Security baseline (FINAL) for Windows 10 v1903 and Windows Server v1903

Microsoft is pleased to announce the *final* release of the security configuration baseline settings for Windows 10 version 1903 (a.k.a., “19H1”), and for Windows Server version 1903.

Download the content from the Microsoft Security Compliance Toolkit (click Download and select Windows 10 Version 1903 and Windows Server Version 1903 Security Baseline.zip).

Note that Windows Server version 1903 is Server Core only and does not offer a Desktop Experience (a.k.a., “full”) server installation option. In the past we have published baselines only for “full” server releases – Windows Server 2016 and 2019. Beginning with this release we intend to publish baselines for Core-only Windows Server versions as well. However, we do not intend at this time to distinguish settings in the baseline that apply only to Desktop Experience. When applied to Server Core, those settings are inert for all intents and purposes.

This new Windows Feature Update brings very few new Group Policy settings, which we list in the accompanying documentation. This baseline recommends configuring only two of those. However, we have made several changes to existing settings, including some changes since the draft version of this baseline that we published last month.

The changes from the Windows 10 v1809 and Windows Server 2019 baselines include:

* Enabling the new “Enable svchost.exe mitigation options” policy, which enforces stricter security on Windows services hosted in svchost.exe, including that all binaries loaded by svchost.exe must be signed by Microsoft, and that dynamically-generated code is disallowed. ***Please pay special attention to this one*** as it might cause compatibility problems with third-party code that tries to use the svchost.exe hosting process, including third-party smart-card plugins.
* Configuring the new App Privacy setting, “Let Windows apps activate with voice while the system is locked,” so that users cannot interact with applications using speech while the system is locked.
* Disabling multicast name resolution (LLMNR) to mitigate server spoofing threats.
* Restricting the NetBT NodeType to P-node, disallowing the use of broadcast to register or resolve names, also to mitigate server spoofing threats. We have added a setting to the custom “MS Security Guide” ADMX to enable managing this configuration setting through Group Policy.
* Correcting an oversight in the Domain Controller baseline by adding recommended auditing settings for Kerberos authentication service.
* Dropping the password-expiration policies that require periodic password changes. This change is discussed in further detail below.
* Dropping the specific BitLocker drive encryption method and cipher strength settings. The baseline has been requiring the strongest available BitLocker encryption. We are removing that item for a few reasons. The default is 128-bit encryption, and our crypto experts tell us that there is no known danger of its being broken in the foreseeable future. On some hardware there can be noticeable performance degradation going from 128- to 256-bit. And finally, many devices such as those in the Microsoft Surface line turn on BitLocker by default and use the default algorithms. Converting those to use 256-bit requires first decrypting the volumes and then re-encrypting, which creates temporary security exposure as well as user impact.
* Dropping the File Explorer “Turn off Data Execution Prevention for Explorer” and “Turn off heap termination on corruption” settings, as it turns out they merely enforce default behavior, as Raymond Chen describes [here](https://devblogs.microsoft.com/oldnewthing/20170620-00/?p=96435).

Additional changes that we have adopted since publishing the draft version of this baseline include:

* Dropping the enforcement of the default behavior of disabling the built-in Administrator and Guest accounts. We had floated this proposal at the time of the draft baseline, and have since decided to accept it. The change is discussed in more detail below.
* Dropped a Windows Defender Antivirus setting that applies only to legacy email file formats.
* Changed the Windows Defender Exploit Protection XML configuration to allow Groove.exe (OneDrive for Business) to launch child processes, particularly MsoSync.exe which is necessary for file synchronization.

***Dropping the password expiration policies.***

There’s no question that the state of password security is problematic and has been for a long time. When humans pick their own passwords, too often they are easy to guess or predict. When humans are assigned or forced to create passwords that are hard to remember, too often they’ll write them down where others can see them. When humans are forced to change their passwords, too often they’ll make a small and predictable alteration to their existing passwords, and/or forget their new passwords. When passwords or their corresponding hashes are stolen, it can be difficult at best to detect or restrict their unauthorized use.

Recent scientific research calls into question the value of many long-standing password-security practices such as password expiration policies, and points instead to better alternatives such as enforcing banned-password lists (a great example being [Azure AD password protection](https://techcommunity.microsoft.com/t5/Azure-Active-Directory-Identity/Azure-AD-Password-Protection-is-now-generally-available/ba-p/377487)) and multi-factor authentication. While we recommend these alternatives, they cannot be expressed or enforced with our recommended security configuration baselines, which are built on Windows’ built-in Group Policy settings and cannot include customer-specific values.

This reinforces a larger important point about our baselines: while they are a solid foundation and should be *part* of your security strategy, they are not a *complete* security strategy. In this particular case, the small set of ancient password policies enforceable through Windows’ security templates is not and cannot be a complete security strategy for user credential management. Removing a low-value setting from our baseline and not compensating with something else in the baseline does not mean we are lowering security standards. It simply reinforces that security cannot be achieved entirely with baselines.

***Why are we removing password-expiration policies?***

*First, to try to avoid inevitable misunderstandings, we are talking here* only *about removing password-expiration policies – we are not proposing changing requirements for minimum password length, history, or complexity.*

Periodic password expiration is a defense *only* against the probability that a password (or hash) will be stolen during its validity interval and will be used by an unauthorized entity. If a password is never stolen, there’s no need to expire it. And if you have evidence that a password has been stolen, you would presumably act immediately rather than wait for expiration to fix the problem.

If it’s a given that a password is likely to be stolen, how many days is an acceptable length of time to continue to allow the thief to use that stolen password? The Windows default is 42 days. Doesn’t that seem like a *ridiculously long time*? Well, it is, and yet our current baseline says 60 days – and used to say 90 days – because forcing frequent expiration introduces its own problems. And if it’s *not* a given that passwords will be stolen, you acquire those problems for no benefit. Further, if your users are the kind who are willing to answer surveys in the parking lot that exchange a candy bar for their passwords, no password expiration policy will help you.

Our baselines are intended to be usable with minimal if any modification by most well-managed, security-conscious enterprises. They are also intended to serve as guidance for auditors. So, what should the recommended expiration period be? If an organization has successfully implemented banned-password lists, multi-factor authentication, detection of password-guessing attacks, and detection of anomalous logon attempts, do they need *any* periodic password expiration? And if they haven’t implemented modern mitigations, how much protection will they *really* gain from password expiration?

The results of baseline compliance scans are usually measured by how many settings are out of compliance: “How much red do we have on the chart?” It is not unusual for organizations during audit to treat compliance numbers as more important than real-world security. If a baseline recommends 60 days and an organization with advanced protections opts for 365 days – or no expiration at all – they will get dinged in an audit unnecessarily and might be compelled to adhere to the 60-day recommendation.

Periodic password expiration is an ancient and obsolete mitigation of very low value, and we don’t believe it’s worthwhile for our baseline to enforce any specific value. By removing it from our baseline rather than recommending a particular value or no expiration, organizations can choose whatever best suits their perceived needs without contradicting our guidance. At the same time, we must reiterate that we *strongly* recommend additional protections even though they cannot be expressed in our baselines.

***Dropping the enforced disabling of the built-in Administrator and Guest accounts***

To keep baselines useful and manageable, we tend to enforce secure defaults for policy settings only when 1) non-administrative users could otherwise override those defaults, or 2) misinformed administrators are otherwise likely to make poor choices about the setting. Neither of those conditions are true regarding enforcing the default disabling of the Administrator and Guest accounts. Note that removing these settings from the baseline does *not* mean that we recommend that these accounts be enabled, nor does removing these settings mean that the accounts *will* be enabled. Removing the settings from the baselines simply means that administrators can now choose to enable these accounts as needed.

***The built-in Guest account.*** The Guest account (RID -501) is disabled by default on Windows 10 and Windows Server. Only an administrator can enable the Guest account, and an admin would presumably do so only for a valid reason such as for a kiosk system.

***The built-in Administrator account.*** The local Administrator account (RID -500) is disabled by default on Windows 10 but not on Windows Server. When installing Windows 10, Windows Setup prompts you for a new account which becomes the primary administrative account for the computer. By contrast, Windows Server’s setup prompts you for a new password for the Administrator account. The main differences between the built-in -500 Administrator account (when enabled) and a custom administrative local account are 1) the -500 account is not subject to account lockout, account expiration, password expiration, or logon hours; 2) the -500 account cannot be removed from the Administrators group; and 3) that by default the -500 account always runs with full administrative rights without UAC prompts, including over the network. This third difference can be removed (as our baselines always do) by enabling the security option, “User Account Control: Admin Approval Mode for the Built-in Administrator account.”

Our recommendations for administrative local accounts include:

* You can choose not to have any administrative local accounts enabled and to administer domain-joined systems only with domain accounts.
* If you choose to use local accounts for computer administration, you should have only one administrative local account enabled per computer. With this change in the baseline, you can choose to use the -500 Administrator account or a custom account, according to your needs. Note that if you rely on account lockout for defense against password-guessing attacks, you should *not* enable the -500 account – and you should consider disabling it on Windows Server.
* The administrative local account’s password should be strong and should be different from the same account’s password on every other computer. We recommend the Local Administrator Password Solution (LAPS) or a similar tool to ensure that passwords are random and strong. LAPS can manage the password of the -500 account or a custom named local account on Active Directory domain-joined Windows clients and domain-joined member servers (but not for Domain Controllers). Note also that LAPS’ password-expiration enforcement is independent from Windows’ password-expiration mechanism, and always applies to whatever account LAPS manages.
* Renaming the Administrator account is perfectly fine but is “security by obscurity.” Renaming is easy to do through Group Policy and doing so can mitigate some threats, but it’s less than a speedbump against other threats.