

CIS Visual Studio Code GPO Benchmark

v1.0.0 - 09-25-2025

Terms of Use

Please see the below link for our current terms of use:

<https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/>

For information on referencing and/or citing CIS Benchmarks in 3rd party documentation (including using portions of Benchmark Recommendations) please contact CIS Legal (legalnotices@cisecurity.org) and request guidance on copyright usage.

NOTE: It is **NEVER** acceptable to host a CIS Benchmark in **ANY** format (PDF, etc.) on a 3rd party (non-CIS owned) site.

Table of Contents

Terms of Use	1
Table of Contents	2
Overview	4
Important Usage Information	4
Key Stakeholders	4
Apply the Correct Version of a Benchmark	5
Exceptions	5
Remediation	6
Summary	6
Target Technology Details	7
Intended Audience.....	7
Consensus Guidance	8
Typographical Conventions.....	9
Recommendation Definitions.....	10
Title.....	10
Assessment Status.....	10
Automated	10
Manual.....	10
Profile	10
Description.....	10
Rationale Statement	10
Impact Statement.....	11
Audit Procedure.....	11
Remediation Procedure.....	11
Default Value.....	11
References	11
CIS Critical Security Controls® (CIS Controls®)	11
Additional Information.....	11
Profile Definitions	12
Acknowledgements	13
Recommendations	14
1 Visual Studio Code	14
1.1 Chat	15
1.1.1 (L2) Ensure 'ChatMCP' is set to 'Disabled' (Automated)	16
1.1.2 (L1) Ensure 'ChatToolsAutoApprove' is set to 'Disabled' (Automated)	18
1.2 Extensions	20

1.2.1 (L2) Ensure 'AllowedExtensions' is configured (Manual).....	21
1.3 Telemetry	23
1.3.1 (L2) Ensure 'EnableFeedback' is set to 'Disabled' (Automated).....	24
1.3.2 (L1) Ensure 'TelemetryLevel' is set to 'Enabled: off' (Automated).....	26
1.4 Update	28
1.4.1 (L1) Ensure 'UpdateMode' is set to 'Enabled: Enable Automatic Update Checks. Code will check for updates automatically and periodically.' (Automated).....	29
<i>Appendix: Summary Table</i>	<i>31</i>
<i>Appendix: Change History</i>	<i>32</i>

Overview

All CIS Benchmarks™ (Benchmarks) focus on technical configuration settings used to maintain and/or increase the security of the addressed technology, and they should be used in **conjunction** with other essential cyber hygiene tasks like:

- Monitoring the base operating system and applications for vulnerabilities and quickly updating with the latest security patches.
- End-point protection (Antivirus software, Endpoint Detection and Response (EDR), etc.).
- Logging and monitoring user and system activity.

In the end, the Benchmarks are designed to be a key **component** of a comprehensive cybersecurity program.

Important Usage Information

All Benchmarks are available free for non-commercial use from the [CIS Website](#). They can be used to manually assess and remediate systems and applications. In lieu of manual assessment and remediation, there are several tools available to assist with assessment:

- [CIS Configuration Assessment Tool \(CIS-CAT® Pro Assessor\)](#)
- [CIS Benchmarks™ Certified 3rd Party Tooling](#)

These tools make the hardening process much more scalable for large numbers of systems and applications.

NOTE: Some tooling focuses only on the Benchmark Recommendations that can be fully automated (skipping ones marked **Manual**). It is important that **ALL** Recommendations (**Automated** and **Manual**) be addressed since all are important for properly securing systems and are typically in scope for audits.

Key Stakeholders

Cybersecurity is a collaborative effort, and cross functional cooperation is imperative within an organization to discuss, test, and deploy Benchmarks in an effective and efficient way. The Benchmarks are developed to be best practice configuration guidelines applicable to a wide range of use cases. In some organizations, exceptions to specific Recommendations will be needed, and this team should work to prioritize the problematic Recommendations based on several factors like risk, time, cost, and labor. These exceptions should be properly categorized and documented for auditing purposes.

Apply the Correct Version of a Benchmark

Benchmarks are developed and tested for a specific set of products and versions and applying an incorrect Benchmark to a system can cause the resulting pass/fail score to be incorrect. This is due to the assessment of settings that do not apply to the target systems. To assure the correct Benchmark is being assessed:

- **Deploy the Benchmark applicable to the way settings are managed in the environment:** An example of this is the Microsoft Windows family of Benchmarks, which have separate Benchmarks for Group Policy, Intune, and Stand-alone systems based upon how system management is deployed. Applying the wrong Benchmark in this case will give invalid results.
- **Use the most recent version of a Benchmark:** This is true for all Benchmarks, but especially true for cloud technologies. Cloud technologies change frequently and using an older version of a Benchmark may have invalid methods for auditing and remediation.

Exceptions

The guidance items in the Benchmarks are called recommendations and not requirements, and exceptions to some of them are expected and acceptable. The Benchmarks strive to be a secure baseline, or starting point, for a specific technology, with known issues identified during Benchmark development are documented in the Impact section of each Recommendation. In addition, organizational, system specific requirements, or local site policy may require changes as well, or an exception to a Recommendation or group of Recommendations (e.g. A Benchmark could Recommend that a Web server not be installed on the system, but if a system's primary purpose is to function as a Webserver, there should be a documented exception to this Recommendation for that specific server).

In the end, exceptions to some Benchmark Recommendations are common and acceptable, and should be handled as follows:

- The reasons for the exception should be reviewed cross-functionally and be well documented for audit purposes.
- A plan should be developed for mitigating, or eliminating, the exception in the future, if applicable.
- If the organization decides to accept the risk of this exception (not work toward mitigation or elimination), this should be documented for audit purposes.

It is the responsibility of the organization to determine their overall security policy, and which settings are applicable to their unique needs based on the overall risk profile for the organization.

Remediation

CIS has developed [Build Kits](#) for many technologies to assist in the automation of hardening systems. Build Kits are designed to correspond to Benchmark's "Remediation" section, which provides the manual remediation steps necessary to make that Recommendation compliant to the Benchmark.

When remediating systems (changing configuration settings on deployed systems as per the Benchmark's Recommendations), please approach this with caution and test thoroughly.

The following is a reasonable remediation approach to follow:

- CIS Build Kits, or internally developed remediation methods should never be applied to production systems without proper testing.
- Proper testing consists of the following:
 - Understand the configuration (including installed applications) of the targeted systems. Various parts of the organization may need different configurations (e.g., software developers vs standard office workers).
 - Read the Impact section of the given Recommendation to help determine if there might be an issue with the targeted systems.
 - Test the configuration changes with representative lab system(s). If issues arise during testing, they can be resolved prior to deploying to any production systems.
 - When testing is complete, initially deploy to a small sub-set of production systems and monitor closely for issues. If there are issues, they can be resolved prior to deploying more broadly.
 - When the initial deployment above is completed successfully, iteratively deploy to additional systems and monitor closely for issues. Repeat this process until the full deployment is complete.

Summary

Using the Benchmarks Certified tools, working as a team with key stakeholders, being selective with exceptions, and being careful with remediation deployment, it is possible to harden large numbers of deployed systems in a cost effective, efficient, and safe manner.

NOTE: As previously stated, the PDF versions of the CIS Benchmarks™ are available for free, non-commercial use on the [CIS Website](#). All other formats of the CIS Benchmarks™ (MS Word, Excel, and [Build Kits](#)) are available for CIS [SecureSuite](#)® members.

CIS-CAT® Pro is also available to CIS [SecureSuite](#)® members.

Target Technology Details

This document provides prescriptive guidance for establishing a secure configuration posture for Visual Studio Code. This guide was tested against VS Code version 1.104 on a Windows Server 2025 operating system. To obtain the latest version of this guide, please visit <http://benchmarks.cisecurity.org>. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Intended Audience

This benchmark is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate Visual Studio Code via Group Policy Objects (GPOs).

Consensus Guidance

This CIS Benchmark™ was created using a consensus review process comprised of a global community of subject matter experts. The process combines real world experience with data-based information to create technology specific guidance to assist users to secure their environments. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS Benchmark undergoes two phases of consensus review. The first phase occurs during initial Benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the Benchmark. This discussion occurs until consensus has been reached on Benchmark recommendations. The second phase begins after the Benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the Benchmark. If you are interested in participating in the consensus process, please visit <https://workbench.cisecurity.org/>.

Typographical Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
<code>Stylized Monospace font</code>	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
<code>Monospace font</code>	Used for inline code, commands, UI/Menu selections or examples. Text should be interpreted exactly as presented.
<code><Monospace font in brackets></code>	Text set in angle brackets denote a variable requiring substitution for a real value.
<i>Italic font</i>	Used to reference other relevant settings, CIS Benchmarks and/or Benchmark Communities. Also, used to denote the title of a book, article, or other publication.
Bold font	Additional information or caveats things like Notes , Warnings , or Cautions (usually just the word itself and the rest of the text normal).

Recommendation Definitions

The following defines the various components included in a CIS recommendation as applicable. If any of the components are not applicable it will be noted, or the component will not be included in the recommendation.

Title

Concise description for the recommendation's intended configuration.

Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

Automated

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

Manual

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

Profile

A collection of recommendations for securing a technology or a supporting platform. Most benchmarks include at least a Level 1 and Level 2 Profile. Level 2 extends Level 1 recommendations and is not a standalone profile. The Profile Definitions section in the benchmark provides the definitions as they pertain to the recommendations included for the technology.

Description

Detailed information pertaining to the setting with which the recommendation is concerned. In some cases, the description will include the recommended value.

Rationale Statement

Detailed reasoning for the recommendation to provide the user a clear and concise understanding on the importance of the recommendation.

Impact Statement

Any security, functionality, or operational consequences that can result from following the recommendation.

Audit Procedure

Systematic instructions for determining if the target system complies with the recommendation.

Remediation Procedure

Systematic instructions for applying recommendations to the target system to bring it into compliance according to the recommendation.

Default Value

Default value for the given setting in this recommendation, if known. If not known, either not configured or not defined will be applied.

References

Additional documentation relative to the recommendation.

CIS Critical Security Controls® (CIS Controls®)

The mapping between a recommendation and the CIS Controls is organized by CIS Controls version, Safeguard, and Implementation Group (IG). The Benchmark in its entirety addresses the CIS Controls safeguards of (v7) "5.1 - Establish Secure Configurations" and (v8) "4.1 - Establish and Maintain a Secure Configuration Process" so individual recommendations will not be mapped to these safeguards.

Additional Information

Supplementary information that does not correspond to any other field but may be useful to the user.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1**

Items in this profile intend to:

- be the starting baseline for most organizations;
- be practical and prudent;
- provide a clear security benefit; and
- not inhibit the utility of the technology beyond acceptable means.

- **Level 2**

This profile extends the "Level 1" profile. Items in this profile exhibit one or more of the following characteristics:

- are intended for environments or use cases where security is more critical than manageability and usability;
- may negatively inhibit the utility or performance of the technology; and
- limit the ability of remote management/access.

Acknowledgements

This Benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Author

Kyle Bavis

Editor

Matthew Woods

Jennifer Jarose

Recommendations

1 Visual Studio Code

1.1 Chat

This section contains recommendations for chat (AI) settings.

1.1.1 (L2) Ensure 'ChatMCP' is set to 'Disabled' (Automated)

Profile Applicability:

- Level 2

Description:

This policy setting controls integration with Model Context Protocol (MCP) servers.

The recommended state for this setting is: **Disabled**.

Rationale:

Integration with third-party MCP servers may result in unintended disclosure of information to the party running the server.

Impact:

Users will not be able to use MCP server integrations.

Audit:

Navigate to the UI Path articulated in the Remediation section and confirm it is set as prescribed. This group policy setting is backed by the following registry location with a **REG_DWORD** value of **0**.

```
HKLM\SOFTWARE\Policies\Microsoft\VSCode:ChatMCP
```

Remediation:

To establish the recommended configuration via GP, set the following UI path to **Disabled**:

```
Computer Configuration\Policies\Administrative Templates\Visual Studio  
Code\Chat\ChatMCP
```

Note: This Group Policy path may not exist by default. It is provided by the Group Policy template **VScode.admx**, which is available in the VS Code installation directory under the **policies** folder.

Default Value:

Enabled

References:

1. https://code.visualstudio.com/docs/setup/enterprise#_centrally-manage-vs-code-settings

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			
v7	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			

1.1.2 (L1) Ensure 'ChatToolsAutoApprove' is set to 'Disabled' (Automated)

Profile Applicability:

- Level 1

Description:

This policy setting controls auto-approval for agent mode chat tools.

The recommended state for this setting is: **Disabled**.

Rationale:

Automatic activation of agent mode tools may result in inadvertent disclosure of sensitive data to third-party servers backing the chat tool or unexpected modification to project files.

Impact:

Users will need to manually approve agent mode before it can be used.

Audit:

Navigate to the UI Path articulated in the Remediation section and confirm it is set as prescribed. This group policy setting is backed by the following registry location with a **REG_DWORD** value of **0**.

```
HKLM\SOFTWARE\Policies\Microsoft\VSCode:ChatToolsAutoApprove
```

Remediation:

To establish the recommended configuration via GP, set the following UI path to **Disabled**:

```
Computer Configuration\Policies\Administrative Templates\Visual Studio Code\Chat\ChatToolsAutoApprove
```

Note: This Group Policy path may not exist by default. It is provided by the Group Policy template **VScode.admx**, which is available in the VS Code installation directory under the **policies** folder.

References:

1. https://code.visualstudio.com/docs/setup/enterprise#_centrally-manage-vs-code-settings

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			
v7	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			

1.2 Extensions

This section contains recommendations for extension settings.

1.2.1 (L2) Ensure 'AllowedExtensions' is configured (Manual)

Profile Applicability:

- Level 2

Description:

This policy setting configures a list of extension selectors that determine which extensions are allowed or blocked.

The recommended state for this setting is: **Enabled** with a list of allowed extensions.

Rationale:

Allowing users to install any extension may result in unintended information disclosure to third parties, or the installation of malware.

Impact:

IT administrators will need to maintain the list of allow-listed extensions. Depending on individual end-user needs, it may not be practical to allow specific extensions. Allowing by publisher may be more practical.

Audit:

Navigate to the UI Path articulated in the Remediation section and confirm it is set as prescribed. This group policy setting is backed by the following registry location with a **REG_SZ** value containing a JSON string.

```
HKLM\SOFTWARE\Policies\Microsoft\VSCode:AllowedExtensions
```

Remediation:

To establish the recommended configuration via GP, set the following UI path to **Enabled** along with a JSON string defining the approved extensions:

```
Computer Configuration\Policies\Administrative Templates\Visual Studio  
Code\Extensions\AllowedExtensions
```

Note: This Group Policy path may not exist by default. It is provided by the Group Policy template **VScode.admx**, which is available in the VS Code installation directory under the **policies** folder.

Default Value:

Null; a user may install any extension.

References:

1. https://code.visualstudio.com/docs/setup/enterprise#_configure-allowed-extensions

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			
v7	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			

1.3 Telemetry

This section contains recommendations for telemetry settings.

1.3.1 (L2) Ensure 'EnableFeedback' is set to 'Disabled' (Automated)

Profile Applicability:

- Level 2

Description:

This policy setting controls feedback mechanisms, such as the issue reporter and surveys.

The recommended state for this setting is: **Disabled**.

Rationale:

The VS Code issue reporter can be configured to submit context such as basic system configuration and application settings. This may inadvertently disclose information that some organizations consider confidential.

Impact:

Users will not be able to use in-application feedback tools.

Audit:

Navigate to the UI Path articulated in the Remediation section and confirm it is set as prescribed. This group policy setting is backed by the following registry location with a **REG_DWORD** value of **0**.

```
HKLM\SOFTWARE\Policies\Microsoft\VSCode:EnableFeedback
```

Remediation:

To establish the recommended configuration via GP, set the following UI path to **Disabled**:

```
Computer Configuration\Policies\Administrative Templates\Visual Studio Code\Telemetry\EnableFeedback
```

Note: This Group Policy path may not exist by default. It is provided by the Group Policy template **VScode.admx**, which is available in the VS Code installation directory under the **policies** folder.

Default Value:

Enabled

References:

1. https://code.visualstudio.com/docs/setup/enterprise#_centrally-manage-vs-code-settings

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			
v7	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			

1.3.2 (L1) Ensure 'TelemetryLevel' is set to 'Enabled: off' (Automated)

Profile Applicability:

- Level 1

Description:

This policy setting controls telemetry for VS Code and extensions.

The recommended state for this setting is: **Enabled: off**.

Rationale:

Enabling this feature sends data to Microsoft and potentially third-party extension developers, which could lead to sensitive data being exposed.

Impact:

Data will not be shared with Microsoft or extension developers.

Audit:

Navigate to the UI Path articulated in the Remediation section and confirm it is set as prescribed. This group policy setting is backed by the following registry location with a **REG_SZ** value of **off**.

```
HKLM\SOFTWARE\Policies\Microsoft\VSCode:TelemetryLevel
```

Remediation:

To establish the recommended configuration via GP, set the following UI path to **Enabled: off**:

```
Computer Configuration\Policies\Administrative Templates\Visual Studio  
Code\Telemetry\TelemetryLevel
```

Note: This Group Policy path may not exist by default. It is provided by the Group Policy template **VScode.admx**, which is available in the VS Code installation directory under the **policies** folder.

Default Value:

all (Crash reports, usage data, and error telemetry will be collected)

References:

1. https://code.visualstudio.com/docs/setup/enterprise#_configure-telemetry-level

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			
v7	0.0 <u>Explicitly Not Mapped</u> Explicitly Not Mapped			

1.4 Update

This section contains recommendations for update settings.

1.4.1 (L1) Ensure 'UpdateMode' is set to 'Enabled: Enable Automatic Update Checks. Code will check for updates automatically and periodically.' (Automated)

Profile Applicability:

- Level 1

Description:

This policy setting controls whether VS Code automatically updates when a new version is released.

The recommended state for this setting is: **Enabled: Enable Automatic Update Checks. Code will check for updates automatically and periodically.**

Rationale:

Software needs to be patched regularly to address vulnerabilities.

Impact:

None; this is the default behavior.

Audit:

Navigate to the UI Path articulated in the Remediation section and confirm it is set as prescribed. This group policy setting is backed by the following registry location with a **REG_SZ** value of **default**.

```
HKLM\SOFTWARE\Policies\Microsoft\VSCode:UpdateMode
```

Remediation:

To establish the recommended configuration via GP, set the following UI path to **Enabled: Enable Automatic Update Checks. Code will check for updates automatically and periodically.:**

```
Computer Configuration\Policies\Administrative Templates\Visual Studio  
Code\Update\UpdateMode
```

Note: This Group Policy path may not exist by default. It is provided by the Group Policy template **VSCode.admx**, which is available in the VS Code installation directory under the **policies** folder.







Default Value:

Default

References:

1. https://code.visualstudio.com/docs/setup/enterprise#_configure-automatic-updates

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>7.3 Perform Automated Operating System Patch Management</u> Perform operating system updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.			
v7	<u>3.4 Deploy Automated Operating System Patch Management Tools</u> Deploy automated software update tools in order to ensure that the operating systems are running the most recent security updates provided by the software vendor.			

Appendix: Summary Table

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
1	Visual Studio Code		
1.1	Chat		
1.1.1	(L2) Ensure 'ChatMCP' is set to 'Disabled' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
1.1.2	(L1) Ensure 'ChatToolsAutoApprove' is set to 'Disabled' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
1.2	Extensions		
1.2.1	(L2) Ensure 'AllowedExtensions' is configured (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
1.3	Telemetry		
1.3.1	(L2) Ensure 'EnableFeedback' is set to 'Disabled' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
1.3.2	(L1) Ensure 'TelemetryLevel' is set to 'Enabled: off' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
1.4	Update		
1.4.1	(L1) Ensure 'UpdateMode' is set to 'Enabled: Enable Automatic Update Checks. Code will check for updates automatically and periodically.' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

Appendix: Change History

Date: 09/25/2025 Version: 1.0.0
Initial Public Release