Prevention Strategies for Modern Living Off the Land Usage

Author: Matthew Alan Vorhees, matthew.vorhees@gmail.com

Advisor: Jonathan Risto

Accepted: February 10, 2024

Abstract

Usage of Living off the Land Binaries (LOLBins) by Advanced Persistent Threat (APT) Actors has risen over recent years, culminating in a high-profile attack on United States critical infrastructure. Current threat trends from Detection and Response vendors have confirmed increased detection of LOLBin techniques, which indirectly confirms that enterprises are generally not prioritizing prevention against threat actors using such methods. The availability of technology solutions does not warrant the lack of prioritization by enterprises, as tooling is available through Windows via AppLocker, Windows Defender Application Control rules, and supplemental vendor solutions.

While there is an adequate amount of modern research and active vendor engagement in detecting LOLBin usage, research on the prevention of LOLBin usage in the contemporary threat landscape needs to be improved. There is value in building upon existing research work done on LOLBin prevention to consolidate, expand on, and test the effectiveness of these prevention controls as well as evaluate their useability if applied to both technical and non-technical employee workstations.

1. Introduction

LOLBins has been known in the IT security community since at least 2018 (Moe, 2018). LOLBins are trusted binaries that come pre-installed on an operating system (OS) by default. The OS uses them to function normally. Abuse of these LOLBins by threat actors is increasingly attractive as they provide many benefits compared to traditional malware. LOLBins are naturally trusted by the OS and, as a result, bypass conventional security tools. Malicious LOLBin activity is sometimes difficult to distinguish from non-malicious LOLBin activity, and LOLBin abuse does not require the download and installation of any malware to the target computer.

Usage of LOLBins by Advanced Persistent Threat (APT) Actors has been rising over recent years (Barr-Smith et al., 2021), culminating in a high-profile attack on United States critical infrastructure (Cybersecurity and Infrastructure Security Agency, 2023) (United States and International Cybersecurity Authorities, 2023). Given that the holistic use of LOLBins by APTs is a relatively new tactic, enterprises should be aware of the most common LOLBin usage by APTs and understand how to adequately implement preventative controls to adapt to this change in approach by threat actors.

1.1. Prevention Not Prioritized

Recent threat trends from Detection and Response vendors have confirmed increased detection of LOLBin techniques (Red Canary, 2023) (Mandiant, 2023). If Detection and Response vendors are detecting threat actors increasingly using LOLBin techniques, this shows that enterprises are generally not prioritizing prevention against such methods. There are tools available through Windows that provide mechanisms for enterprises to prevent as well as detect these techniques. Some of these options include AppLocker, Windows Defender Application Control rules, and supplemental vendor solutions (Cottingham & Schell, 2023). Enterprises will continue to incur increased monetary and operational costs from allowing LOLBin usage to be managed through expensive

Detection and Response vendors and workflows. Informed prevention strategies significantly reduce the need for Detection and Response of this increasingly common class of attack techniques.

1.2. Modern Research on Prevention Lacking

While there is an adequate amount of modern research (Mostafa, 2022) (Store, 2023) and active vendor engagement in the detection of LOLBin usage, research on the prevention of LOLBin usage in the modern threat landscape seems lacking. Some historical analysis (Brown, 2020) on utilizing Windows prevention tooling through AppLocker has been performed but was limited in the scope of LOLBin techniques. Encouragingly, modern research (Barr-Smith et al., 2021) on the most common LOLBin usage leveraged by APTs has been performed and corroborated through Detection and Response vendor reports (Mandiant, 2023) (Red Canary, 2023).

1.3. Research Objective

Armed with this new research on the most prevalent LOLBin usage, there is value in building upon existing research work done on LOLBin prevention to consolidate, expand on, and test the effectiveness of these prevention controls as well as evaluate their useability when applied to both technical and non-technical employee workstations.

2. Research Method

This research will focus on typical enterprise use cases. This focus will require representing various user profiles within a standard enterprise. To balance the aspects of research scope and applicability, testing was performed on the latest version of the Windows 11 OS that will be supported for enterprise users the longest. Due to the different use cases of this OS across enterprises, it was valuable to establish typical user profiles for testing purposes. For example, a finance employee typically uses Microsoft applications (e.g., Excel) and a web browser. They usually do not need an Integrated Development Environment (e.g., Visual Studio). Conversely, a technical user may

execute PowerShell scripts, require an Integrated Development Environment, and use an SSH client (e.g., Putty).

2.1. Testing Environment

Windows 11 22H2 was used as the testing environment. There were three versions: a base install, an install with typical non-technical employee applications, and an install with typical technical employee applications. A matrix of identification codes that will be used for each version and configuration combination in this research can be found in Table 1 below.

Windows Version	**Base Install**	**Non-Technical Employee**	**Technical Employee**
Windows 11 Pro 10.0.22631 22H2	11_Base	11_N_Tech	11_Tech

Table 1: Identification Code for Each Testing Environment

2.2. Testing Procedure

Each identified LOLBin will be executed on each OS to confirm successful exploitation. The variable that will be modified is the implementation of specific preventative controls via AppLocker and Windows Defender Application Controls rules to evaluate the effectiveness of the defense in preventing the LOLBin execution. Further, typical work tasks will be conducted with typical workstation applications to assess if the performance of any task is impaired due to the preventative controls that have been implemented.

2.2.1. Identifying Top LOLBins

Recent studies of malware samples and incident response data from various threat actors have recently been performed to substantiate the claim that LOLBins are

increasingly being leveraged to have a more significant impact. Incident response data was a valuable resource demonstrating how common LOLBin usage by threat actors is in achieving impact on enterprises. Red Canary's 2023 analysis found the top LOLBins in incidents they detected and responded to, as listed in Table 2 below.

LOLBin	**Percentage of Organizations Affected**
Cmd.exe	33.3%
Powershell.exe	36.4%
Wmic.exe	12.4%
Rundll32.exe	14.6%

Table 2: 2023 Red Canary Data on LOLBin Usage

Additional research has been performed by analyzing malware samples to identify the frequency of LOLBin usage being employed by threat actors. This research concluded that LOLBins were used 26.26% by APTs in their malware samples (Barr-Smith et al., 2021). The most frequently used LOLBins by APTs from this study's dataset of over 3 million malware samples showed some overlap with Table 2 results and identified additional LOLBins to consider.

LOLBin	**Percentage of APT Malware Samples Containing LOLBin**	**LOLBin**	**Percentage of APT Malware Samples Containing LOLBin**
Ping	25.96%	Net	4.05%

Rundll32	12.01%	Tasklist	2.53%
Reg	11.16%	Ipconfig	2.47%
Wscript	10.22%	Expand	2.32%
Xcopy	6.42%	Systeminfo	2.16%

Table 3. IEEE Malware LOLBin Research

In addition to vendor and academic research, governmental agencies have researched and reported on APT usage of LOLBins. Governmental advisories on APT activity in 2023 showed an apparent prominence of LOLBin usage, particularly during highly impactful campaigns (United States and International Cybersecurity Authorities, 2023). This additional source of information reiterated many of the previously identified LOLBins from vendor and academic sources. Table 4 below represents the complete set of LOLBins that will be considered during this research.

LOLBin	
Ping	Cmd
Rundll32	Powershell
Reg	Wmic
Wscript	Tasklist
Xcopy	Ipconfig

Net	Expand
Systeminfo	

Table 4. Most Common LOLBins Used by APTs Used in This Research

2.2.2. Configuration of Testing Environments

There will be two testing environments that will be configured after base installation: 11_N_Tech and 11_Tech. 11_N_Tech will have traditional thick-client Windows applications installed that are the most common workplace utilities on a Windows OS: the Windows Office Suite. 11_Tech will have thick-client applications installed that technical users would typically require to perform administrative tasks, develop code in various languages, and run container images. Applications that are usually required for these tasks, which will be installed on 11_Tech, are Visual Studio, Docker, and Putty.

3. Findings and Discussion

3.1. Windows Defender Application Control (WDAC) and AppLocker

WDAC is a security feature introduced in Windows 10 and available in Windows 11. It provides the user with code integrity policy options to restrict what code can run in both user-mode and kernel-mode. It is the preferred solution for typical application control on a Windows host and is currently receiving feature and security updates from Microsoft.

AppLocker is the predecessor to WDAC and is still available for use in Windows 11. Although it is not receiving any feature updates from Microsoft, it still receives security updates. AppLocker is used for granular application control on a Windows host and can permit an allow-list approach of only permitting pre-identified binaries to execute or a deny-list approach of allowing all binaries to run except pre-identified ones.

3.2. Application Control Strategy for WDAC and AppLocker

Microsoft reinforces in its application control documentation that WDAC and AppLocker should be used as complementary tools. As WDAC was Microsoft's latest and preferred application control tool, initial research on LOLBin prevention strategies began with WDAC configuration. WDAC policies are more robust and intelligent than AppLocker equivalents, providing application control based on traditional binary hashes, publisher signatures, or binary paths. However, it also provides for application control through the current binary reputation as determined by Microsoft's Intelligent Security Graph and the identity of the binary installer. WDAC policies can be deployed through PowerShell or the WDAC Wizard (jsuther1974 & vinaypamnani-msft, 2023). The design of WDAC policies is based on a "circle of trust" unique to each enterprise's business requirements. Microsoft created several examples of base policies to serve as a starting point for organizations implementing WDAC, which are more easily accessible and implemented through the WDAC Wizard. WDAC also allows for the creation of Supplemental Policies that extend on the Base Policy for specific use cases that might be different across user profiles in an enterprise.

The initial configuration of WDAC policies for testing LOLBins led to the conclusion that WDAC is not natively designed as a preventative control for LOLBins. By default, almost all the base policies created by Microsoft trust all Windows binaries included with a new OS install. These are the very binaries that are increasingly abused by APTs that this research seeks to build preventative policies to control. This preference is demonstrated abstractly through the WDAC Wizard policy options, which all inherently include Windows binaries in their "circle of trust."

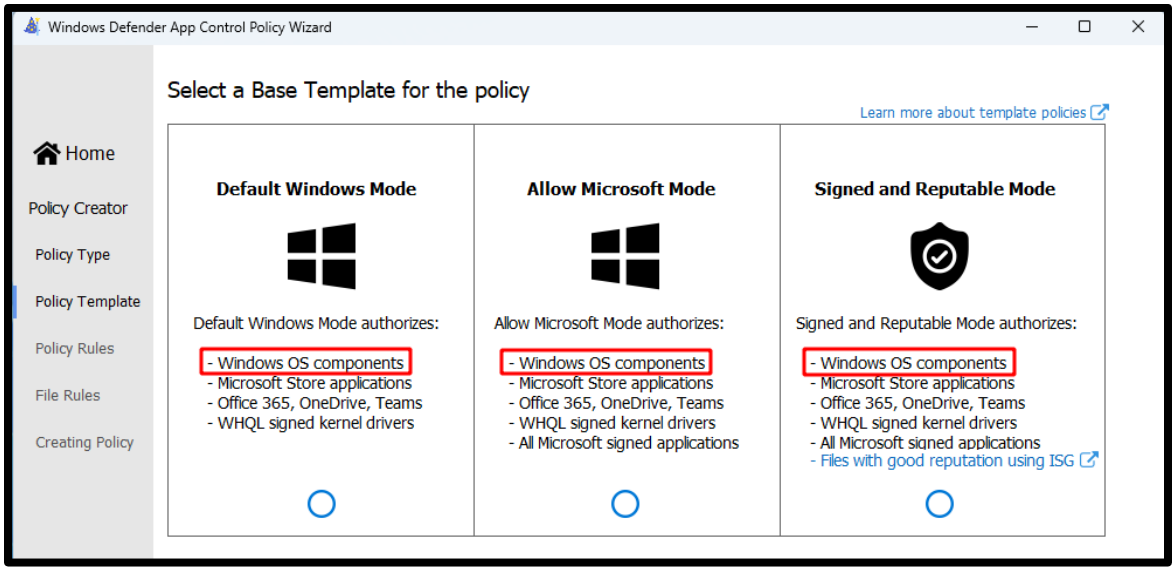

Figure 1. WDAC Default Policies Inherently Trust All Windows OS Binaries

WDAC does offer the ability to create custom block-list rules for pre-identified binaries based on traditional attributes like hash, publisher, or file path. However, a test of using the "Default Windows Mode" WDAC policy with a single deny rule for the hash of a known LOLBin like "ping.exe" did not result in blocking the execution of that binary at the command line, as shown in Figure 2.

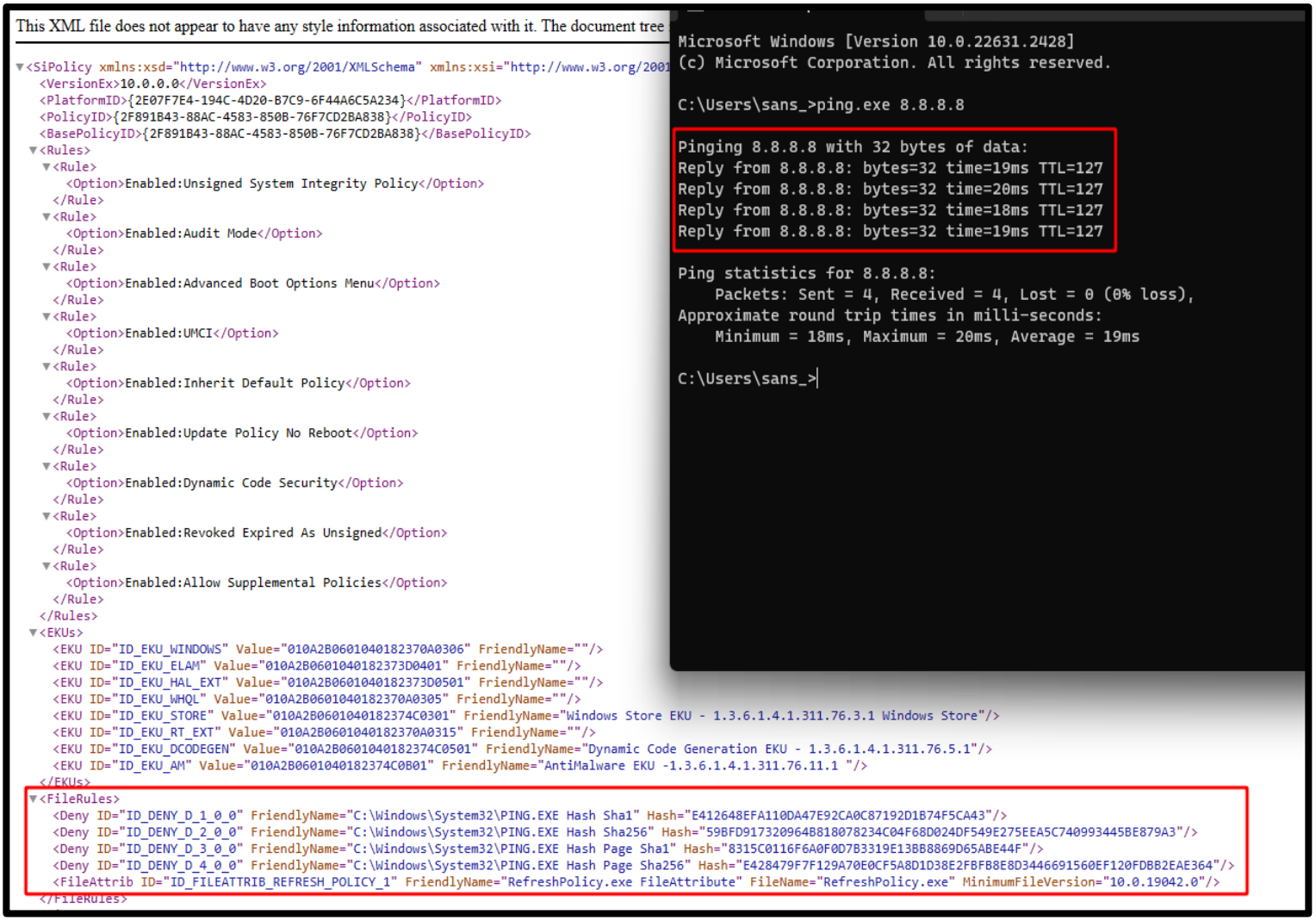

Figure 2. WDAC Default Windows Mode Policy with Ping.EXE Deny Rule Still Allowing Ping.exe Execution

The root cause of the failure of this simplistic test was not identified during variations of different WDAC policies, deny rules, and deployment methods. The most likely reason for the ineffectiveness of configuring a WDAC policy to block a LOLBin is the design assumption of the tool to include Windows OS binaries in the “circle of trust” by default. Configuring a tool like WDAC to intentionally contradict an apparent core

design assumption like this leads to undue complexity and ineffectiveness. Further evidence demonstrating the difficulty of leveraging WDAC for granular deny-list application control is that the user can only implement deny-list rules in the Base Policy. WDAC Supplemental Policies do not allow for the creation of deny-list rules. If an enterprise wants to update its LOLBin prevention deny-list, it must change the core Base Policy that applies to all users, which does not allow the configuration of nuanced deny-lists for different user profiles across the enterprise.

This result does not disqualify WDAC as an application control tool in a defense-in-depth approach. It is the preferred method for application control and should be used as part of a defense-in-depth strategy for application Control initiatives. WDAC is valuable in establishing a baseline of what groups of applications should be trusted based on various static and dynamic analysis options. However, AppLocker presents a unique benefit, providing more granular control over blocking specific binaries already inside the "circle of trust," as determined by WDAC. Continuing with our previous example of creating an application control policy to stop the execution of Ping.exe effectively, a WDAC policy was established using the "Default Windows Mode" option and no other configurations from the default. In addition, an AppLocker policy was configured to specifically deny the execution of the file hash associated with the "ping.exe" binary. Additional configuration to enforce the AppLocker deny rule required manually starting the "AppIDSvc" service for the rules to take effect.

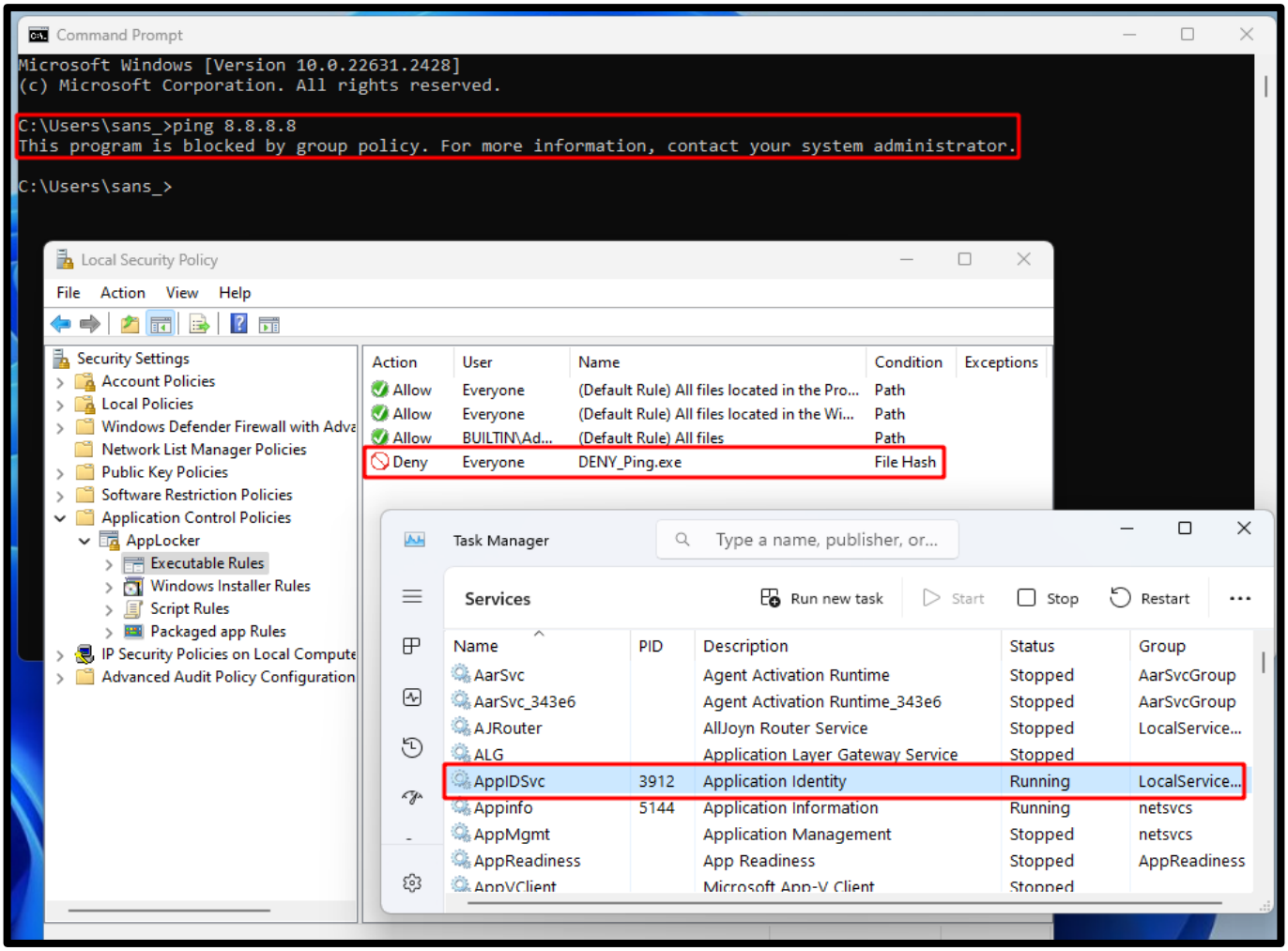

Figure 3. LOLBin "ping.exe" Execution Blocked with AppLocker Deny Rule

This study implemented a complementary approach of WDAC and AppLocker policies. WDAC defined broader allow-list policies. AppLocker determined granular deny-list policies on specific LOLBins inherently included in the WDAC allow-list policies.

3.3. 11_Base Testing Results

3.3.1. WDAC and AppLocker Executable Rules

A new Windows 11 OS was installed with no additional applications beyond what comes by default. The WDAC policy was established at the "Default Windows Mode" level, with all identified LOLBins denied through AppLocker by their file hash. Of note for the WMIC LOLBin, two executable files must be included in the blocklist: %SYSTEM32%\WBEM\WMIC.EXE and %SYSTEM32%\WMIC.EXE.

Evaluating this installation with these configurations encompassed two parameters: if the configurations prevented usage of LOLBins on the host and if the host's usability was impacted.

To determine the level of host usability impact, the following five everyday activities that a user takes with a fresh install of a Windows OS were performed: installs applications through the Microsoft Store, installs applications through a Web browser, customizes the host, uses the media player, and creates/organizes files in the file system. The usability score would be "No Impact" if all activities could be performed without impact. The usability score would be "Impacted" if one or two activities were impacted. If more than two activities were affected, the usability score would be "Severely Impacted."

Since the command prompt and PowerShell prompt were turned off through AppLocker configuration, AppLocker deny rules were created for all but the command prompt to assess if all other LOLBins were successfully blocked. After this testing, the deny command prompt rule was established to determine that this final LOLBin was blocked. Usability testing was then performed.

Usability testing with these WDAC and AppLocker configurations to deny execution of all identified LOLBins based on their hash for the pre-identified fresh install activities is summarized in Table 5 below.

Usability Activity	Usability Impacted?
Installs applications through the Microsoft Store	Impacted: Unable to open Microsoft Store application.
Installs applications through a Web browser	No Impact
Customizes the host	Impacted: Unable to open native Picture applications.
Uses the media player	Impacted: Unable to play sound or video files through any default Windows applications.
Creates/organizes files in the file system	Impacted: Unable to open multiple default Windows utilities like Notepad.

Table 5. 11_Base Useability Results

Usability was Severely Impacted unexpectedly, whereas usage of default Windows applications for traditional tasks was blocked. Log analysis was necessary to determine what rules were blocking the usage of these standard, default Windows applications. Any application blocked through AppLocker has logs generated describing the reasons for blocking or allowing execution of specific binaries in the Event Viewer -> Applications and Service Logs -> Microsoft -> Windows -> AppLocker. There are four categories of AppLocker logs: EXE and DLL, MSI and Script, Packaged app-Deployment, Packaged app-Execution as shown in Figure 4.

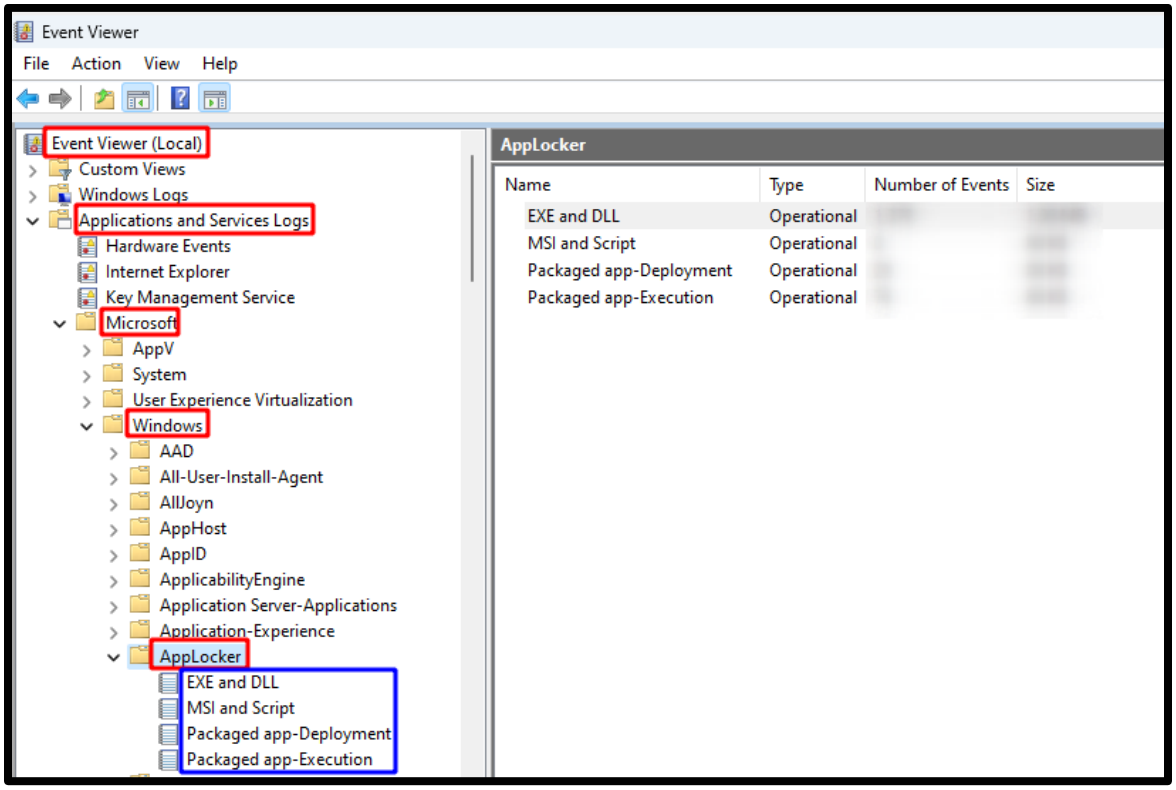

Figure 4. AppLocker Event View Log Location

Analysis of these logs showed the recurrence of Error 8027 in the Packaged app-Execution location every time a default Windows application was prevented from execution, as seen in Figure 5.

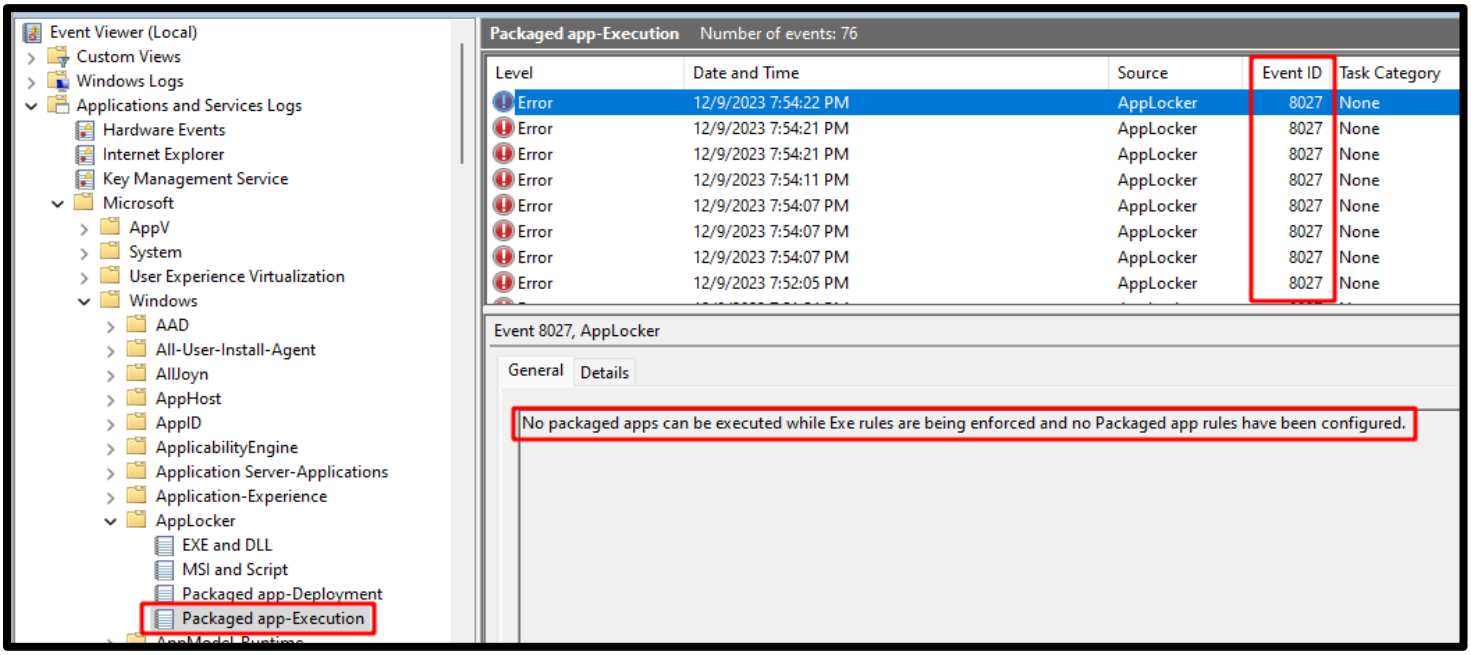

Figure 5. Event ID 8027 Blocking Execution of Default Windows Applications

As described, there is a requirement to configure Packaged app Rules to allow the execution of these default Windows applications. This configuration can be performed through the Local Security Policy tool and the Application Control Policies -> AppLocker -> Packaged app Rules. Various methods exist to create rules to allow these default Windows applications to execute. However, the most effective way for this use case of a base install of Windows 11 is to right-click on “Packaged app Rules” and select “Automatically Generate Rules” as seen in Figure 6.

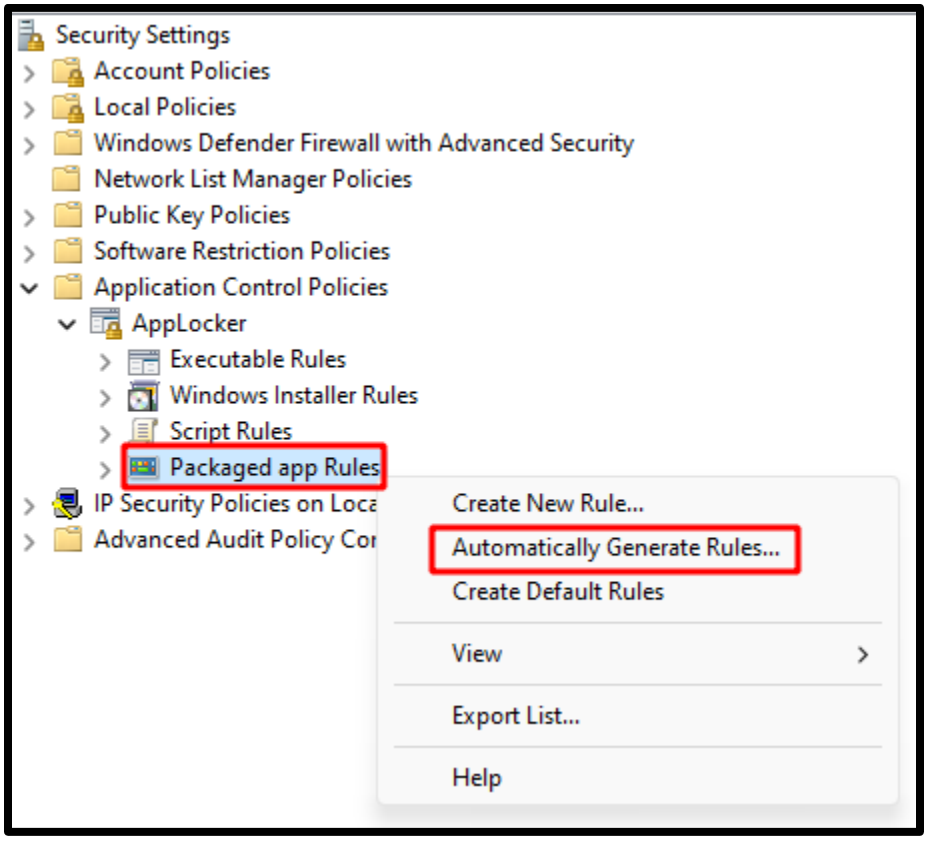

Figure 6. Automatically Generate Packaged app Rules

This method allows the usage of a Wizard to create these rules. For our testing purposes, the "Generate rules for all packaged apps installed on this computer" was selected, which allows a baseline of trusted packaged applications to be established at a point in time. This baseline can be expanded as necessary in the future. However, an overall recommendation is to establish a reasonable baseline of applications on a host by user profile before generating these AppLocker packaged application rules.

3.3.2. WDAC, AppLocker Executable Rules, and AppLocker Packaged app Rules

After these AppLocker Packaged app Rules were generated, the usability testing of 11_Base was re-performed, with results in Table 6 below.

Usability Activity	Usability Impacted?
Installs applications through the Microsoft Store	Impacted: Most applications except Games were installed.
Installs applications through a Web browser	No Impact
Customizes the host	No Impact
Uses the media player	No Impact
Creates/organizes files in the file system	No Impact

Table 6. 11_Base Useability Results

Usability was still considered Impacted. However, it was minimal compared to previous testing without the Packaged app Rules. For some enterprises, the impact would be beneficial as non-productivity applications like Games could not be downloaded even through the Microsoft App Store. This result reiterates the first conclusion that a baseline of acceptable user profile applications should be established before generating the AppLocker Packaged app allow-list Rules.

In addition to useability testing, effectiveness testing was performed with these application control configurations to prevent LOLBin usage. The conclusions of 11_Base testing are found in Table 7 below.

LOLBin	11_Base Prevention Status	LOLBin	11_Base Prevention Status

Tasklist	Successful	Ping	Successful
Ipconfig	Successful	Rundll32	Successful
Expand	Successful	Reg	Successful
Systeminfo	Successful	Wscript	Successful
Cmd.exe	Successful	Xcopy	Successful
PowerShell	Successful	Net	Successful
WMIC	Successful		

Table 7. 11_Base LOLBin Prevention Testing Results

3.4. 11_N_Tech Testing Results

A new Windows 11 OS was installed with the Microsoft Office Suite. The WDAC policy was established at the "Default Windows Mode" level with all identified LOLBins denied through AppLocker by their file hash as well as the same AppLocker Packaged app Rules that were configured for the 11_Base with the addition of the Microsoft Office Suite in the Packaged app Rules.

The same LOLBin prevention and host usability testing was performed on 11_N_Tech. To determine the level of 11_N_Tech host usability impact, the following five everyday activities that a typical non-technical user would use their Windows OS during a regular business day were performed: using Outlook for email communication, using virtual meeting software, using Teams for collaboration, document management on the local file system as well as OneDrive and SharePoint, usage of Microsoft Excel and PowerPoint for business administration tasks. The usability score would be "No Impact"

if all activities could be performed without impact. The usability score would be "Impacted" if one or two activities were impacted. If more than two activities were impacted, the usability score would be "Severely Impacted."

Like the 11_Base testing, since the command and PowerShell prompt were turned off through AppLocker configuration, AppLocker deny rules for all but the command prompt were created to test if all other LOLBins were successfully blocked. After this testing, the deny command prompt rule was established to test that this final LOLBin was blocked. Usability testing was then performed.

Usability testing with these WDAC and AppLocker configurations to deny execution of all identified LOLBins based on hash for the pre-identified 11_N_Tech activities is summarized in Table 8 below.

Usability Activity	**Usability Impacted?**
Outlook for email communication	No Impact
Virtual meeting software for audio/video/screen sharing	No Impact
Teams for collaboration	No Impact
Document management on the local file system, as well as OneDrive and SharePoint	No Impact
Microsoft Excel and PowerPoint for business administration tasks	No Impact

Table 8. 11_N_Tech Useability Results

Usability was not considered impacted. In addition to useability testing, effectiveness testing was performed with these application control configurations to prevent LOLBin usage. The conclusions of 11_N_Tech LOLBin prevention testing are found in Table 9 below.

LOLBin	**11_N_Tech Prevention Status**	**LOLBin**	**11_N_Tech Prevention Status**
Tasklist	Successful	Ping	Successful
Ipconfig	Successful	Rundll32	Successful
Expand	Successful	Reg	Successful
Systeminfo	Successful	Wscript	Successful
Cmd.exe	Successful	Xcopy	Successful
PowerShell	Successful	Net	Successful
WMIC	Successful		

Table 9. 11_N_Tech LOLBin Prevention Testing Results

3.5. 11_Tech Testing Results

A new Windows 11 OS was installed with standard utilities that technical employees would use in an enterprise. These applications included Visual Studio, Putty SSH client, Docker, PowerShell/PowerShell ISE (already included), and Git. The WDAC policy was established at the "Default Windows Mode" level with all identified LOLBins denied

through AppLocker by their file hash as well as the same AppLocker Packaged app Rules that were configured for the 11_Base with the addition of the pre-identified technical user applications in the Packaged app Rules. Since PowerShell was already identified as an application that a technical enterprise user would use, 11_Tech testing did not block the execution of PowerShell in AppLocker rules.

The same LOLBin prevention and host usability testing was performed on 11_Tech. To determine the level of 11_Tech host usability impact, the following five everyday activities that a typical technical user would use the pre-identified applications on a Windows OS for during a regular business day were performed: connecting to an SSH server with an SSH client, write and compile code with an IDE, push code to a cloud repository, write and execute a PowerShell script, download and run a Docker container. The usability score would be "No Impact" if all activities could be performed without impact. The usability score would be "Impacted" if one or two activities were impacted. If more than two activities were impacted, the usability score would be "Severely Impacted."

Like the 11_Base and 11_N_Tech testing, since the command prompt was turned off through AppLocker configuration, AppLocker deny rules for all but the command prompt were created to test if all other LOLBins were successfully blocked. After this testing, the deny command prompt rule was established to test that this final LOLBin was blocked. Usability testing was then performed.

Usability testing with these WDAC and AppLocker configurations to deny execution of all identified LOLBins based on hash for the pre-identified 11_Tech activities is summarized in Table 10 below.

Usability Activity	**Usability Impacted?**

Connect to an SSH server with an SSH client	No Impact
Write and compile code with an IDE	No Impact
Push code to a cloud repository	No Impact
Write and execute a PowerShell script	No Impact
Download and run a Docker container	No Impact

Table 10. 11_Tech Useability Results

Usability was not considered Impacted. In addition to useability testing, effectiveness testing was performed with these application control configurations to prevent LOLBin usage. The conclusions of 11_Tech testing are found in Table 11 below.

LOLBin	**11_Tech Prevention Status**	**LOLBin**	**11_Tech Prevention Status**
Tasklist	Successful	Ping	Successful
Ipconfig	Successful	Rundll32	Successful
Expand	Successful	Reg	Successful
Systeminfo	Successful	Wscript	Successful
Cmd.exe	Successful	Xcopy	Successful

PowerShell	N/A	Net	Successful
WMIC	Successful		

Table 11. 11_Tech LOLBin Prevention Testing Results

4. Recommendations and Implications

This research demonstrates effective prevention of the most common LOLBin usage by technical and non-technical employees while maintaining the useability of a Windows OS. These prevention controls are built into currently supported versions of Windows and can be deployed through Group Policy configurations to various user profiles across a common enterprise.

4.1. Recommendations for Practice

Two built-in Windows tools, WDAC and AppLocker, were used to achieve the prevention results. This research identified nuances in using both tools that would be valuable for professionals initially configuring these tools.

The WDAC tool is powerful and actively receiving new features and functionality from Microsoft. WDAC policy development is a deep, technical topic that requires nuance to deploy appropriately to organizations. Microsoft provides a detailed guide (jsuther1974 & vinaypamnani-msft, 2023) for planning, designing, creating, and deploying WDAC policies to an organization. This guide should be followed when considering WDAC as an application control solution, as it includes business and technical recommendations that increase the likelihood of a successful WDAC implementation. Finally, WDAC implementation is not a project with a defined beginning and end; it is a continuous process that demands consistent evaluation and configuration that aligns with how the internal enterprise and the external threat landscape change.

AppLocker is less powerful but more straightforward than the WDAC tool. The difficulty with AppLocker is the proper initial configuration to have it operate as designed. When configuring AppLocker, the enterprise should have a baseline of acceptable applications for each defined user profile. Once this baseline of allowed applications is installed on a baseline host for each user profile, the enterprise can generate the baseline AppLocker Packaged app allow-list Rules to ensure the usability of those applications. Additionally, the AppLocker LOLBin deny-list rules are static and may change with updates to the Windows OS. As such, testing the effectiveness of AppLocker LOLBin deny-list rules and Packaged app allow-list Rules should be integrated into the testing procedures for new OS image updates distributed to enterprise users. This testing will demonstrate if changes are needed to the AppLocker configurations in future versions of the Windows OS. Finally, AppLocker rules are only effective if the "AppIDSvc" service is running on the host. Therefore, monitoring each host with AppLocker rules configured is necessary to ensure the rules operate effectively. Whenever this service is turned off, it should be cause for investigation by the security team.

Like other security tools, application control technologies should be part of a broader defense-in-depth strategy. There are known bypasses for WDAC policies (jsuther1974 & vinaypamnani-msft, 2023) as well as other LOLBins that can be used to achieve the same result as the most common LOLBins that were tested. The tested application control technologies are intended to be a baseline configuration that prevents the most common LOLBin usage by APTs and should be supplemented based on the risk tolerance, business objectives, and threat landscape uniquely affecting each enterprise.

4.2. Implications for Future Research

There is an opportunity for further research on this topic, as more needs to be done in the public domain on modern-day application control approaches. The following list the areas of opportunities that became most prominent during this research.

The user profiles established for this research were simplistic and without deep nuance. More than two user profiles should be selected for an enterprise of any material size. Future application control research could investigate a broader set of user profiles that can provide further refinement to application control configurations and useability testing.

The testing environment compromised one version of Windows 11 with minor differences in installed applications. Testing of other operating systems was outside the scope of this research effort. The most likely candidates for further research include Windows Domain Controllers and Windows Internet Information Services (IIS) Servers, commonly targeted and compromised by APTs.

The useability testing criteria were established through personal experience. However, there is an opportunity for defining broader and nuanced useability testing criteria to evaluate further how effective these application control configurations are at allowing expected behavior while disallowing LOLBin usage. Additionally, other useability testing tools are available in the public domain (Airlock Digital, 2023) that can be used and expanded upon for more granular and expansive useability testing to meet the goals of each enterprise's application control approach.

5. Conclusion

LOLBin usage by APTs has been increasingly identified by both the private sector and governmental entities. Evidence from these intrusions shows that many current enterprises still need to implement essential application controls to prevent the execution of modern-day LOLBins. Additionally, there needs to be more modern research on the configuration, testing, and implementation of application control technologies. This research shows that configuring and implementing application control policies through built-in Windows tools can effectively prevent LOLBin usage while also allowing for the useability of the OS not to be materially impacted. However, application control should

be one of many security controls employed by an enterprise to achieve a resilient security posture. A key learning from this research is that adequate planning is necessary before application control deployment. Defining different user profiles that accurately represent the enterprise's groups of end users will help create nuanced application control configurations that strive to balance useability and protection.

References

Barr-Smith, F., Ugarte-Pedrero, X., Graziano, M., Spolaor, R., & Martinovic, I. (2021). Survivalism: Systematic Analysis of Windows Malware Living-Off-The-Land. 2021 IEEE Symposium on Security and Privacy (SP), 1557-1574.

Brown, D. (2020, January 6). Preventing Living off the Land Attacks. SANS Whitepapers, 1-33. Retrieved from https://www.sans.org/white-papers/39450/

Cottingham, D., & Schell, D. (2023, June 26). Risky Biz Soap Box: Defeating Living of the Land. (P. Gray, Interviewer) Retrieved from https://risky.biz/soapbox77/

Cybersecurity and Infrastructure Security Agency. (2023, November 16). Scattered Spider. Retrieved from Cybersecurity and Infrastructure Security Agency: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-320a

jsuther1974, & vinaypamnani-msft. (2023, August 02). Applications that can bypass WDAC and how to block them. Retrieved from Microsoft: https://learn.microsoft.com/en-us/windows/security/application-security/application-control/windows-defender-application-control/design/applications-that-can-bypass-wdac

Mandiant. (2023, September 01). Review of M-Trends 2023. Retrieved from Mandiant: https://mandiant.widen.net/s/dlzgn6w26n/m-trends-2023

Moe, O. (2018, October 22). Living Off The Land Binaries, Scripts and Libraries. Retrieved from LOLBAS: https://lolbas-project.github.io/

Mostafa, A. (2022). No-Budget Living-Off-the-Land Detection. SANS Whitepapers, 1-28. Retrieved from https://www.sans.org/white-papers/no-budget-living-off-the-land-detection/

Red Canary. (2023, May 01). Review of 2023 Threat Detection Report. Retrieved from Red Canary: https://redcanary.com/resources/guides/threat-detection-report/

Store, J. (2023). Living Off the Land as a Defender: Detecting Attacks with Flexible Baselines. SANS Whitepapers, 1-36. Retrieved from https://www.sans.org/white-papers/living-off-the-land-as-a-defender-detecting-attacks-with-flexible-baselines/

United States and International Cybersecurity Authorities. (2023). People's Republic of China State-Sponsored Cyber Actor Living off the Land to Evade Detection. Retrieved from United States Department of Defense: https://media.defense.gov/2023/May/24/2003229517/-1/-1/0/CSA_PRC_State_Sponsored_Cyber_Living_off_the_Land_v1.1.PDF