EUROPEAN
COMMISSION

Brussels, 19.5.2020
C(2020) 3404 final

IT security standard

PowerShell

CONTENTS

1. INTRODUCTION

PowerShell is a task automation and configuration management framework, developed by Microsoft, which consists of a command-line shell and an associated scripting language. Administrators commonly use it to perform and automate the management of Windows systems, because it provides full access to the underlying operating system. For the same reason, however, attackers make extensive use of it to access systems and move laterally within an environment.

2. PURPOSE

This security standard establishes:

- conditions for authorising the use of PowerShell;
- minimum security requirements for the configuration of Windows operating systems to ensure resilience to attacks using PowerShell as an attack vector and to limit the extent of potential damage; and
- minimum requirements for logging PowerShell activities.

3. SCOPE

This standard applies to all applications and IT systems hosted on Commission premises. It sets out:

- requirements (subject to compliance verification); and
- recommendations (not subject to compliance verification), based on market best practice – these may be made mandatory in future versions of the standard, in the light of customer feedback, further best practice and compliance analysis.

4. EXCEPTIONS

The adoption of this standard is mandatory and an integral part of the Commission's information security policy. Exceptions must be handled in accordance with Article 2.2 of the "IT security standard on IT security compliance management" and the "implementing rules for Articles 3, 5, 7, 8, 9, 10, 11, 12, 14 and 15 of Decision 2017/46 on the security of communication and information systems in the Commission".

5. DEFINITIONS

For the purposes of this standard, in addition to the definitions in Article 2 of "COMMISSION DECISION (EU, Euratom) 2017/46 of 10 January 2017 on the security of communication and information systems in the European Commission" and Article 2 of the "implementing rules for Articles 3, 5, 7, 8, 9, 10, 11, 12, 14, 15 of Decision 2017/46 on the security of communication and information systems in the Commission", the following definitions also apply:

Term	Definition
application whitelisting	Allow only preapproved applications (listed in an index) to be executed on a system.
attack vector	A path or means by which an attacker can gain access to a computer or network server in order to deliver a payload or malicious outcome.
execution policy	A PowerShell setting that determines the conditions under which PowerShell loads configuration files and runs scripts.
language mode	Determines, in part, which elements of the PowerShell language can be used in a session.
Windows management framework (WMF)	A management interface in Windows systems for executing and automating administration tasks.

6. AUTHORISATION CONDITIONS

This section lists the requirements that have to be met for the use of PowerShell to be authorised in the Commission. These enable a secure configuration of PowerShell on all Windows systems and make security controls available to mitigate its use as an attack vector.

6.1. Use latest version of Windows management framework

The latest versions of the WMF (of which PowerShell is part) have relevant security features and should be installed on all Windows devices.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
1.	All Windows systems **shall** have the latest version of WMF installed.[1]	M

[1] For more information on the latest approved version of WMF, please contact the policy team.

6.2. Disable legacy versions

In the latest versions of Windows, legacy versions of PowerShell can be used to perform a downgrade attack bypassing PowerShell security controls.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
2.	All Windows systems **shall** have legacy versions of PowerShell disabled, where technically feasible.	M

6.3. Restrict legacy versions

Legacy versions of PowerShell can be easily enabled using administration privileges to bypass PowerShell security controls and carry out downgrade attacks.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
3.	Use of legacy versions of PowerShell **shall** be restricted by application whitelisting technology where technically feasible.	M

6.4. Limit PowerShell access

Standard users do not require access to PowerShell for daily tasks; they should be prevented from executing it.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
4.	Application whitelisting technology should be used to limit access to PowerShell to administrators only.	R

7. SECURITY CONTROLS

To prevent PowerShell being used as an attack vector, the built-in security controls in recent versions must be implemented as set out below.

7.1. Script execution policy

PowerShell's execution policy setting determines the conditions under which it loads configuration files and runs scripts, and whether scripts have to be digitally signed before they are run.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
5.	PowerShell execution policy **shall** be enforced to RemoteSigned.	M
6.	PowerShell execution policy **should** be enforced to AllSigned.	R

7.2. Constrained language mode

PowerShell's language modes determine, in part, which elements of the PowerShell language can be used in a session. Constrained language mode disallows access to non-core PowerShell features that are often abused in hacking tools / malicious scripts. Constrained language mode could be enforced using the environment variable PSLockdownPolicy, but this enforcing mechanism is not effective should be used for testing purposes only.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
7.	Constrained language mode **shall** be enforced using application whitelisting technology.	M
8.	PSLockdownPolicy **shall** not be used to enforce Constrained language mode	M

7.3. Just Enough Administration

Just Enough Administration (JEA) is a feature of PowerShell that limits administration privileges for remote sessions on managed endpoints on the basis of 'least privilege'.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
9.	Roles defined in JEA **should** be restricted to allow minimum required privileges.	R
10.	JEA **should** be implemented for users connecting remotely to Windows systems using PowerShell.	R

7.4. Windows Remote Management hardening

Windows Remote Management (WinRM) is a Windows built-in remote management protocol used in PowerShell remoting. To mitigate lateral movement using WinRM, the service must be hardened using Kerberos, the native Windows authentication protocol. PowerShell Direct is a feature that allows to run arbitrary PowerShell commands in a Windows virtual machine from its Hyper-V host regardless of network configuration or remote management settings and should therefore be disabled.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
11.	Kerberos **shall** be the only authentication protocol used for connecting remotely to Windows systems using PowerShell.	M
12.	Remote access to WinRM **shall** be limited.	M
13.	PowerShell Direct, where available, **should** be disabled	R

7.5. Antimalware Scan Interface

Antimalware Scan Interface (AMSI) is an anti-malware integration application program interface for analysing dynamic content of scripting engines. It prevents the execution of file-less malware by multiple scripting engines.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
14.	An anti-virus product that supports AMSI **must** be installed on all Windows systems.	M

8. LOGGING

To detect malicious activity, PowerShell's logging features must be activated and the data collected centrally for analysis.

8.1. Script block logging

PowerShell script block logging records detailed information from the processing of commands and scripts in the Windows event log. It deobfuscates obfuscated code used by attackers to bypass detection capabilities.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
15.	Script block logging **shall** be enabled on all Windows systems.	M
16.	Protected Event Logging **shall** be enabled on all Windows systems.	M

8.2. Module logging

Module logging records pipeline PowerShell execution details, including variable initialisation and command invocations, in the event log.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
17.	Module logging **shall** be enabled on all Windows systems for all PowerShell modules.	M

8.3. Transcript logging

Transcript logging records all input and output of every PowerShell session as it appears in the console. Transcripts are written to text files, with timestamps and metadata for each command to aid analysis.

(M = mandatory, R = recommended)

S/N	Requirement	Implementation
18.	Transcript logging **shall** be enabled on all Windows systems.	M
19.	PowerShell transcripts **shall** be collected at a secure location to prevent user tampering.	M

9. FINAL PROVISION

This standard shall enter into force on the day following the date of its adoption.

ACRONYMS

AMSI	Antimalware Scan Interface
JEA	Just Enough Administration
WDAC	Windows Defender Application Control
WinRM	Windows Remote Management
WMF	Windows management framework