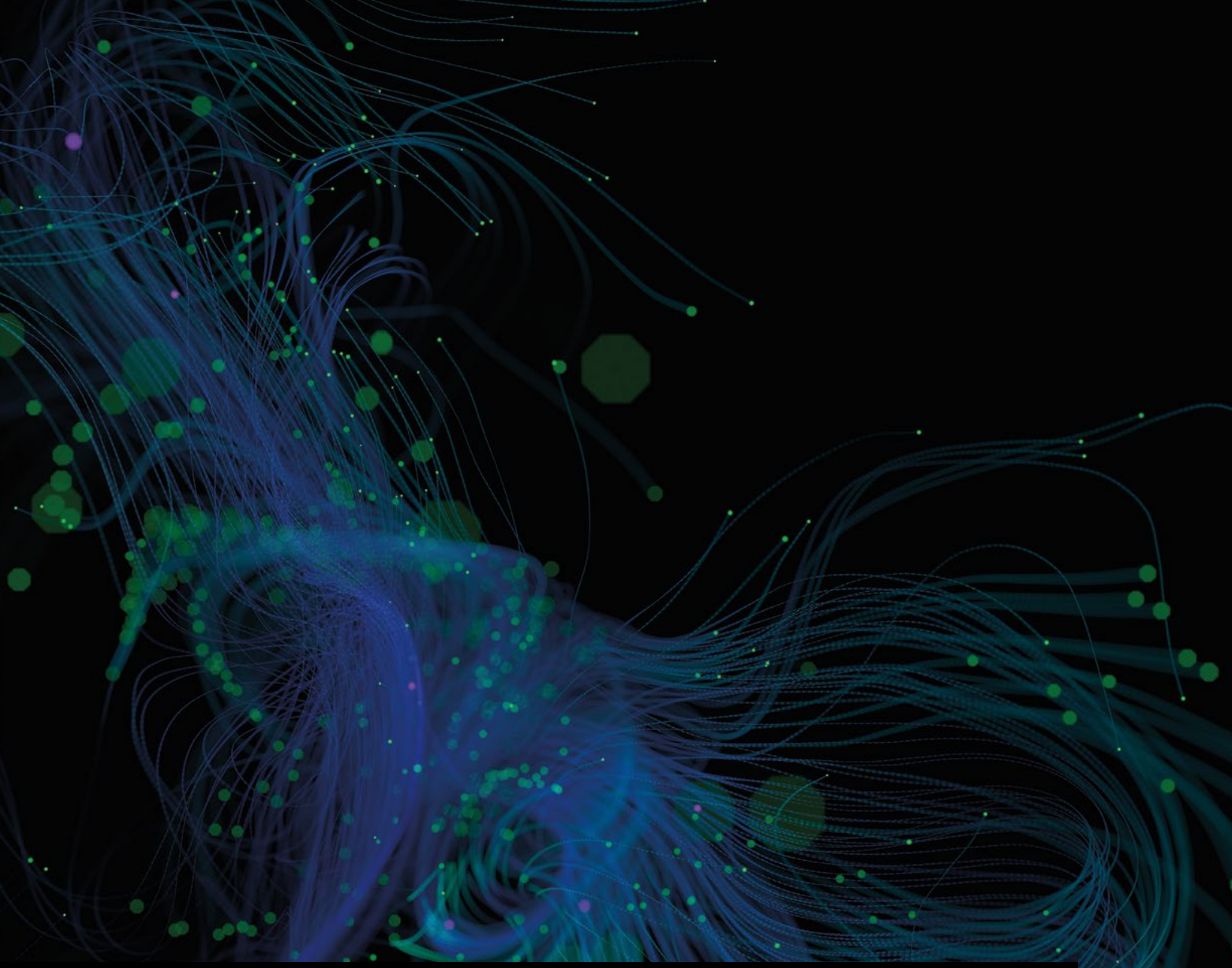


CylancePROTECT® Script Control

Feature Focus

CYLANCE™

Why Is CylancePROTECT Script Control Important?

Scripting has become a leading mechanism for malware distribution. The 2017 Verizon Data Breach Investigations Report identifies JavaScript as the leading propagator conduit (59%) for ransomware. The rationale for this is simple: malicious scripts are easily obtainable in the cybercrime underworld. Further, scripts are often difficult for some security products to detect, as scripts are commonly used by security administrators for non-nefarious purposes, so a script's conviction needs to be scrutinized by the intent of the user.

CylancePROTECT offers integrated script control to assist its superior artificial intelligence and machine learning based malware execution prevention technologies, giving administrative control over when, where, and how scripts are used in your environment. This ultimately reduces the attack surface on which an evildoer may distribute malware.

How Does CylancePROTECT Script Control Work?

CylancePROTECT Script Control protects users from malicious scripts running on their devices by injecting itself into a script interpreter (responsible for the execution of scripts) to monitor and protect against scripts running in your environment. The agent is then able to detect the script and script path before the script is executed.

How To Use CylancePROTECT Script Control

Depending on the policy set for CylancePROTECT Script Control (Alert or Block), the agent will allow or block the execution of the script.

Alert Mode

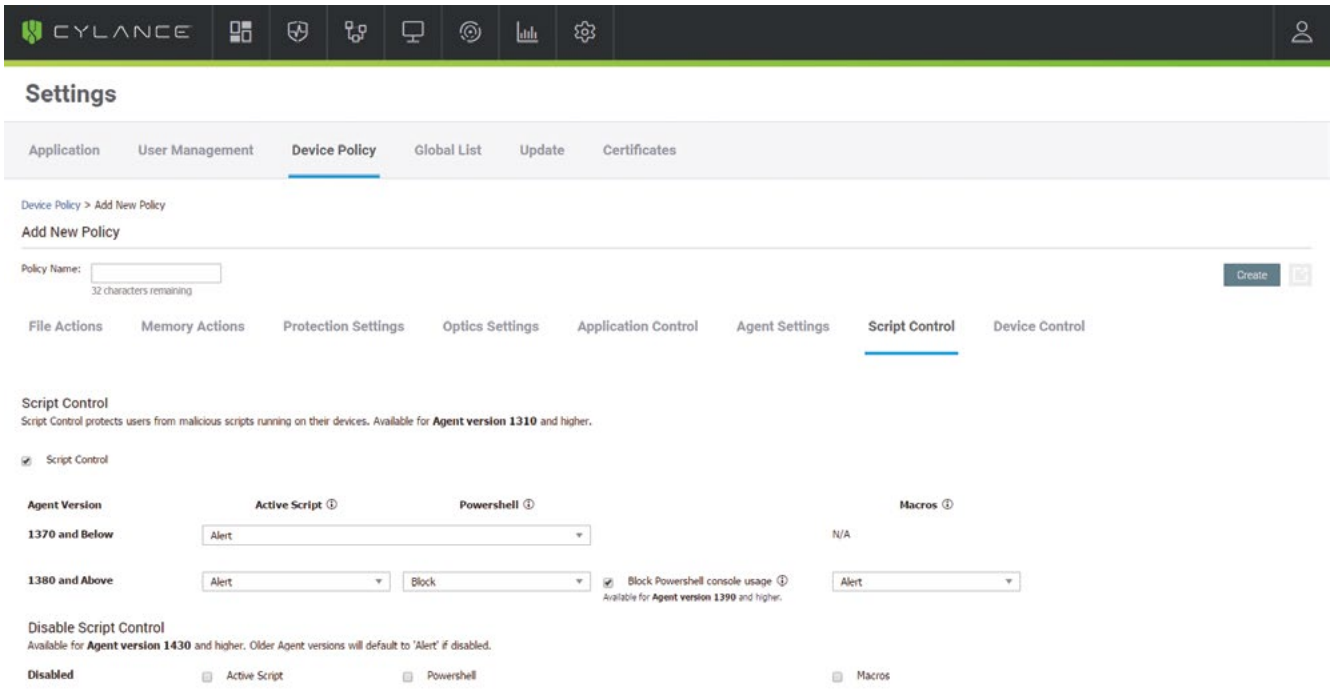
Allows all scripts to run, but alerts you when scripts are run.

It is recommended that administrators initially enable CylancePROTECT Script Control in Alert Mode to monitor and observe all scripts running in their environment.

Block Mode

Blocks all scripts. Approved scripts can be allowed to run using the Approve scripts in these folders (and subfolders) option (see information below).

Once administrators have a good understanding of all scripts running in their environment, they can change their settings to block mode and only allow scripts to run out of specified folders.



To enable Script Control from the Cylance Console, go to **Settings -> Device Policy -> Script Control** and turn on Script Control. Script Control can either be utilized in Alert Mode or Block Mode.

CylancePROTECT Script Control supports PowerShell and Active Scripts.

- PowerShell requires Agent version 1310 or higher
- Active Script requires Agent version 1340 or higher
- Microsoft Office Macros requires Agent version 1380 or higher

For more information on configuring CylancePROTECT Script Control, please see this knowledge base article.

FAQs

How does script control work?

Script control injects into a script interpreter (responsible for the execution of scripts) to monitor and protect against scripts running in your environment. By injecting into the interpreter, the agent is able to detect the script and script path before the script is executed. Depending on the policy set for script control (alert or block), the agent will allow or block the execution of the script.

What script types does CylancePROTECT Script Control detect?

CylancePROTECT Script Control detections vary per agent version:

- PowerShell - Agent 1310 and higher
- Active Scripts - Agent 1340 and higher
- Microsoft Office Macros - Agent 1380 and higher

What is Active Scripting?

With CylancePROTECT Script Control, the agent can detect two Active Scripting engines, VBScript and JScript, that run from the Windows Script Host (WSH). WSH is a language-independent scripting host and provides an environment for scripts to run by invoking the appropriate scripting engine. In this case, it is referring to the Active Scripting engines - VBScript and JScript. WSH can run in GUI mode (wscript.exe) or command-line mode (cscript.exe). See Microsoft's KB 188135 for more information regarding WSH.

Why are scripts running from PowerShell ISE not detected?

CylancePROTECT Script Control only detects PowerShell scripts from the PowerShell Interpreter, not the PowerShell ISE Interpreter.

Does CylancePROTECT Script Control protect against browser-based scripts?

No. CylancePROTECT Script Control only detects scripts that run natively on the device operating system.

What are the [*COMMAND*] events that I see in CylancePROTECT Script Control?

When PowerShell is set to **Block** and **Block PowerShell console usage** is enabled, any attempts to run the PowerShell console (or one-liner commands) will be blocked and logged. The exact commands, up to 250 characters, will be reported in the filepath/filename field.

If CylancePROTECT Script Control for PowerShell is set to Alert, do I have visibility into the PowerShell console usage?

No. Visibility into PowerShell console usage and the ability to block it requires that PowerShell be set to **Block,** and **Block PowerShell console usage** must also be enabled.

Does CylancePROTECT Script Control for PowerShell protect against one-liners?

Yes. When PowerShell is set to block, access to the PowerShell console is also blocked by default. Approved scripts can still be invoked by using the -F parameter in the Command Console (cmd). Otherwise, any attempts to use PowerShell commands (one-liners) will be blocked per policy.

Example: If c:\temp\approved\sample.ps1 is an approved script (as indicated in the exclusion folder, set in a policy), this script can be invoked by typing ***Powershell -F c:\temp\approved\sample.ps1*** in the Command Console (cmd.exe).

Is JScript the same as JavaScript?

No. JScript and JavaScript are different scripting engines, but have similar functionality. Both JScript and JavaScript scripts that are executed via CScript or WScript are detected by CylancePROTECT Script Control, and any actions are applied (Alert or Block). If these scripts are invoked via a web browser, CylancePROTECT Script Control will not detect or take any actions on these scripts.

About Microsoft Office Macros

Microsoft Office macros use Visual Basic for Applications (VBA) that allows embedding code inside an Office document (typically Word, Excel, and PowerPoint). The main purpose for macros is to simplify routine actions, like manipulating data in a spreadsheet or formatting text in a document. However, malware creators can use macros to run commands and attack the system. It is assumed that a Microsoft Office macro trying to manipulate the system is a malicious action. This is what CylancePROTECT Agents look for – malicious actions originating from a macro that affects things outside the Microsoft Office products.

Tip: Starting with Microsoft Office 2013, macros are disabled by default. Most of the time, you should not be required to enable macros to view the content of an Office document. You should only enable macros for documents you receive from users you trust, and when you have a good reason to enable them. Otherwise, macros should always be disabled.

+1-844-CYLANCE
sales@cylance.com
www.cylance.com
18201 Von Karman Avenue, Suite 700, Irvine, CA 92612

20170519-1096