Enhanced Security: Best Practices for Optimizing and Hardening a Windows 10 Environment

Atticus Emilsson
School of Cybersecurity
Old Dominion University
Norfolk, United States
atemilsson@gmail.com

***Abstract*—The Windows operating system is an incredibly powerful and versatile environment; however, it does come with default applications and configurations that impose on the performance and security of the system. One can effectively mitigate these issues by removing unnecessary software, disabling unnecessary data collection and transmission, and adding advanced security controls. Such configurations will reduce complexity, simplify management, and enable analysts to easily identify legitimate threats to the confidentiality, integrity, and availability of a Windows environment.**

Keywords—Windows, optimize, harden, safeguard

I. Research Overview

During the initial provisioning/configuration of a workstation, it is unlikely that one will be installing a minimal operating system. This means that the operating system will come with various pre-installed software, utilities, and drivers which allow for minimal configuration steps. Unfortunately, as these more commercialized operating systems – namely Microsoft Windows – become the standard operating system for desktop users, they have started to include more unnecessary software. This is not inherently an issue; however, this unnecessary software hampers system and network performance, individual and organizational privacy, and system security. This type of software is colloquially known as "bloatware" which has numerous "privacy and security costs" [1] and provides little (if any) benefit to the user experience. The Windows operating system and many of the applications included by default such as Skype, Microsoft Edge, and OneDrive also engage in the transmission of telemetry, which is data that is collected and transmitted for "monitoring, display, and recording" [2] which can have a considerable impact on privacy and system performance.

Bloatware can also be used to describe software that has an unnecessarily large codebase which creates undue complexity. Research suggests that the removal of redundant code and unused dependencies can be "applied for hardening purposes" or with the intent to reduce an application's attack surface [3]. Researchers from the Pennsylvania State University leveraged a "static-analysis enabled approach to trimming unused code" from applications and discovered that the size of applications using Java can be, on average, "reduced by 44.5%," or nearly half [4]. This releases resources that can be used more efficiently such as disk capacity and memory. Such "trimming" also reduces the complexity of the codebase, which inherently reduces the application's attack surface.

Although the Windows 10 operating system is much more secure compared to its predecessor [5], there is still room for improvement outside of basic cybersecurity hygiene, especially in an enterprise environment. In a study of various enterprise operating systems that aimed to identify which is the most vulnerable, based on the number of severe/critical vulnerabilities. Out of six unique and widely used operating systems in the enterprise environment, Windows was the most insecure [6]. There are various sectors of the Windows environment that can be configured with an approach tailored to hardening the system's security controls. There are also other areas in which the security of a Windows system can be improved, such as using software that is historically less vulnerable to exploitation. By aggregating the various methods proposed by the referenced journals and writings, one can effectively optimize and safeguard a Windows system, regardless of the operating environment.

II. Processes and Methodologies

When provisioning a machine that uses the Windows 10 operating system, specifically those designed for usage as a workstation, there are various steps that can be taken to optimize its performance and reduce the system's attack surface. The methodology presented in this writing aggregates various methods and processes that enable anyone to harden and optimize their system. The methods can be tailored to the needs of an individual, or those of an organization. Most of the frameworks used for optimizing and hardening the Windows environment are directed towards Windows 10; however, many of the processes are cross-compatible with the next iteration of the operating system, Windows 11.

A. INITIAL CONFIGURATION

There are various manners in which one can secure their basic configuration of the Windows environment and consists of basic cybersecurity hygiene. The first and foremost security measure that can be taken is to verify the integrity of the Window's installation being used. Using a compromised Windows installation leads to an inherently insecure system and should be avoided at all costs. While following through the Windows installation wizard, any options that feature unnecessary data collection and/or processing should be disabled. Any default user/administrator credentials should be changed to use a unique and complex password, and any guest users should be disabled and removed. The Windows System

Firewall should also be enabled and configured, alongside the Windows Defender system protection suite. Full disk encryption utilities such as Bitlocker also greatly enhance the confidentiality of a system's data. In a scenario where an attacker gains physical access to an encrypted device, they would not be able to gain immediate access to the data on the drive. By coupling Bitlocker with a strong passphrase/secret key and enabling the TPM, the physical security of a Windows workstation is greatly enhanced. It is of note that not all Windows 10 editions/iterations support Bitlocker such as the Windows 10 Home Edition [7]. However, most, if not all Windows Pro, Enterprise, and Education editions can leverage Bitlocker encryption [7].

B. BIOS SETTINGS

There are various system configurations that exist in the system BIOS (Basic Input Output System). The key combinations/shortcuts used to access the system BIOS may vary depending on the system/device manufacturer; however, many of the configurations that exist are quite similar. If applicable, one should enable the TPM (Trusted Platform Module) which keeps "the secrets of the operating system" [8] secure. Secure Boot should also be enabled, which ensures that a "device boots using only software that's trusted by the original equipment manufacturer" [9] meaning untrusted/unsigned installations/CDs will not boot. Another protection that exists in the BIOS configuration is a "boot/bios password" which can restrict an adversary from "booting or loading another system" on the workstation [10]. The system administrator can also disable any USB ports, revoking an adversary's ability to leverage the port as a channel of transmission/execution [10].

C. SYSTEM DEBLOATING

The next step in reducing the system attack surface and prioritizing resource usage is to "debloat" the system. The act of debloating can be described as removing any instances of bloatware from a system. One can remove any unnecessary software through various available Windows applets. Through the control panel, one can manage the installed software through the "Programs and Features" applet. Through system settings, one can manage installed software through the "Apps & Features" section. These sections enable a user to perform various actions on installed software such as repair an installation, restart or terminate the application, or uninstall the application. By uninstalling any unused applications, one can free up valuable disk space and cut down on any unnecessary transmission of telemetry. One might also consider removing tasks/applications from the Windows startup process via Task Manager, as some applications' startup processes are very resource-intensive.

As previously mentioned, the constant collection, processing, and transmission of telemetry and other data can hamper system and network performance and could pose a threat to individual/organizational privacy; therefore, the next step is to disable such transmissions to the greatest extent possible/necessary. Many of the settings pertaining to the collection and transmission of this information can be found in Windows' privacy settings. Many of the telemetry processes can be disabled without negatively affecting operations, so one can freely revoke any permissions/abilities depending on their use case. This includes disabling Windows' ability to track application activity, search results, location information, and other application diagnostics. One might also disable inking and typing personalization and Windows diagnostic telemetry. By disabling these data collection and transmission processes, one effectively reserves resources and bandwidth for other applications.

Such observations can also be made regarding other commonly used applications such as web browsers. By default, many browsers have minimal privacy and security controls enabled. Some browsers also perform more data collection and transmission (telemetry) by default, compared to others. By enabling an effective ad-blocking solution and configuring one's browser to stop unnecessarily tracking user browsing activity, one can better conserve their privacy and security and free up valuable resources to be used elsewhere.

D. SAFEGUARDING AGAINST MALICIOUS EXECUTABLES

There are further configurations to be made which protect a system, even when malicious software is executed by a user. These methods make it "theoretically impossible for current-day malware to execute directly" [11] in the Windows environment, particularly Windows 10. One can effectively blacklist or whitelist the execution of specific applications using AppLocker and DeviceGuard [11] respectively. The use of application whitelisting can be compared to a firewall with a deny-by-default rule, where only specified forms of traffic are allowed through. Using whitelisting compared to blacklisting is typically more secure and more comprehensive.

E. APPLICATION SELECTIONS

Outside of the configuration of the system, there are also various applications that are oftentimes more targeted than others. This may be because of their prevalence within the industry, or they are just inherently less secure due to the technologies being used. Zamora proposes an interesting approach [5] to selecting applications where one might evaluate an application's security based on the number of discovered vulnerabilities and their corresponding severity. One example presented by Zamora revolves around choosing a PDF reading and/or editing suite, where the commonly used Adobe Acrobat (878 discovered vulnerabilities) was substituted in favor of PDF X-Change Editor (one discovered vulnerability) [5]. Hence, one can effectively leverage CVE (Common Vulnerabilities and Exposures) statistics to better identify and select secure applications in favor of insecure (albeit commonly used) applications.

III. TOOLS AND RESOURCES

The debloating process is relatively straightforward; however, it can become tedious when performing it manually.

By leveraging scripts and tools provided by the open-source community, one can quickly debloat the Windows environment at scale. This not only increases the speed at which one can debloat a system but also enables an individual to remove other bloatware that is otherwise very difficult to remove. This includes removing OneDrive and/or Microsoft Edge from a system (if unused of course), disabling other system telemetry, and removing bloatware keys from the Windows registry.

One such tool for automating this process is *Windows10Debloater*, developed by a network analyst named Richard Newton. The *Windows10Debloater* is a PowerShell utility that enables an individual to remove bloatware and any corresponding registry keys, disable scheduled tasks, remove Cortana, and perform other tasks relating to the debloating of the Windows operating system [12]. Another PowerShell utility that can effectively debloat the Windows operating system is Win-Debloat-Tools which was developed by Plínio Larrubia. Larrubia's implementation aims to "debloat and improve privacy/performance and system responsiveness" with respect to the Windows operating system [13]. System debloating is by far one of the easiest and most powerful steps that can be taken to reduce a system's potential attack surface and improve performance without expanding the hardware or investing in other software solutions, and the use of specialized software makes the process significantly easier.

IV. Conclusion

Although Windows 10 and other iterations of the world-class operating system are incredibly useful and can be found on a significant portion of desktop computers, it is severely lacking in security compared to its competitors. This is due to the proprietary and critically complex nature of the system which facilitates a broader attack surface. A significant amount of bloatware, a characteristic of the Windows environment, also exists on these workstations which further threatens the performance, privacy, and security of the system. One can effectively optimize and harden the system and safeguard individual/organizational privacy by applying various configuration measures including (but not limited to) debloating the system, manipulating system and BIOS settings, and evaluating application choices based on CVE statistics. The steps presented can be applied to desktop computers belonging to an individual, or a workstation that is a part of a larger enterprise network. The steps are also mostly self-contained within the system, and do not *require* any extra tools; however, the usage of specialized software can expedite the process significantly.

Acknowledgments

Professor Gladden—For bettering my first semester at Old Dominion University with an incredibly educational and comprehensive course experience.

References

[1] H. Elahi, G. Wang, and X. Li, "Smartphone bloatware: An overlooked privacy problem," in Security, Privacy, and Anonymity in Computation, Communication, and Storage, G. Wang, M. Atiquzzaman, Z. Yan, and K. R. Choo, Eds., Springer International Publishing, 2017, pp. 169–185.

[2] "Telemetry," Britannica. https://www.britannica.com/technology/telemetry (accessed Apr. 11, 2024).

[3] P. S. E, W. Fischer, H. Plate, and A. Sabetta, "The used, the bloated, and the vulnerable: Reducing the attack surface of an industrial application," in 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 555–558. doi: https://doi.org/10.1109/ICSME52107.2021.00056.

[4] Y. Jiang, D. Wu, and P. Liu, "JRed: Program customization and bloatware mitigation based on static analysis," in 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), pp. 12–21. doi: https://doi.org/10.1109/COMPSAC.2016.146.

[5] P. Zamora, M. Kwiatek, V. Bippus, and E. Elejalde, "Increasing Windows security by hardening PC configurations," EPJ Web of Conferences, vol. 214, p. 08019, Jan. 2019, doi: https://doi.org/10.1051/epjconf/201921408019.

[6] J. Softić and Z. Vejzović, "Windows 10 operating system: Vulnerability assessment and exploitation," in 2022 21st International Symposium INFOTEHJAHORINA (INFOTEH), pp. 1–5. doi: https://doi.org/10.1109/INFOTEH53737.2022.9751274.

[7] "BitLocker overview - Windows Security," Microsoft, Nov. 06, 2023. https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/#system-requirements (accessed Apr. 11, 2024).

[8] A. Numminen, "Windows technical hardening against the most prevalent threats," jyx.jyu.fi, 2023. https://jyx.jyu.fi/handle/123456789/87270 (accessed Apr. 2024).

[9] "Enable Secure Boot on Windows devices - Microsoft Intune," Microsoft, Jul. 24, 2023. https://learn.microsoft.com/en-us/mem/intune/user-help/you-need-to-enable-secure-boot-windows (accessed Apr. 11, 2024).

[10] H. Nair and R. Sridaran, "An innovative model (HS) to enhance the security in windows operating SystemA case study," in 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1207–1211.

[11] R. Durve and A. Bouridane, "Windows 10 security hardening using device guard whitelisting and Applocker blacklisting," in 2017 Seventh International Conference on Emerging Security Technologies (EST), pp. 56–61. doi: https://doi.org/10.1109/EST.2017.8090399.

[12] R. Newton, "Sycnex/Windows10Debloater," GitHub, 2017. https://github.com/Sycnex/Windows10Debloater (accessed Apr. 11, 2024).

[13] P. Larrubia, "LeDragoX/Win-Debloat-Tools," GitHub, 2015. https://github.com/LeDragoX/Win-Debloat-Tools (accessed Apr. 11, 2024).