CAUID: Chip Against Unintended Information Disclosure

Wednesday 22nd May, 2024 - 23:38

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 is based on the scale of difficulty that users experience while 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 utilization of system resources by the antivirus software during operation. It includes the consumption of CPU processing power, memory (RAM), disk input/output (I/O), and network bandwidth. Effective antivirus software should have minimal resource usage to avoid impacting the overall performance of the system. High usage can lead to slowdowns, system freezes, and decreased productivity. Therefore, antivirus solutions are evaluated based on their ability to maintain low resource usage while providing efficient protection against malware threats. Optimized usage ensures that the antivirus software runs smoothly in the background without significantly affecting the performance of other applications and tasks on the computer.

Effectiveness Effectiveness in antivirus systems refers to the ability of the software to accurately detect and neutralize threats. It should utilize multiple detection methods, including signature-based detection, heuristic analysis, behavior-based detection, and machine learning algorithms, to identify and block malicious files and activities. Additionally, the antivirus should have a low false-positive rate, meaning it should correctly identify legitimate files and applications without flagging them as malware. Moreover, effectiveness also encompasses the speed and responsiveness of the antivirus system in detecting and responding to threats in real-time. A highly effective antivirus system provides timely protection, minimizing the risk of malware infection and ensuring the security of the computer system and its data.

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 com-

puter 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 has the possibility of being effectively implemented. To achieve this, the section is to be divided into four parts, three of which are within the Design section and the last being in the Production section. Each of these subsections delves into distinct aspects of the security chip, which are crucial for a complete understanding of its design and functionality. These sections should provide information about the architecture, operational mechanisms, and potential applications of the security chip. Each of these aspects is necessary to be discussed as they provide a comprehensive overview of the functionality and capabilities of the HBA, ensuring that every 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 should be thoroughly described, providing detailed explanations for its purpose and its role within the framework of the security chip. By providing information about each of its parts and their interactions in detail, an understanding of the workings and the integration of components that make up the CAUID is to be given.
- **Characteristics** This part concentrates on the important decisions, related to the qualities of CAUID. A HBA is designed to minimally impact, or fully not affect, the performance of the device it is implemented in while simultaneously being adaptive and providing protection against malicious attacks. To achieve this balance, specific design solutions have been made to ensure that CAUID meets these requirements. This part of the BSP provides an in-depth explanation of some of the key characteristics of the HBA, offering detailed justifications for why these particular design choices were made.
- **Functionality** This part shows the theoretical working cycle of CAUID, providing an explanation of its entire process. The explanation describes the mechanisms through which the aforementioned components the chip uses in order to detect and respond to potential threats.

In order to further validate the concept of CAUID, the production section contains a program coded in Python to demonstrate the functionality of the security chip in action. This program simulates the behavior of the HBA system. Additionally, the subsection provides a detailed explanation for the parts of the code which are of most importance, showing how each part contributes to the overall operation of CAUID.

6.2. Design

6.2.1. Components.

Control Unit The control unit (CU) 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. Additionally, the CU handles unexpected errors, such as false positives, communication problems, and data corruption. By effectively managing these issues, the control unit ensures the stability of the antivirus system and enhances the protection against malicious activities.

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 perform real-time threat detection and analysis, thereby enhancing the overall effectiveness and responsiveness of the antivirus system.

Comparator The comparators are tasked to juxtapose 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 and effective mitigation measures to be implemented.

6.3. Producton

7. Conclusion