge about how communication between parts of the computer works. It is important for one to know what the purpose of caches is, as the security chip bears much resemblance to it. The production of this section contains a program created in the Python programming language; therefore, an intermediate on the topic is necessary.

5. Scientific Deliverable

Question: Are hardware-based antivirus systems better than software-based ones?

5.1. Requirements

The main objective of this section of the BSP is to arrive at a validated answer to the previously mentioned scientific question. The justified conclusion is the result of a thoroughly examined comparison between hardware-based and software-based antivirus systems. The observation is theoretically based and analyses the differences between both security systems by individually assessing their performance based on features that they are both accustomed to.

Before beginning the comparison, the upcoming design section is to contain a subsection that includes a set of thoroughly defined criteria that are based on characteristics that outline what a desired antivirus system is. The procedure for the juxtaposition between hardware-based antivirus (HBA) and software-based antivirus (SBA) is as follows:

- **Feature and its Definition:** In terms of an antivirus, a feature is a set of algorithms and/or data recognized by the security system with the purpose of increasing the protection of a device against potential threats. The features listed in this comparison are necessary for antivirus systems to be acquainted with in order to establish a secure space against cyberattacks. Before the comparison between the performances, a paragraph containing an in-depth definition of each feature is given with the purpose of showing its importance for the creation of such systems.
- **Comparison between HBA and SBA** With the assistance of the predetermined criteria as well as the detailed definition of the feature, the comparison between HBA and SBA is possible to be carried out correctly. We show the positive and negative traits of both systems and compare their performances in regards to the current feature.

In order to arrive at an answer to the scientific question, a table is to be created, containing a shortened but informative version of the results concluded from the theoretical comparison. Upon an in-depth analysis of the provided outcome, a conclusion to the question of whether HBA is superior to SBA is reached.

5.2. Design

5.2.1. Criteria. In order to give a justification for the results of the comparison between the HBA and SBA systems, it is essential to establish certain criteria for evaluation. These criteria will serve as the foundation with which it is possible to assess the capabilities of both types of antivirus.

Usability The usability of an antivirus system focuses on the scale of difficulty that users experience when interacting with it. Various tasks and factors, such as installing, ease of interface navigation, updating, and configuring, are what this criterion encompasses. The desired security system is user-friendly and does not require proficiency in the sphere of technology in order for one to interact with it. Its installation on a device should be straightforward, as should the user interface. The available features need to be quickly accessible and understandable to the average person in order to avoid confusion and minimize complexity. The usability of an antivirus should also include the possibility of self-repairing whenever an error within the system occurs. Conclusively, the intention of this criterion is to evaluate features of the security system that improve the experience of a user.

Performance Performance refers to the amount of system resources an antivirus software needs to efficiently operate. The utilization includes consumption of the power of the CPU, RAM and network bandwidth. For the ideal security system it is vital to minimize the usage of resources, while in parallel preserving its capabilities. By optimising the utilization, smooth working and lack of significant effects on the running of a device are ensured.

Effectiveness In the context of antivirus systems, effectiveness relates to the precision which they acquire to detect and eliminate malicious attacks. Making use of various detection methods against malware, such as real-time detection and heuristic analysis increases the chances of success for the antivirus. The rate of false-positives is another factor covered by the effectiveness of the system. Instances of identifying non-malicious files as a potential threat must be minimal to non-existent. In addition to accuracy, the effectiveness of a cybersecurity system is also based on its responsiveness. Time is of most importance when an attempt of security penetration is initiated, therefore the antivirus must apply the appropriate countermeasures in a nimble manner.

5.3. Comparison of Features

5.3.1. Real-Time Detection. Real-time threat detection allows for the immediate identification and response to malicious activities. Software-based antiviruses continuously monitor system activities and incoming files, comparing them against a database of known malware signatures to detect suspicious behavior. While SBA provide effective real-time protection, their reliance on the operating system can introduce delays in threat detection and response. On the other hand, hardware-based antivirus systems operate independently of the operating system, embedded directly into the CPU architecture. This integration enables HBA to perform real-time threat detection with minimal latency, offering immediate identification and mitigation of threats.

5.3.2. Resource Consumption. System resource consumption directly impacts the overall performance of the computer system, influencing its effectiveness and usability. SBA systems typically require a significant amount of system resources, including CPU processing power,RAM memory, and disk I/O, especially during scanning operations. This resource consumption can lead to system slowdowns, increased boot times, and reduced performance, particularly

on older or less powerful computers. Additionally, frequent updates and background scanning processes further contribute to resource usage, affecting the usability of SBA. In contrast, HBA generally has lower resource consumption compared to its software counterpart because it utilizes dedicated hardware resources for threat detection and mitigation. HBAs have minimal impact on system performance, allowing for smoother operation and an improved user experience, even during intensive tasks or high-demand scenarios. Thus, in terms of system resource consumption, HBA systems outperform SBA systems, providing more efficient protection without compromising system performance.

5.3.3. Resistance to Malware. Malware creators continuously develop sophisticated techniques to evade detection; therefore, it is important for antivirus software to be able to apply countermeasures as fast as possible. SBA systems primarily rely on signature-based detection and heuristic analysis to identify malware patterns and behaviors. However, these methods can be easily bypassed by encrypted malware variants that alter their code to evade detection. Additionally, rootkit techniques and fileless malware can hide from traditional antivirus scanners by exploiting vulnerabilities in the operating system. In contrast, HBA systems offer enhanced resistance to malware evasion techniques. HBA operate at a lower level of the system, monitoring CPU instructions and system behavior in real-time. This enables them to detect and block malware at the hardware level, making it more difficult for malware to evade detection.

5.3.4. Updating. With the aim of ensuring their reliability, antivirus systems require frequent updates with information about newly developed versions of viruses. In both HBA and SBA, these updates are essential for keeping the antivirus protection current and capable of detecting and neutralizing the latest threats. However, there are differences in their implementation and impact based on these criteria. HBA systems often rely on firmware updates provided by the manufacturer. These updates, while less frequent, are critical for enhancing the system's malware detection capabilities. On the other hand, SBA systems receive regular software updates from the vendor, which are more frequent and seamless but can impact system performance during the update process. Additionally, SBA require internet connection in order for them to be updated. This allows for the creation of files, that could identify as ones needed for updating the software, but are actually malicious. Upon their downloading, they could manipulate and compromise the entire antivirus.

5.3.5. Behaviour-based Detection. Behavior-based detection in antivirus systems involves monitoring the behavior of programs and processes to identify suspicious or malicious activity that may indicate the presence of malware. In SBA systems, behavior-based detection is typically implemented through heuristic analysis and machine learning algorithms. These systems analyze the behavior of programs in real-time, looking for anomalies or patterns indicative of malicious behavior. SBA solutions can quickly adapt to new threats by updating their detection algorithms based on the latest behavioral patterns observed. However, the reliance on software-based algorithms can sometimes lead to false positives or missed detections. HBA systems approach behavior-based detection differently, as they do not recognize software directly but follow sequences of instructions. Instead, HBA solutions focus on monitoring the execution of instructions and detecting deviations from expected behavior at the hardware level.

5.3.6. Scheduled Scanning. Scheduled scanning is a feature in antivirus systems that enables users to set specific times for automatic virus scans. In SBA systems, users can schedule scans to run at designated times, providing convenience and allowing for regular scans without manual intervention. However, scheduled scanning in SBA systems may impact system performance and usability, as it consumes resources and may slow down other tasks. Conversely, HBA systems do not typically support scheduled scanning due to their real-time monitoring approach and lack of software support. HBA continuously monitors system activities, offering immediate threat detection without the need for scheduled scans.

5.3.7. Heuristic Analysis. This approach is used by antivirus systems to detect previously unknown or new malware based on behavioral patterns. In SBA systems, heuristic analysis involves analyzing the behavior of programs and files to identify potential threats based on their attributes and actions. SBA systems use complex algorithms to assess the risk level of files and programs, allowing them to detect and block suspicious activity before it can cause harm. However, heuristic analysis in SBA systems can sometimes lead to false positives or missed detections, as it relies on identifying patterns associated with known malware. On the other hand, HBA systems approach heuristic analysis differently.

5.4. Production

This section contains an assessment of the results from section 4.2. Upon the completion of analysis of the data, a solution to the scientific question is reached.

5.4.1. Creation of a Table. In order to adequately address the scientific question posed in this BSP, a table has been created to summarize the results of the comparison between HBA and SBA systems. This table aims to provide a simplified overview of the capabilities and performance of each type of antivirus system, facilitating a comprehensive analysis to answer the scientific question question.

TABLE 1: Comparison of Features between HBA and SBA

Features	HBA	SBA
Real-Time Detection	Yes (OS independent)	Yes (OS dependent)
Resource Consumption	Yes (Minimal)	Yes (Noticeable)
Resistance to Malware	Yes (Strong)	Yes (Dependence on Updates)
Updating	Yes (Manual)	Yes (Automatic)
Behaviour-based Detection	Yes (Instruction Monitoring)	Yes (Software-based Monitoring)
Scheduled Scanning	No (Real-Time Monitoring)	Yes (Convenient, Customizable by User)
Heuristic Analysis	Yes (Instruction Comparison)	Yes (Algorithm Usage)

5.4.2. Result Analysis. As shown in Table 1, HBA systems excel in real-time threat detection due to their integration into the CPU architecture, which enables immediate identification and response to malicious activities. Unlike software-based antivirus, HBA operates at the hardware level, allowing it to monitor instructions from the processor directly. This deep integration allows the hardware-based systems to detect and block malware in real-time, significantly reducing the window of opportunity for malicious activities. By analyzing CPU instructions, HBA systems can detect anomalies and deviations from expected behavior, making them highly effective in identifying and mitigating threats. Additionally, the hardware-level monitoring provided by HBA systems offers enhanced resistance to malware evasion techniques, such as rootkits and fileless malware. This proactive approach to security makes it more difficult for malware to evade detection, thereby providing a higher level of protection for the system and its users.

One of the strengths shown by HBA systems, as highlighted in Table 1, is their ability to maintain minimal impact on system performance. By utilizing dedicated hardware components for threat detection and mitigation, HBA systems effectively reduce resource consumption compared to their software-based counterparts. This reduction translates into smoother system operation and an overall improved user experience. Furthermore, the integration of HBA systems into the CPU architecture enables them to excel in real-time threat detection and resistance to malware evasion. These hardware-based solutions provide immediate identification and response to malicious activities, making it easier for it to capture malicious attacks.

While HBA systems display superiority in several aspects, including real-time threat detection and malware evasion resistance, they also exhibit limitations. Notably, HBA systems lack certain features such as scheduled scanning and automatic updates. These limitations arise from their real-time monitoring approach and the inherent constraints of their hardware-based design, as indicated in Table 1. Despite these drawbacks, the robust performance and enhanced security offered by HBA systems make them a compelling choice for users seeking advanced protection against malware threats.

In terms of behavior-based detection, Table 1 shows that SBA systems utilize Programs and algorithms to monitor and analyze program behavior in real-time. These systems can quickly adapt to new threats by updating their detection algorithms based on the latest behavioral patterns observed. However, the reliance on software-based algorithms in SBA systems can sometimes lead to false positives or missed detections, especially when dealing with sophisticated malware variants. In contrast, HBA systems, as given in Table 1, follow sequences of CPU instructions and detect deviations from expected behavior at the hardware level. This method of monitoring enables the antivirus system to efficiently detect and respond to suspicious activity without relying on software-based analysis. However, implementing behavior-based detection in hardware can be challenging and may require specialized hardware components, making it less flexible than software-based approaches. Despite these challenges, HBA systems offer robust protection against malware threats by leveraging real-time hardware monitoring and detection capabilities.

5.4.3. HBA vs SBA. When evaluating HBA and SBA systems, it becomes evident that each approach offers distinct advantages and drawbacks. SBA systems, excel in their flexibility and ease of updating, allowing for seamless integration with existing operating systems and frequent updates to combat emerging threats. However, they also often suffer from higher resource consumption, leading to system slowdowns and reduced performance, particularly on older or less powerful computers. Moreover, their reliance on software-based algorithms for behavior-based detection may result in false positives or missed detections, compromising their effectiveness in detecting sophisticated malware variants.

On the other hand, HBA systems, boast real-time threat detection capabilities and minimal impact on system performance. By integrating directly into the CPU architecture and monitoring system activities at the hardware level, HBA solutions offer immediate identification and response to malicious activities, enhancing overall system security. Additionally, their lower resource consumption ensures more efficient system operation and improved user experience, even during intensive tasks. While HBA systems may lack certain features, such as scheduled scanning and automatic updates, their robust protection against malware evasion techniques and efficient behavior-based detection mechanisms outweigh these limitations.

With its real-time threat detection, minimal impact on system performance, and efficient behavior-based detection mechanisms, hardware-based antivirus provides robust security solutions unmatched by software-based alternatives. Given these factors, we can reach a solution to the scientific

question, which is that HBA is superior against cyber threats.

6. Technical Deliverable

6.1. Requirements

The goal of the technical section of the BSP is to provide thoroughly justified proof that the concept of CAUID can be effectively implemented. To achieve this, the section is divided into four parts, three within the Design section and one in the Production section. Each subsection delves into distinct aspects of the security chip, crucial for a comprehensive understanding of its design and functionality. These sections will cover the architecture, operational mechanisms, and potential applications of the security chip, providing a detailed overview of its functionality and capabilities. Each critical component and feature of the security chip is thoroughly examined and justified.

- **Components** This part focuses on the individual parts and components used for the development of CAUID. Each component will be thoroughly described, explaining its purpose and role within the framework of the security chip.
- **Characteristics** This section focuses on the key design decisions that define the qualities of CAUID. It highlights specific features and implementation choices that enhance the effectiveness of the HBA. This part of the BSP offers a detailed explanation of these essential characteristics and provides thorough justifications for the selected design choices.
- **Functionality** This part outlines the theoretical working cycle of CAUID, explaining its entire process. It describes in detail the method through which the components of the chip communicate with each other with the goal of detecting and responding to potential threats.

To further validate the concept of CAUID, the Production section contains a Python program that demonstrates the functionality of the security chip in action. This program simulates the behavior of the HBA system, and the subsection provides a detailed explanation of the critical parts of the code, showing how each part contributes to the overall operation of CAUID.

6.2. Design

Before continuing with the design section of the technical deliverable, it is essential to introduce the concept of signature-based malware, as the functionality of CAUID is centered around detecting and mitigating this type of threat.

Signature Malware is defined by specific patterns in the data and memory addresses that interact with the memory of the CPU and its caches. These patterns consist of unique sequences or behaviors that set malware apart from legitimate data. For instance, malware might access certain memory locations, execute instructions in an unusual sequence, or manipulate data in ways that are atypical for benign software.

6.2.1. Components. This subsection contains information regarding the most important parts of the security chip used during the working cycle of the HBA.

Control Unit The control unit is responsible for organizing the operations within CAUID, ensuring efficient threat detection and response. It manages the interaction between the data storage and the comparator, coordinating the analysis of incoming data. When data arrives, the component directs it to the appropriate storage location and then ensures that the comparators receive the necessary data blocks for analysis against known malware signatures stored in the data storage. The CU handles unexpected errors, such as false positives, communication problems, and data corruption, by implementing corrective measures and rerouting processes as needed. This includes rechecking data or instructing the system to re-evaluate specific segments to confirm the presence of threats. By effectively managing these issues and optimizing the workflow between data storage and comparators, the control unit ensures the stability of the antivirus system and enhances protection against malicious activities, maintaining a seamless operation of CAUID within the CPU.

Data Storage The data storage component is responsible for archiving various predetermined methods of security penetration. It maintains an extensive database of known malware signatures that are used for identifying and mitigating threats. By storing this information, the data storage ensures that CAUID has immediate access to the necessary data to analyse, thereby enhancing the overall effectiveness and responsiveness of the antivirus system.

Comparators The comparators are tasked with juxtaposing the data provided by the CPU with the repository of malware signatures housed within the data storage of CAUID. This process enables CAUID to identify any potential threats or malicious activities, allowing for effective mitigation measures to be implemented.

6.3. Characteristics

To achieve an optimal balance between performance and security, particular design solutions are adopted. These solutions ensure that CAUID protects the system without significantly impacting its operational efficiency. By carefully selecting and integrating specific components and mechanisms, the design achieves robust protection against malicious attacks while maintaining the device's performance.

Placement of CAUID The strategic placement of CAUID within the hardware system achieves a delicate balance between performance and security through meticulous modifications to the structure of the CPU. Specifically, the

HBA is integrated inside the processor itself, necessitating significant alterations to its architecture. This integration grants CAUID direct hardware access to the caches and registers within the CPU, removing the need for communication through bus lines. This direct access facilitates swift and efficient data analysis, important for real-time monitoring and response to potential threats. Moreover, the strategic positioning of CAUID within the CPU architecture enables comprehensive analysis from the CPU's boot-up, in order to ensure robust security measures from the outset.

The implementation of CAUID within the CPU architecture also involves prioritizing access to Level 1 cache, enhancing its ability to analyze CPU operations immediately after boot-up. This emphasis on Level 1 cache access underscores the importance of proactive approach to security of CAUID, allowing it to swiftly detect and respond to any malicious activities occurring at the earliest stages of system operation. By seamlessly embedding CAUID within the CPU structure and granting it direct access to caches and registers, the hardware-based security chip effectively strengthens the overall security posture of the system while minimizing performance overhead. This strategic integration underscores the commitment to achieving optimal security without compromising system performance, positioning CAUID as a vital component in safeguarding against a wide range of potential threats.

Number of Comparators The decision to incorporate 1000 comparators within CAUID comes from the necessity of efficiently handling malware variants while maintaining a balance between detection speed and accuracy. Given the number of signature malwares, a larger number of comparators allows CAUID to analyze incoming data against a diverse range of known signatures. To ensure efficient comparison, the malware data is divided into equal-sized blocks of data, which are then processed through. By distributing the workload across multiple comparators simultaneously, CAUID can significantly enhance processing speed while maintaining accuracy.

Performance and Effectiveness A concern with the operation of CAUID is the ability to match the operation speed of the CPU, which is essential to the effectiveness of the security chip. Not being able to keep up the pace with the processor can lead to delays in malware detection and the operation of the device overall. To address this issue and ensure that the chip can keep up with the processor, several features have been incorporated into its design in order to optimize the processing speed and maintain the efficiency of the antivirus system.

Although the aforementioned comparators within CAUID are effective for individually analyzing malware signatures, having all comparators working on a single instruction simultaneously can be inefficient and slow. To overcome this, parallelization techniques are implemented, allowing multiple instructions to be compared simultaneously. Additionally, when receiving it, the malware data gets divided into equally sized data blocks, making the comparison process more manageable and enhancing the overall speed and efficiency of CAUID.

The implementation of this feature leads to increased resource consumption. Despite the additional resource usage from parallelization, the impact on the system's performance is minimal. The operations utilized by CAUID are not high demanding, which ensures that the resource consumption remains within acceptable limits, making it nearly unnoticeable during regular operation.

Voltage Comparator The purpose of the voltage comparator, not to be mistaken with the comparators used to capture malware within CAUID, is to monitor the power supply to detect whether the CPU is operational. When it detects a change in the power supply, indicating an activation of the CPU, CAUID also activates. This mechanism ensures that the chip remains vigilant, ready to initiate its security measures whenever the CPU undergoes changes in its operational status.

Access to RAM The direct connection between CAUID and the RAM of the device is essential for its security functions, requiring a direct connection to compare its stored data with that from the CPU. Once the voltage comparator signals the activation of the CPU simultaneously with the CPU copying the operating system onto the RAM, the security chip also copies its stored data there. This parallel process ensures that CAUID has access to the latest system information, enabling it to perform real-time comparisons and effectively safeguard the device against potential security threats.

Updating The update method for CAUID is designed to ensure the integrity and security without compromising efficiency. The decision to utilize peripheral hardware devices or USBs for updates is due to the difficulty of attacking CAUID directly. By leveraging external storage devices, the update process minimizes the risk of unauthorized access or tampering with CAUID's firmware.

The update process is to occur during the bootup sequence of the computer. Upon bootup, before the operating system fully initializes, the PC scans for the presence of a compatible USB device containing the update files. Once detected, the update process initiates automatically, integrating data against newly found malware versions into the data base of CAUID.

Upon completion of the update process within the brief window of the bootup sequence, the USB device is removed from the system. This ensures that no residual access points or vulnerabilities remain after the update. By adhering to this methodical approach to updates, CAUID ensures that its firmware remains up-to-date and resilient against emerging threats, thereby safeguarding the integrity and security of the systems it protects.

Detection Method The detection method used in CAUID requires a 75% resemblance between analyzed data sequences and stored data to initiate countermeasures. This

threshold enhances the security chip's ability to capture malicious attacks by avoiding the need for exact matches, which can lead to false negatives due to minor data variations or errors. In dynamic computing environments, where data transmission can be affected by noise or interference, expecting perfect matches is unrealistic and often results in missed detections of malicious activity. By allowing for some variance while ensuring substantial resemblance, CAUID effectively identifies malicious behavior, minimizes false positives, and maintains operational stability. This 75% threshold ensures high accuracy in identifying potential threats, reducing the risk of false negatives, and maintaining system reliability.

6.4. Functionality

CAUID, integrated within the CPU architecture, activates promptly upon detecting a surge in power supply. This mechanism ensures simultaneous startup with the CPU, offering immediate protection from system boot-up. The chip's power detection circuit, sensitive to voltage changes, triggers its initialization process, ensuring operational synchronization with the CPU.

Following initialization, CAUID proceeds to intake CPU instructions directly. Subsequently, the control unit of the chip divides the recieved data into blocks and distributes it among the available comparators. Once data distribution is accomplished, the comparators of the HBA actively scrutinize the incoming data by comparing it against pre-stored information within the data storage if CAUID. These comparators work in parallel, analysing multiple instructions at a time and continuously scan and compare data to detect any potential anomalies or similarities indicative of malicious activity.

If the comparators fail to identify a 75% resemblance between incoming and stored data, CAUID continues its comparison procedure uninterrupted. However, upon reaching the similarity quota between analyzed data and stored malicious data, the chip triggers a response mechanism. This prompts the chip to send a direct instruction to the CPU, initiating a device shutdown.

6.5. Producton

7. Conclusion