Microsoft

The Essential Eight for Security in Practice

Application Control

Kenny Singh (CISSP CCSP CISM)
Product Manager, Cyber Security and Compliance
Kenny.singh@microsoft.com

Agenda

Introduction

Essential Eight and Maturity Model

Microsoft Approach

New capabilities

Next steps

Strategies to Mitigate Cyber Security Incidents

"...the new cyber security baseline for all organisations"

"Organisations need to identify their assets and perform a risk assessment to identify the level of protection required from various cyber threats"

	Relative Security Effectiveness Rating				
Mitigation Strategy	**Essential**	**Excellent**	**Very Good**	**Good**	**Limited**
Prevent Malware Delivery and Execution	**4**	5	5	1	2
Limit the Extent of Cyber Security Incidents	**3**	3	4		
Detect Cyber Security Incidents and Respond		1	3		2
Recover Data and System Availability	**1**		2		
Preventing Malicious Insiders			1		

Total of 37 Controls

Prioritise control implementation

https://www.cyber.gov.au/publications/strategies-to-mitigate-cyber-security-incidents

The Essential Eight

Relative Security Effectiveness Rating	Mitigation Strategy	Potential User Resistance	Upfront Cost (staff, software and hardware)	Ongoing Maintenance Cost
Mitigation Strategies to Prevent Malware Delivery and Execution				
Essential	Application Whitelisting	Medium	High	Medium
Essential	Patch Applications	Low	High	High
Essential	Configure Microsoft Office macro settings	Medium	Medium	Medium
Essential	User Application Hardening	Medium	Medium	Medium
Mitigation Strategies to Limit the Extent of Cyber Security Incidents				
Essential	Restrict Administrative Privileges	Medium	High	Medium
Essential	Patch Operating Systems	Low	Medium	Medium
Essential	Multi-factor Authentication	High	High	Medium
Mitigation Strategies to Recover Data and System Availability				
Essential	Daily Backups	Low	High	High

The Essential Eight Maturity Model

Maturity Level	Description
Maturity Level 1	Partly aligned with the intent of the mitigation strategy
Maturity Level 2	Mostly aligned with the intent of the mitigation strategy
Maturity Level 3	Fully aligned with the intent of the mitigation strategy

"Once organisations have implemented their desired mitigation strategies to an initial level, they should focus on increasing the maturity of their implementation such that they eventually reach full alignment with the intent of each mitigation strategy."

Reference - https://www.cyber.gov.au/acsc/view-all-content/publications/essential-eight-maturity-model (June 2020)

Application Allow listing (Whitelisting)

Relative Security Effectiveness Rating	Potential User Resistance	Upfront Cost (staff, software and hardware)	Ongoing Maintenance Cost
Essential	Medium	High	Medium

Application allow listing

Mitigation Strategies to Prevent Malware Delivery and Execution

What application allow listing is	What application allow listing is not
Application allow listing is a security approach designed to protect against malicious code (also known as malware) executing on systems. When implemented properly it ensures that only approved applications (e.g. executables, software libraries, scripts and installers) can be executed. While application whitelisting is **primarily designed to prevent the execution and spread of malicious code**, it can also prevent the installation or use of unapproved applications.	**The following approaches are not considered to be application allow listing**: • providing a portal or other means of installation for approved applications • using web or email content filters to prevent users from downloading applications from the internet • checking the reputation of an application using a cloud-based service before it is executed • using a next-generation firewall to identify whether network traffic is generated by an approved application.

ACSC guidance - https://www.cyber.gov.au/publications/implementing-application-whitelisting

Application Control

Mitigation Strategies to Prevent Malware Delivery and Execution

Maturity Level 1	Maturity Level 2	Maturity Level 3
Application Control is implemented on all **workstations** to restrict the execution of executables to an approved set. Application control is implemented on all **servers** to restrict the execution of executables to an approved set.	Application control solution is implemented on all **workstations** to restrict the execution of executables, **software libraries, scripts and installers to an approved set.** An application control is implemented on all **servers** to restrict the execution of executables, **software libraries, scripts and installers to an approved set.**	An application control solution is implemented on all **workstations** to restrict the execution of executables, software libraries, scripts and installers to an approved set. An application whitelisting solution is implemented on all **servers** to restrict the execution of executables, software libraries, scripts and installers to an approved set. **Microsoft's latest recommended block rules are implemented to prevent application whitelisting bypasses.**

Reference https://www.cyber.gov.au/acsc/view-all-content/publications/essential-eight-maturity-model

AppLocker and Defender Application Control

WDAC is best when:

- You are adopting application control primarily for security reasons.
- Your application control policy can be applied to all users on the managed computers.
- All of the devices you wish to manage are running Windows 10.

VS

AppLocker is best when:

- You have a mixed Windows operating system (OS) environment.
- You need to apply different policies for different users or groups on a shared computer.
- You are using application control to help users avoid running unapproved software, but you do not require a solution designed as a security feature.
- You do not wish to enforce application control on application files such as DLLs or drivers.

TOGETHER

AppLocker can also be deployed as a complement to WDAC to add user- or group-specific rules for shared device scenarios where its important to prevent some users from running specific apps.

As a best practice, you should enforce WDAC at the most restrictive level possible for your organization, and then you can use AppLocker to fine-tune the restrictions to an even lower level.

Comparing Defender Application Control and AppLocker

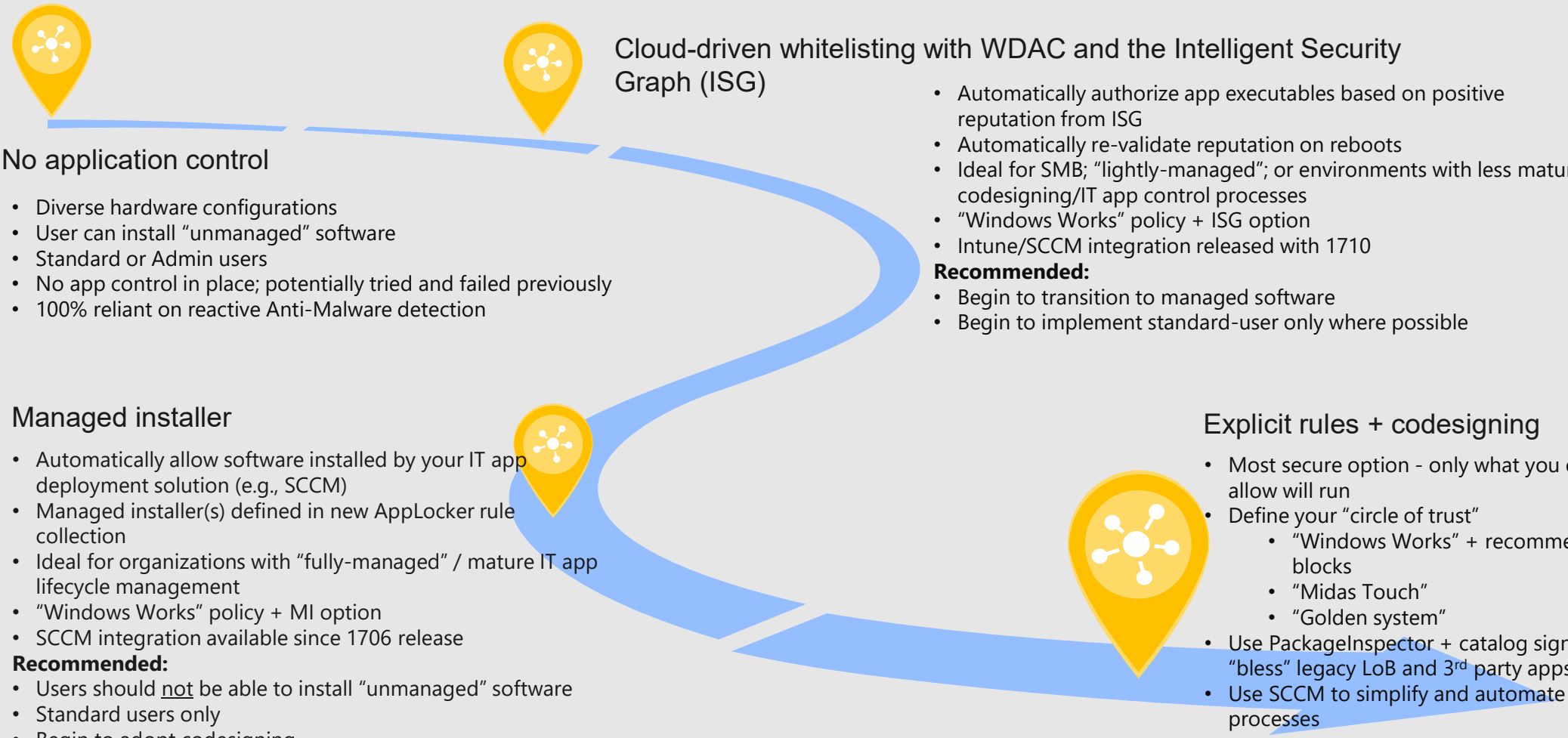
Capability	Application Control	AppLocker
Supported OS	Windows 10 Windows Server 2016 Windows Server 2019	Windows 7+
Policy Management	Intune (limited/custom), SCCM, Powershell, GPO*	Intune (limited/custom), SCCM, Powershell, GPO*
Policy assignment	Assigned to Device (affects all users)	Assigned to User/Groups or Devices
Supported policy attributes	Software libraries, scripts and installers to an approved set	Software libraries, scripts and installers to an approved set
Reputation-based execution analysis	Yes	No
Managed Installer Support	Yes	No
Local Admin Protection	Yes	No

https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/windows-defender-application-control

AppLocker / WDAC comparison

	Windows Defender Application Control		AppLocker	
	Win10/WS2016 or newer		Win17/WS2008R2 or newer	☑
☑	*Some* enforcement in hypervisor		Enforcement in Windows kernel	
☑	Designed to *also* handle drivers, services		Designed to constrain interactive session exec.	
	One machine-wide policy for all users (incl. admins)		Different rules for different users/groups	☑
	Rules for signed files: publisher, metadata	↔	Rules for signed files: publisher, metadata	
	Rules for unsigned files: catalog sign		Rules for unsigned files: hash of file	☑
	Can't do this (yet)		Rules based on file system path	☑
☑	Block a specific EXE from loading a specific DLL		Can’t do this (EMET could)	
☑	Full Windows investment		Very little ongoing Windows investment	
	Level of effort (real-world): massive PITA		Level of effort without “AaronLocker”: massive PITA Level of effort *with* “AaronLocker”: easy	☑

Getting to Secure: Achieving lockdown



No application control

- Diverse hardware configurations
- User can install "unmanaged" software
- Standard or Admin users
- No app control in place; potentially tried and failed previously
- 100% reliant on reactive Anti-Malware detection

Cloud-driven whitelisting with WDAC and the Intelligent Security Graph (ISG)

- Automatically authorize app executables based on positive reputation from ISG
- Automatically re-validate reputation on reboots
- Ideal for SMB; "lightly-managed"; or environments with less mature codesigning/IT app control processes
- "Windows Works" policy + ISG option
- Intune/SCCM integration released with 1710

Recommended:

- Begin to transition to managed software
- Begin to implement standard-user only where possible

Managed installer

- Automatically allow software installed by your IT app deployment solution (e.g., SCCM)
- Managed installer(s) defined in new AppLocker rule collection
- Ideal for organizations with "fully-managed" / mature IT app lifecycle management
- "Windows Works" policy + MI option
- SCCM integration available since 1706 release

Recommended:

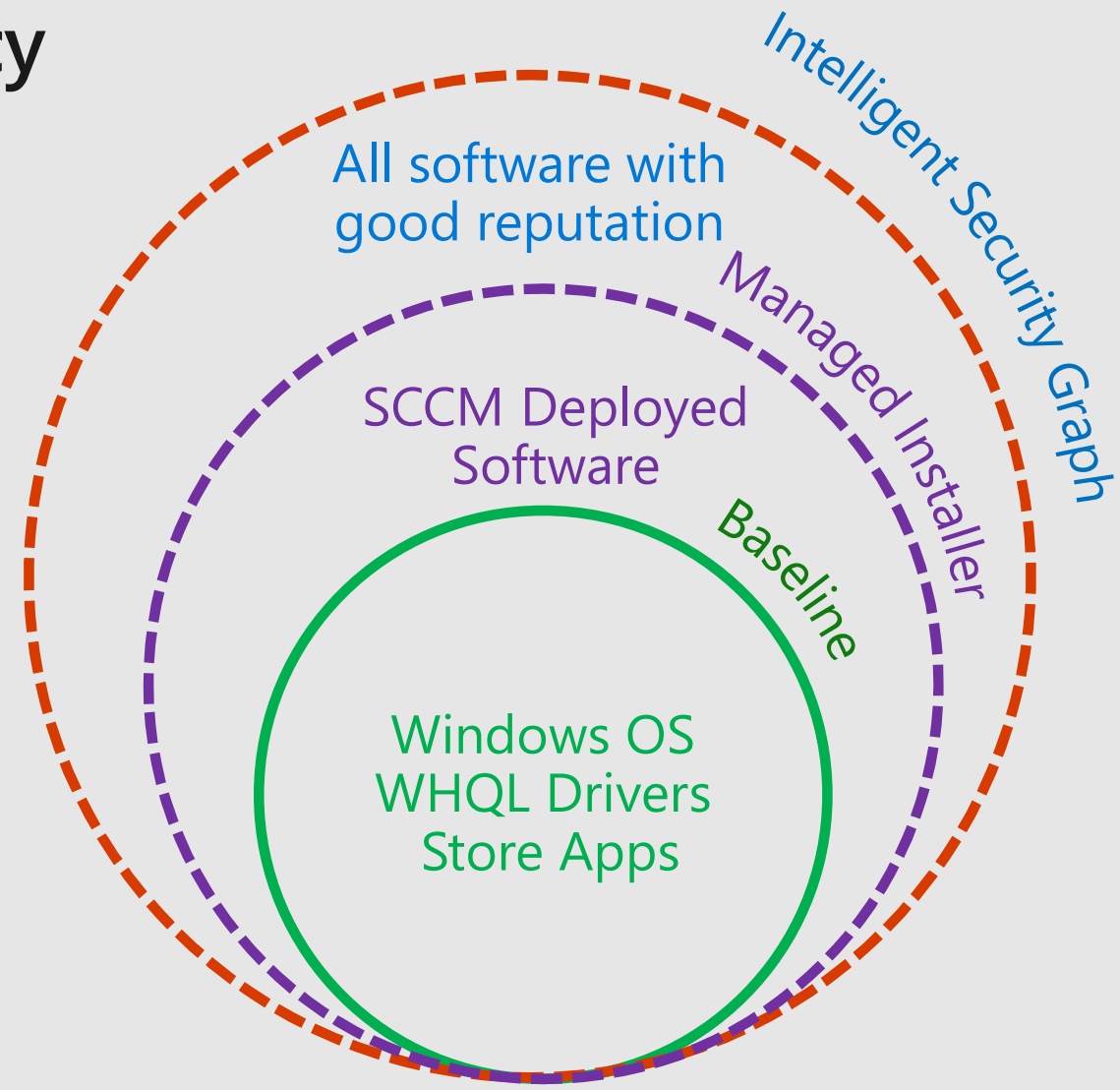
- Users should <u>not</u> be able to install "unmanaged" software
- Standard users only
- Begin to adopt codesigning

Explicit rules + codesigning

- Most secure option - only what you explicitly allow will run
- Define your "circle of trust"
  - "Windows Works" + recommended blocks
  - "Midas Touch"
  - "Golden system"
- Use PackageInspector + catalog signing to "bless" legacy LoB and 3rd party apps
- Use SCCM to simplify and automate processes

Circle of Trust: SCCM policy

- Baseline of trust allows basics
- Managed Installer allows SCCM deployed apps
- Allowing apps with good reputation widens trust palette for better or for worse



What is the Ideal AppLocker rule design

Allow execution from %windir% and %ProgramFiles%

- Except for user-writable subdirectories
- Except for common bypasses (e.g., mshta.exe)
- Except for abused tools (e.g., cipher.exe)

Allow execution from additional "safe" paths (e.g., logon scripts)

Allow approved code in user-writable directories

- E.g., OneDrive, Microsoft Teams (in user profile)

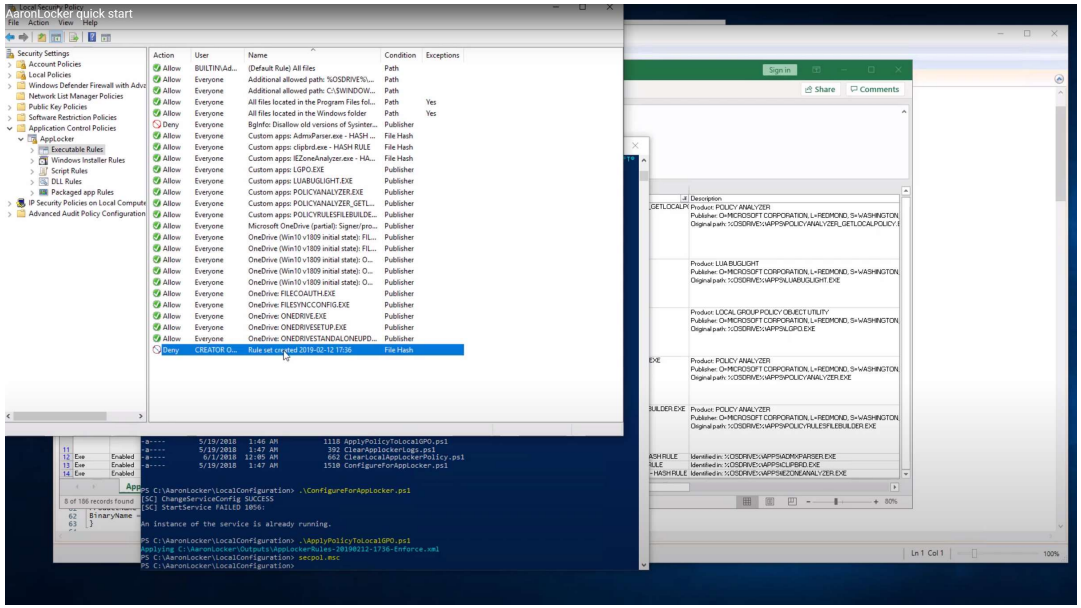
Solution Accelerator

AaronLocker - [GitHub - microsoft/AaronLocker: Robust and practical application control for Windows]

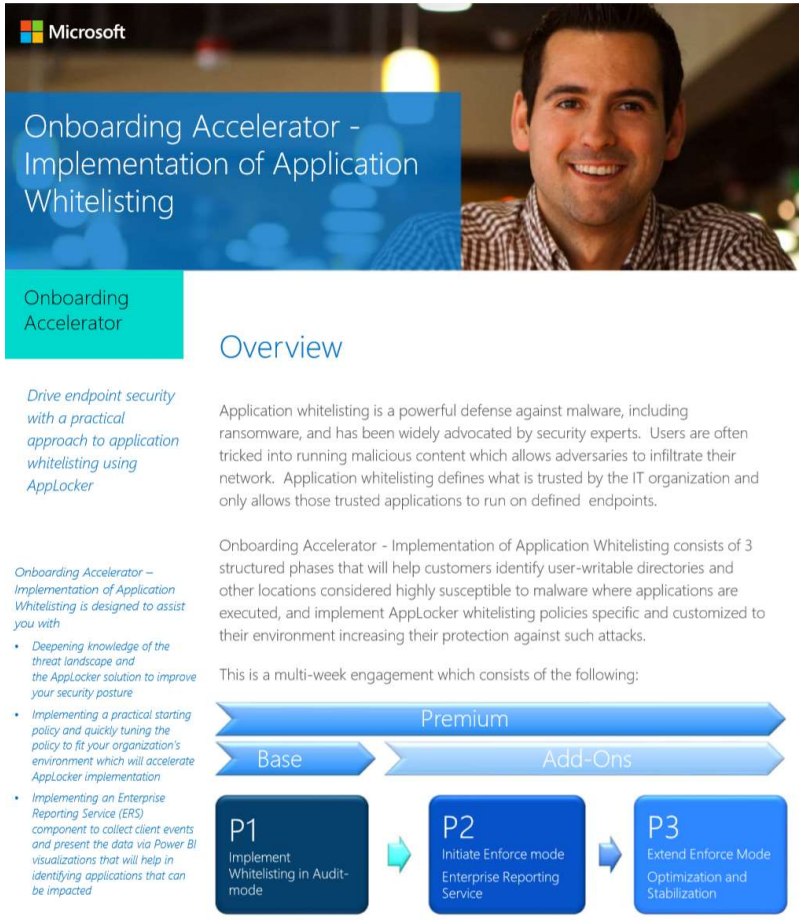
AaronLocker includes scripts to synthesize event data

As you analyze the data, ask the following questions:

- Is this a valid program?
- Is it possible to change the program to work with the existing ruleset?
- Are the files in a “safe” directory?
- Are the files signed



How can Microsoft help



Useful Resources and References

- Defender Application Control - https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/windows-defender-application-control
- AppLocker - https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/applocker-overview
- AaronLocker - https://github.com/Microsoft/AaronLocker
- Adaptive Application Control - https://docs.microsoft.com/en-us/azure/security-center/security-center-adaptive-application