STIG Implementation Report

- **Intern Credit Application For:** Bruce Thornton
  Date: 09/29/2025
  STIG Finding: STIG ID: WN11-CC-000326
- **SRG:** SRG-OS-000042-GPOS-00020
  Severity: medium
  Vulnerability ID: V-253414 **CCI:** CCI-000135

---

1. Introduction

This report documents the process of identifying, remediating, and verifying the fix for a Windows 11 STIG compliance finding. The selected finding was: STIG ID: WN11-CC-000326 “PowerShell script block logging must be enabled on Windows 11.”

---

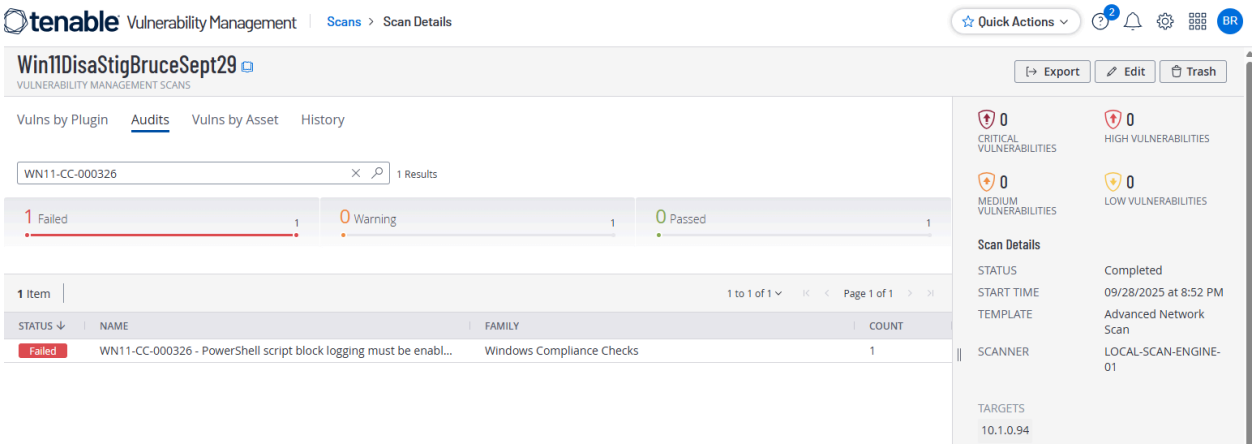
2. Initial Scan Results

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000326
- Status: **Fail** (non-compliant)

📎 **Evidence:** First identified the STIG:

https://stigaview.com/products/win11/v2r3/WN11-CC-000326/

Initial scan result:



---

3. Manual Remediation Steps

Run “gpedit.msc”

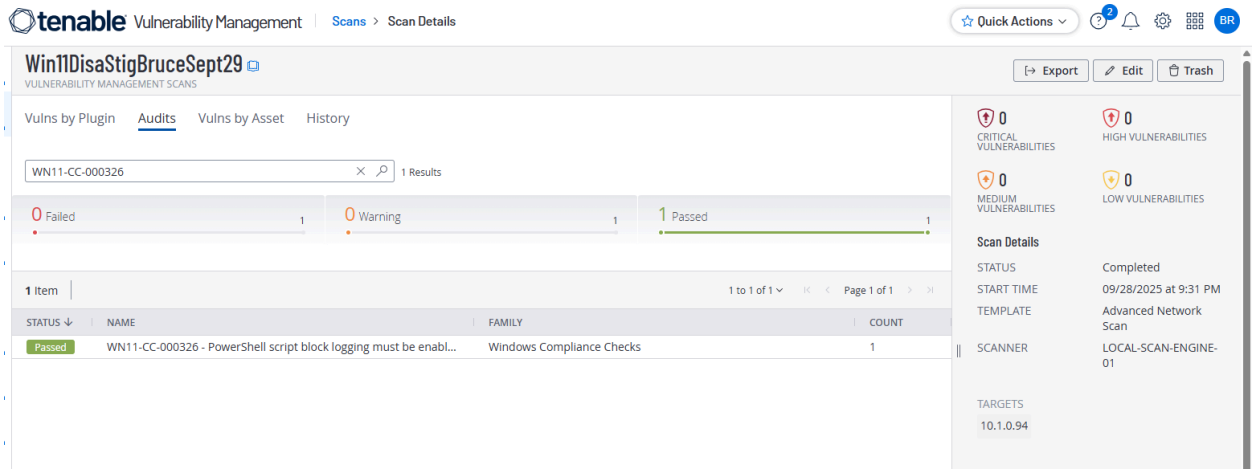
Configure the policy value for Computer Configuration >> Administrative Templates >> Windows Components >> Windows PowerShell >> "Turn on PowerShell Script Block Logging" to "Enabled".

Run “gpupdate /force” and restart.

Scan again,

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000326
- Status: Passed

Evidence:



---

4. Reintroduction of Finding (Manually Undo Test)

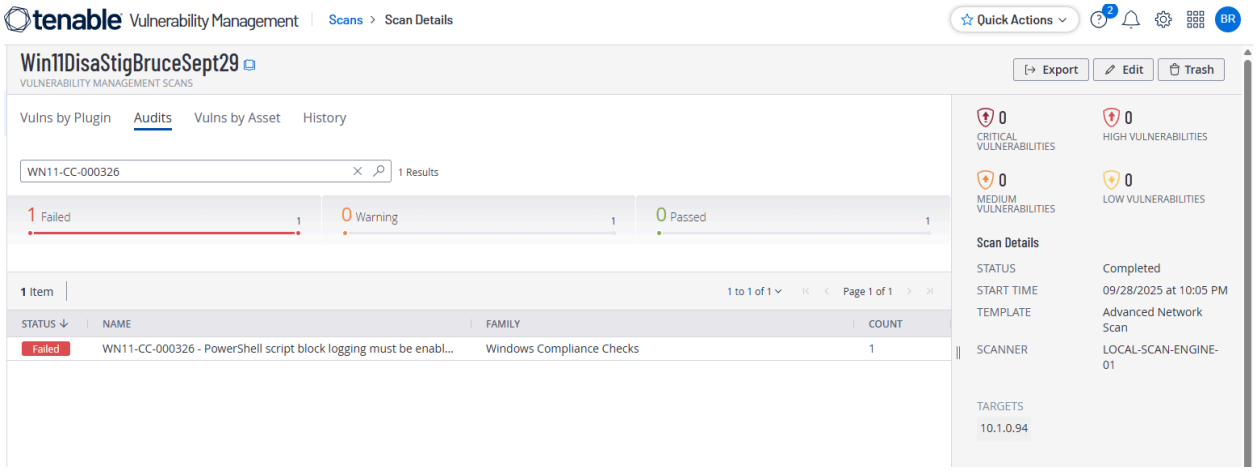
To demonstrate full control of the setting, the fix was undone:

- Disabled the setting. Open Group Policy Management “gpedit.msc” and followed the instructions for remediation from before and set it to the original setting: “Not Configured.”

- Ran “`gpupdate /force`” and rescanned.

Status: Failed, Non-Compliant

Evidence:



---

5. Remediation with PowerShell Script

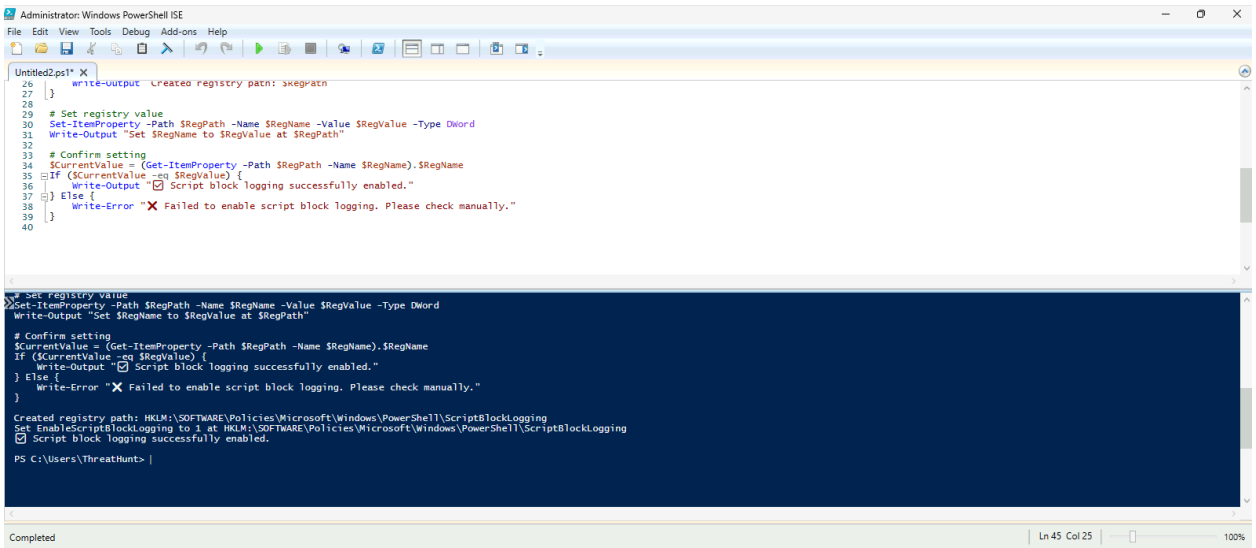
We opened PowerShell ISE within the Virtual Machine and ran this script:

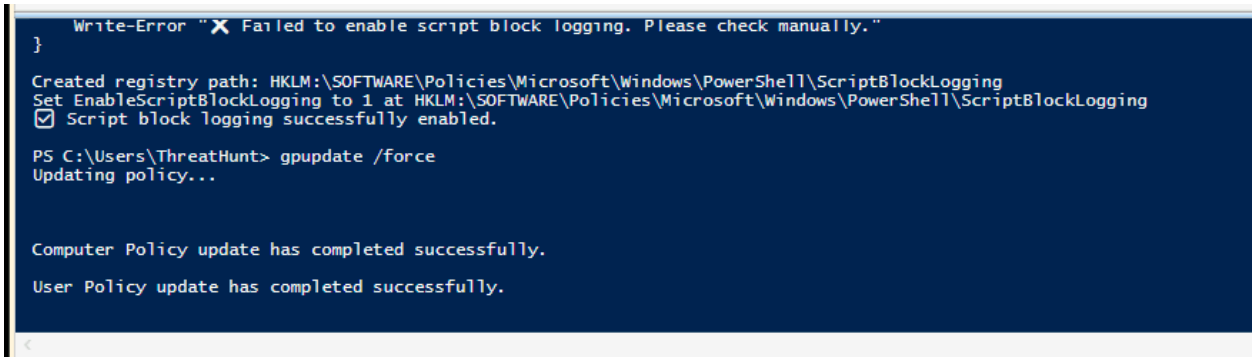
```
<#
.SYNOPSIS
  Remediates STIG ID WN11-CC-000326:
  Enables PowerShell Script Block Logging on Windows 11.

.DESCRIPTION
  This script creates or updates the registry key required
  to enforce script block logging. Must be run with Administrator privileges.
#>

# Ensure running as Administrator
If (-NOT ([Security.Principal.WindowsPrincipal]
[Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole(`
    [Security.Principal.WindowsBuiltinRole] "Administrator")) {
    Write-Error "You must run this script as Administrator."
    Exit 1
}

# Define registry path and value
$RegPath = "HKLM:\SOFTWARE\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging"
$RegName = "EnableScriptBlockLogging"
$RegValue = 1

# Create registry path if missing
If (-Not (Test-Path $RegPath)) {
    New-Item -Path $RegPath -Force | Out-Null
    Write-Output "Created registry path: $RegPath"
}

# Set registry value
Set-ItemProperty -Path $RegPath -Name $RegName -Value $RegValue -Type DWord
Write-Output "Set $RegName to $RegValue at $RegPath"

# Confirm setting
$CurrentValue = (Get-ItemProperty -Path $RegPath -Name $RegName).$RegName
If ($CurrentValue -eq $RegValue) {
    Write-Output "✅ Script block logging successfully enabled."
} Else {
    Write-Error "❌ Failed to enable script block logging. Please check manually."
}
```

How it works:

1. Checks if you're running as **Administrator**.
2. Creates the registry path if missing.
3. Sets `EnableScriptBlockLogging = 1`.
4. Confirms and outputs success/failure.

This can be saved as: `Remediate-WN11-CC-000326.ps1` and run it in **PowerShell ISE (Run as Admin)**.

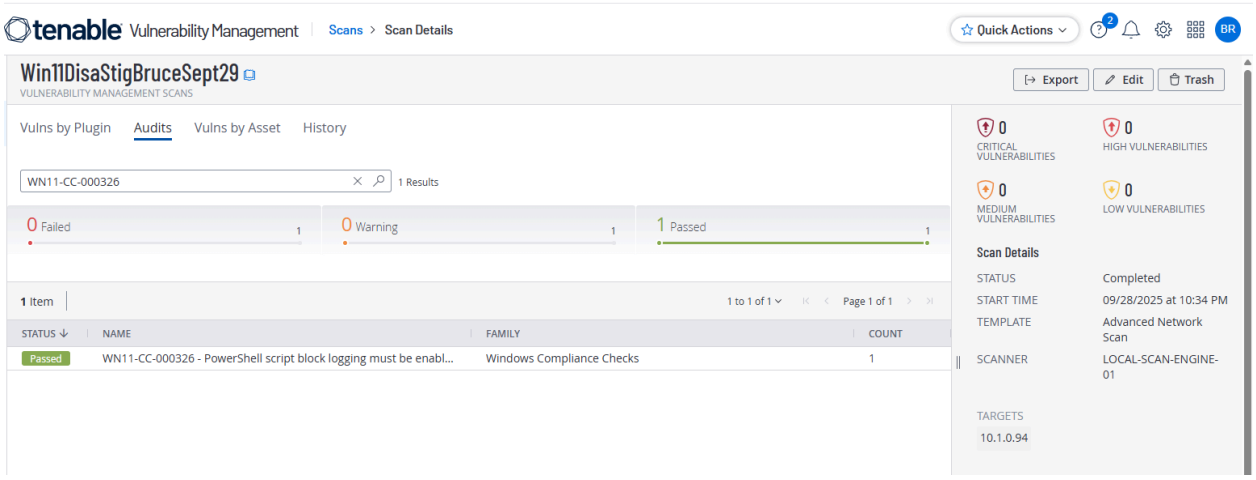




Ran gpupdate /force and then restart.

Status: Passed

Evidence:



---

6. Conclusion

The finding **WN11-CC-000326** was successfully:

- Detected in an initial Tenable STIG Audit scan,
- Remediated manually,
- Verified through a second scan,
- Undone and confirmed as vulnerable again,
- Finally re-applied through PowerShell automation, and validated with a third scan.

This demonstrates the ability to manage Windows STIG compliance both manually and through PowerShell automation.