Evaluating the Implications of Microsoft Pluton on the Trusted Platform Module Platform

Alexander Lotero
aal562@nyu.edu

Abstract---It appears as though security considerations are finally being made during the earliest steps computing products' initial design phase. At least, that is one of Microsoft's many claims about the future of the Windows operating system. Within this future is the introduction to a change to the Central Process Unit (CPU), the brain of every PC, itself. Microsoft's Pluton project proposes integrating the Trusted Platform Module, a microcontroller responsible for several security processes within a device, within the CPU itself. Our goal was to analyze the project through a theoretical lens, as no such CPU has not been released to the public yet. Our analysis shows promise for attacks that exploit the hardware in existing systems, but software vulnerabilities appear to be unchanged by this development.

I. Introduction and Motivating Example

Oftentimes, true, next-generation hardware developments are only possible when a manufacturer with considerable resources and industry relationships leverages these perks to realize their ambition. This is what Microsoft is doing with Project Pluton [1]. At least, according to Microsoft.

In partnership with the leading manufacturers of central processing units for PCs[1], CPUs, Microsoft has proposed an overhaul of the Trusted Platform Module (TPM) design to include the module itself within the CPU die.

In short, the TPM is a microcontroller designed specifically to service PC's security. The chip's primary responsibilities are encryption key storage and integrity checks at boot time to ensure that the system begins in a trusted state. Traditionally, this chip lives on the motherboard[2], usually connected to the CPU via the Low Pin Count (LPC) bus, a bus for devices with minimal bandwidth requirements.

Microsoft makes many claims about the security improvements made possible by Pluton, but to what extent are these claims valid and more than just marketing? What impact does this hardware overhaul have on existing attacks against the TPM?In this project, we discuss previous research on the TPM platform, theorize Plutons defences against existing attacks, and discuss whether or not we agree with Microsoft's direction.

II. Hypothesis

TPMs have become an essential part of the PC hardware ecosystem. With more than 15 years to develop attacks on the TPMs, malicious actors and researchers have developed a list of several possible exploits.

This list of attacks includes exploits of the hardware, and of the software that runs on the hardware. We theorize that while Microsoft's plan to integrate the TPM directly on the CPU die does directly combat or eliminate hardware attacks on existing TPM configurations, it may remain vulnerable to additional attacks that exploit the software running on top of the hardware. We analyze three existing attacks outlined by Sparks [2].

III. Related Research

An essential predecessor to our work is provided by Sparks [2]. Sparks introduces and tests the viability of three distinct attack vectors within the traditional implementations of the TPM. While Sparks carried out his research on TPM 1.2, the current TPM 2.0 standard retains many of the same vulnerabilities as its predecessors.

Sparks' three attacks are as follows. First, the software attack in which an attacker modifies regions of memory after the TPM's boot time inspection. Second, the timing attack in which an attacker measures the time taken by the TPM to process each bit of an encryption key during the "TPM_Unseal" operation. Evaluating these timings allows the attacker to infer the value of each bit in the key. Finally, we have the reset attack in which attackers with access to the hardware are able to signal the TPM with a hardware reset before feeding it a previously copied image of a trusted state to "trick" it into trusting its new, malicious state.

To gain a better understanding of Microsoft's proposed Pluton TPM implementation in practice, we visit the work of Mattioli et al [3] and their analysis of Microsoft's Xbox One X video game console. Their project investigates numerous hardware and software design decisions made by Microsoft when creating "limited function" systems like a video game console. They then discuss the benefits of those decisions, whether those decisions could be migrated to the PC, and if those same benefits would be present if they were migrated to the PC. Specifically, we are interested in their work, because the Xbox One X can be viewed as Microsoft Pluton 1.0 as it was Microsoft's first system to adopt the "TPM within the CPU" design. As such, the fact that this console remains "unhacked" at the time of writing this lends credence to Microsoft's security claims about Pluton.

Microsoft Pluton is yet unreleased. As such, it remains untested by users, white hat hackers, and even black hat hackers. That said, our project is focused on the theoretical aspects of Pluton. Upon release, researchers will have the opportunity to put the claims of Microsoft, and even our own to the test[3]. For now, we consider if the changes brought about by Pluton can, at least theoretically, defend against several attacks on the current TPM iterations as they exist today.

---

[1] Microsoft's CPU manufacturing partners are Intel, AMD, and Qualcomm.

[2] TPMs can be integrated by the board manufacturer, or added in later to a dedicated TPM port on the motherboard.

[3] More on this in the *Future Work* section.

IV. Empirical Evidence

As stated above, our analysis includes three unique attacks that target three different attack surfaces. First, is the software attack. The software attack sees the malicious user alter the *.text* segment of a program by copying the page table to the kernel's page table to locate the program's physical memory addresses to cause the program to behave "different" than intended. Next is the reset attack, otherwise known as the hardware attack. The reset attack takes advantage of a malicious users' physical access to the machine. Finally, the timing attack. In the timing attack, crafted by Boneh and Bromley, a malicious user provides the TPM with two different encrypted inputs several thousand times, and measures the time taken for decryption. This knowledge of time taken for decryption can then later be used to guess the value of each bit in the key thereby exposing the key. We draw on the attacks utilized by Sparks, for a more detailed breakdown of each attack please visit their work provided in the references section [2].

To actually judge the improvements made by Microsoft's implementation, we have devised a system to "score" different categories to measure improvement. This system assigns each attack a classification within four individual categories and then scores the attack as a whole based on these classifications. The categories are: hardware access (boolean attribute), cost, in dollars and/or time, of performing the attack (3 classifications), required knowledge of the OS functionality (3 classifications), and whether or not root privileges are required (boolean attribute). In this case, "hardware access" refers to the situation in which access to the hardware makes the attack possible. As a result a *No* in this category means that the system is less vulnerable to remote exploitation. Scoring will be performed as follows:

- Hardware Access Improvement: 1 point for *No*, 0 for *Yes*
- Cost: 2 for *High*, 1 for *Medium*, and 0 for *Low*
- Required Knowledge: 2 point for *High*, 1 for *Medium*, and 0 for *Low*
- Root Access Required: 1 point for *Yes*, 0 for *No*

A higher score represents an attack that is less accessible to malicious users. Because our system uses relatively low numbers, i.e. the highest score possible in a category is 2 for cost and knowledge respectively, we stress that any score improvement by Pluton over its traditional counterpart is a credit to Microsoft's implementation.

Let us examine the tables below to discover how Pluton compares to its contemporaries:

Software Attack	**Traditional (External) TPM**	**Microsoft Pluton (TPM inside CPU)**
Hardware Access	No	No
Cost	Low	Low
Knowledge of OS	High	High
Root Access Required?	Yes	Yes
Final Score	4	4

Fig 1: *"Scorecard" comparison of the software attack on Pluton vs. traditional TPM design.*

The final score within the software attack for both traditional TPM implementations and Microsoft Pluton is 4 (see Figure 1). This lack of improvement is to be expected. This attack takes advantage of vulnerabilities in the OS kernel, specifically those that permit users to run arbitrary code. As such, because the attack is reliant on software vulnerabilities, such as root users with the ability to insert a *loadable kernel module*. A hardware overhaul alone will not combat this attack. As long as these kernel vulnerabilities exist, this attack remains possible regardless of the TPM implementation. This is because the TPM's security attestation measurements are taken at boot time, so any changes in memory after that are not inspected by the TPM.

Reset Attack	**Traditional (External) TPM**	**Microsoft Pluton (TPM inside CPU)**
Hardware Access	Yes	No
Cost	Medium	High
Knowledge of OS	High	High
Root Access Required?	No	No
Final Score	3	5

Fig 2: *"Scorecard" comparison of reset attacks on Pluton vs. traditional TPM implementations.*

In the case of the reset attack, the traditional TPM implementation scores a 3, while the new Pluton implementation scores a 5 (see Figure 2). Similar to the software attack, the reset attack results are to be expected. Pluton is a change in hardware configuration, so it is unsurprising that the hardware attack is deflected by its deployment. The key difference is the hardware access category, traditional implementations are categorized as *Yes*, meaning that possession/access to the hardware enables this attack. As introduced above, the reset attack requires access to the low pin count bus (LPC bus on which the TPM and CPU communicate) to record the signal exchange and later send specified signals. Pluton was categorized a *No* here because the TPM is now a part of the CPU, meaning this bus is no longer necessary for the two components to communicate. We classified the cost for Pluton as *High*, the only TPM/attack combination to receive this classification in this category, because we believe this attack to be rendered impossible by including the TPM within the actual CPU, even when malicious actors have access to the hardware. The attack may still be possible by delidding the CPU[4], but we believe this is outside of the scope of the vast majority of attackers. That said, we will not know for sure until the implementation is publicly released for testing.

Timing Attack	**Traditional (External) TPM**	**Microsoft Pluton (TPM inside CPU)**
Hardware Access	No	No
Cost	High	High
Knowledge of OS	High	High
Root Access Required?	No	No
Final Score	5	5

[4] CPU Delidding: The process of removing a CPU's IHS (integrated heat spreader) which sits above the process control block (PCB) [4].

Fig 3: *"Scorecard" comparison of timing attacks on Pluton vs. traditional TPM implementations.*

Finally, the timing attack scores, much like the software attack, are identical between traditional TPM solutions and Pluton with a score of 5 (see Figure 3). While there exist several possible defenses against this attack, it is not an exploitation of the hardware, rather an exploitation of the TPM's logic and policy. Such defenses include programming the TPM to perform decryption in constant time. This removes the opportunity for malicious actors to accurately guess individual bits of the key by measuring timings. That said, as part of the upgrade to Pluton, in addition to the hardware overhaul, we predict that Microsoft will include policy improvements to combat this attack as well. Again, this is merely a prediction, we will not know for sure until Pluton is released to the public.

This concludes our scoring section, with the only visible improvement against the hardware-based reset attack. Now, let us break our findings down further and analyze what the results mean for Pluton and the future of the TPM platform in our upcoming *Conclusions and Future Work* section.

V. Conclusions and Future Work

The Pluton project to integrate the Trusted Platform Module (TPM) on the CPU die will eliminate existing hardware attacks, but on its own will do little to combat many existing software vulnerabilities (see Figure 4). This was our hypothesis. Now, after examining each attack under the lense of Microsoft Pluton, we stand by our predictions.

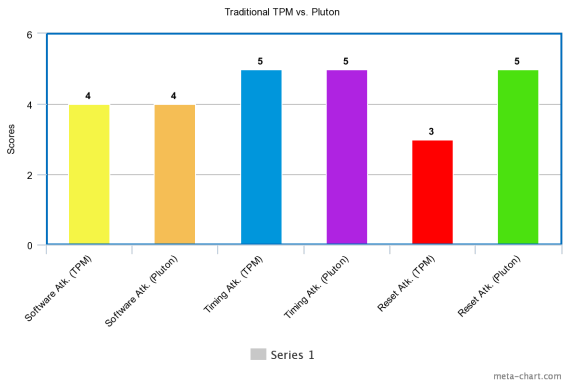

Fig 4: *Bar graph comparing traditional TPM implementations to Pluton for each attack from Sparks [2].*

Of course, this makes sense. A new development in the hardware configuration will have a direct impact on attacks that rely on the existing state of the platform. However, we anticipate that Microsoft has more in store for the future of the TPM platform.

Recent headlines have been packed with talk of Microsoft's first "full" OS revision in over five years with Windows 11, due out this year. Among these headlines is the discussion of the revision's hardware requirements, with one item garnering the most attention. Trusted Platform Module (TPM) version 2.0 required [5]. Windows 11 is Microsoft's first OS to require a TPM be enabled as a minimum requirement. Further, the OS requires the most recent revision of TPM, version 2.0. Press about the platform is loaded with marketing jargon about the choice to require this feature, but the important takeaway is Microsoft's emphasis on security and finally making the TPM an essential part of every PC.

This brings us to a discussion on where the project goes next. The answer: hands-on testing. Upon the release of the first generation of Pluton, sometime in the next one-to-two years. We look forward to seeing cyber security researchers such as ourselves put Pluton under a microscope, attempt existing TPM exploits on the new implementation, and discover what new exploits were created by the shift.

References

[1] D. Weston, "Meet the Microsoft Pluton processor – The security chip designed for the future of Windows PCs," *Microsoft Security*, 16-Mar-2021. [Online]. Available: https://www.microsoft.com/security/blog/2020/11/17/meet-the-microsoft-pluton-processor-the-security-chip-designed-for-the-future-of-windows-pcs/. [Accessed: 12-Jun-2021].

[2] E. R. Sparks, "A Security Assessment of Trusted Platform Modules," *Dartmouth Digital Commons*, 28-Jun-2007. [Online]. Available: https://digitalcommons.dartmouth.edu/senior_theses/53/. [Accessed: 17-Jun-2021].

[3] M. Mattioli and A. Lahtiranta, "Hidden Potential Within Video Game Consoles," *IEEE Xplore*, 29-Jan-2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9340369. [Accessed: 12-Jun-2021].

[4] S. Harding, "What Is CPU Delidding? A Basic Definition," *Tom's Hardware*, 06-Mar-2019. [Online]. Available: https://www.tomshardware.com/reviews/-delidding-definition,5738.html. [Accessed: 21-Jul-2021].

[5] D. Winder, "Windows 11 Security Stink Reveals Massive Microsoft ransomware red herring," *Forbes*, 08-Jul-2021. [Online]. Available: https://www.forbes.com/sites/daveywinder/2021/07/03/windows-11-security-stink-reveals-massive-microsoft-ransomware-red-herring/?sh=64428f222e1d. [Accessed: 21-Jul-2021].

[6] S. Berger, R. Cáceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn, "vTPM: Virtualizing the Trusted Platform Module," *USENIX*, 12-May-2006. [Online]. Available: https://www.usenix.org/conference/15th-usenix-security-symposium/vtpm-virtualizing-trusted-platform-module. [Accessed: 11-Jun-2021].

[7] J. D. Osborn and D. Challener, "[PDF] Trusted Platform Module Evolution: Semantic Scholar," *semanticscholar*, 2013. [Online]. Available: https://www.semanticscholar.org/paper/Trusted-Platform-Module-Evolution-Osborn-Challener/87f039c3ff03bb7dcd4fc66067f0e0f21449e7d5. [Accessed: 12-Jun-2021].

[8] "Trusted platform module (tpm) summary," *Trusted Computing Group*, 07-Mar-2018. [Online]. Available: https://trustedcomputinggroup.org/resource/trusted-platform-module-tpm-summary/. [Accessed: 12-Jul-2021].