STIG Implementation Report

- **Intern Credit Application For:** Bruce Thornton
  Date: 10/27/2025
  STIG Finding: WN11-CC-000120
- **SRG:** SRG-OS-000095-GPOS-00049
  Severity: medium
  Vulnerability ID: V-253378 **CCI:** CCI-000381

---

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-000120 “The network selection user interface (UI) must not be displayed on the logon screen.”

---

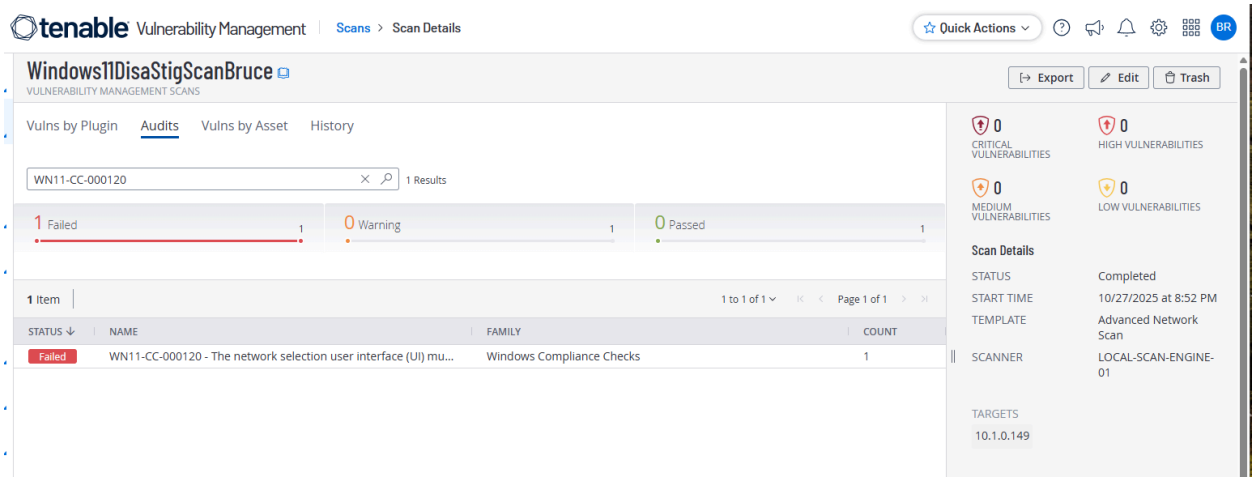
2. Initial Scan Results

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000120
- Status: **Failed** (non-compliant)

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

https://stigaview.com/products/win11/v2r3/WN11-CC-000120/

Initial scan result:



---

3. Manual Remediation Steps

Run “gpedit.msc”

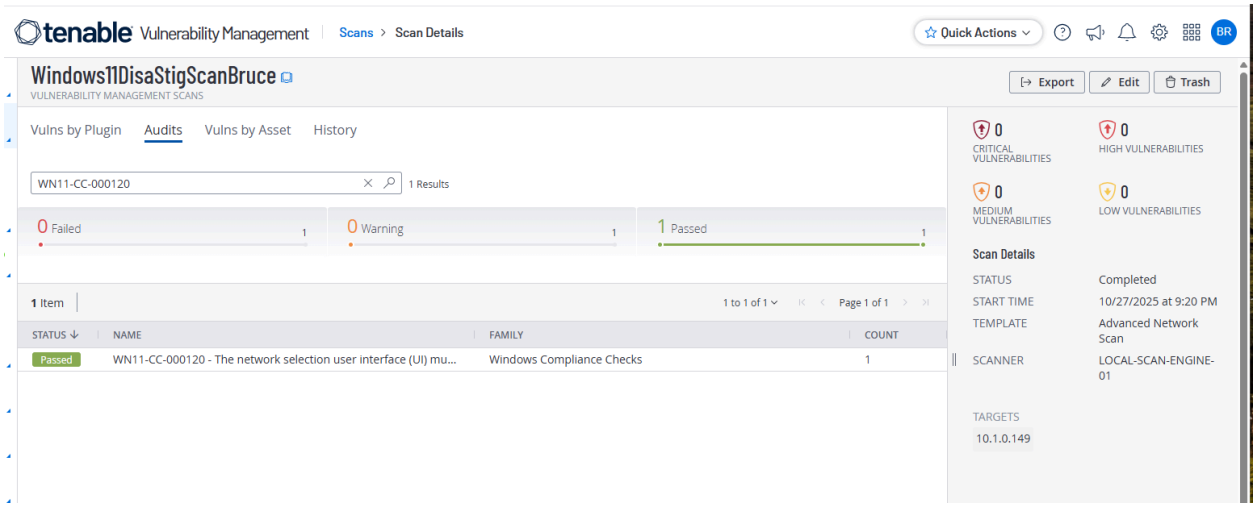
Configure the policy value for Computer Configuration >> Administrative Templates >> System >> Logon >> "Do not display network selection UI" to "Enabled".

Run “gpupdate /force” and restart.

Scan again,

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000120
- Status: **Passed**

Evidence:



---

4. Reintroduction of Finding (Manually Undo Test)

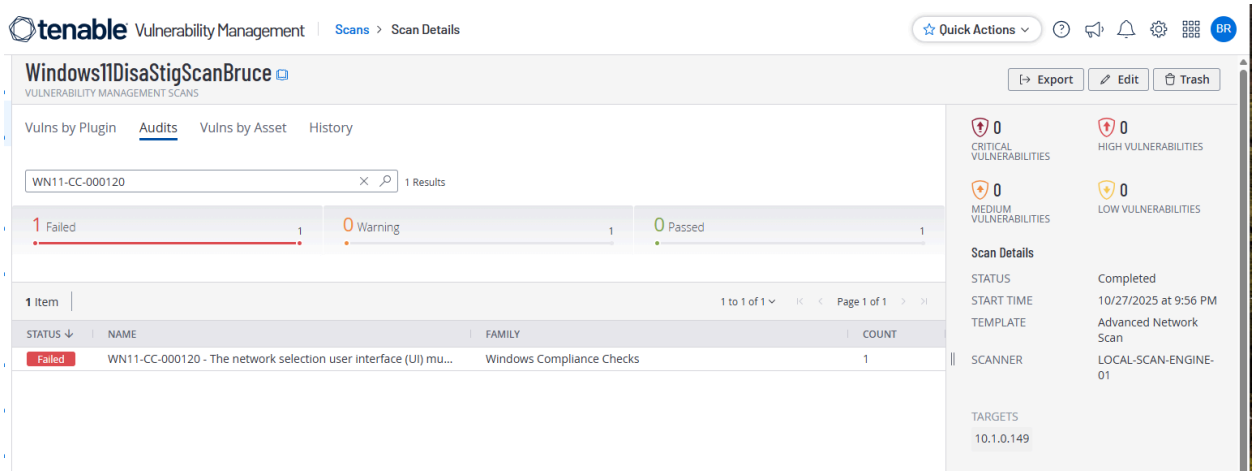
To demonstrate full control of the setting, the fix was undone:

- 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:



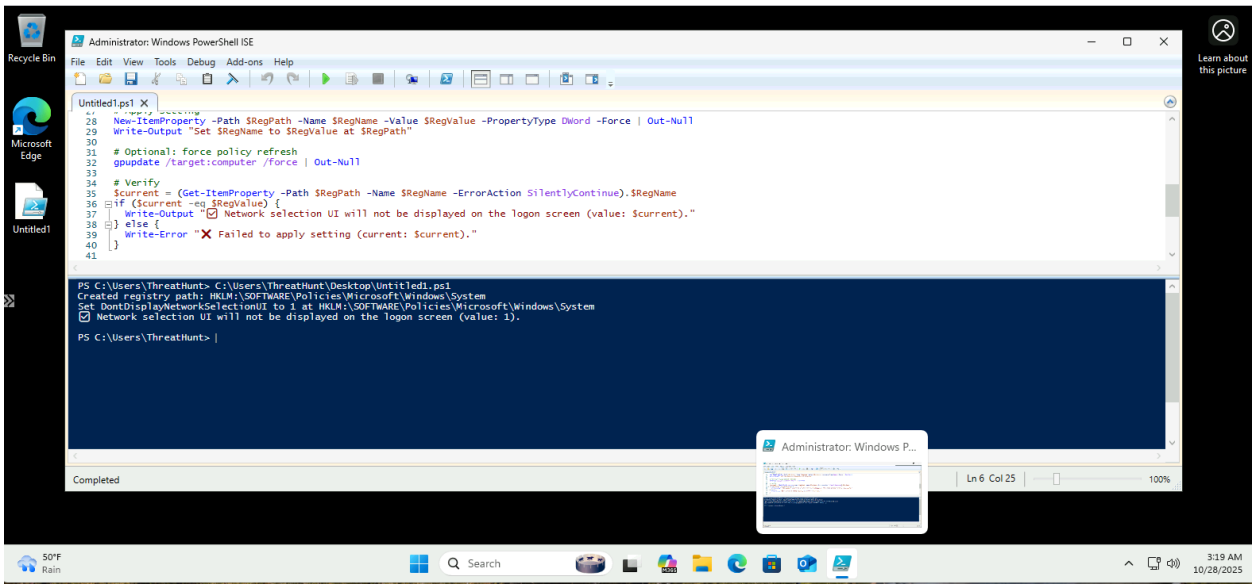
5. Remediation with PowerShell Script

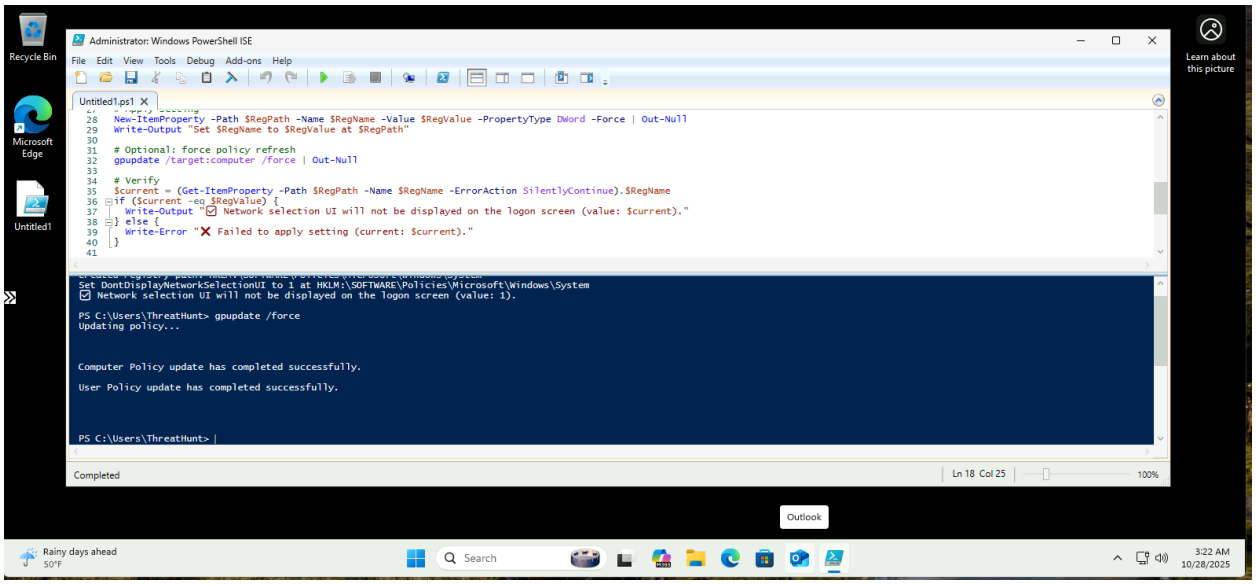
Save as: `Remediate-WN11-CC-000120.ps1` and run **as Administrator** utilizing PowerShell ISE:

```
<#
.SYNOPSIS
  Remediates STIG WN11-CC-000120:
  Hides the network selection UI on the Windows logon screen.

.NOTES
  Run as Administrator. Logoff/reboot may be needed for logon screen to reflect the change.
#>

# Require admin
if (-not ([Security.Principal.WindowsPrincipal] [Security.Principal.WindowsIdentity]::GetCurrent()
  ).IsInRole([Security.Principal.WindowsBuiltinRole] "Administrator")) {
  Write-Error "Run this script as Administrator."
  exit 1
}

$RegPath  = "HKLM:\SOFTWARE\Policies\Microsoft\Windows\System"
$RegName  = "DontDisplayNetworkSelectionUI"
$RegValue = 1

# Ensure path exists
if (-not (Test-Path $RegPath)) {
  New-Item -Path $RegPath -Force | Out-Null
  Write-Output "Created registry path: $RegPath"
}

# Apply setting
New-ItemProperty -Path $RegPath -Name $RegName -Value $RegValue -PropertyType DWord
-Force | Out-Null
Write-Output "Set $RegName to $RegValue at $RegPath"

# Optional: force policy refresh
gpupdate /target:computer /force | Out-Null

# Verify
```

```
$current = (Get-ItemProperty -Path $RegPath -Name $RegName -ErrorAction SilentlyContinue).$RegName
if ($current -eq $RegValue) {
  Write-Output "✅ Network selection UI will not be displayed on the logon screen (value: $current)."
} else {
  Write-Error "❌ Failed to apply setting (current: $current)."
}
```

—--



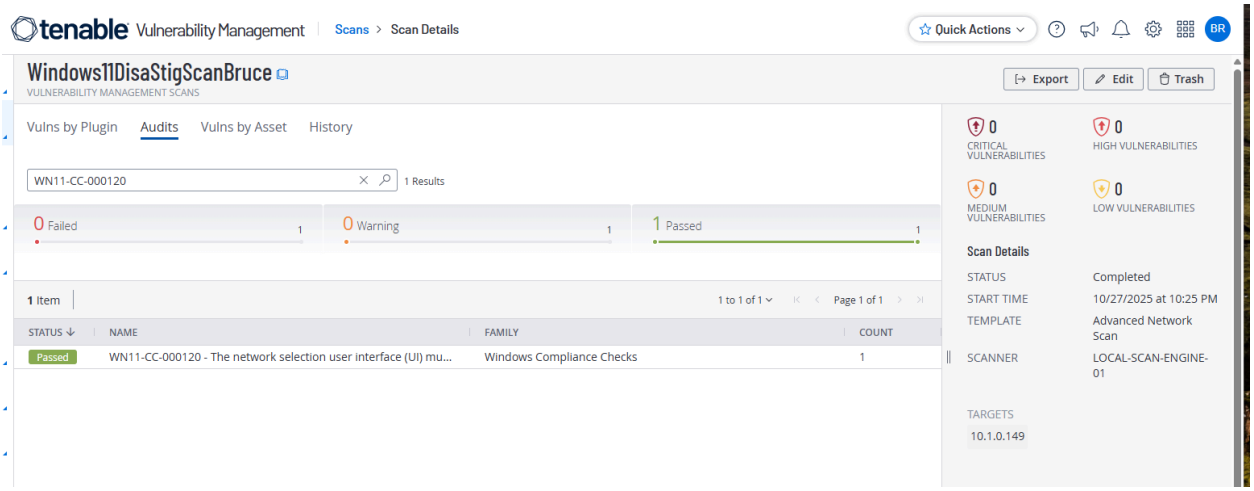


Run “gpupdate /force” and restart.

Scan again,

- Tool: Tenable.sc / Nessus (Windows 11 STIG Audit Policy)
- Finding ID: WN11-CC-000120
- Status: **Passed**

Evidence:



What it sets

- **GPO**: Computer Config → Admin Templates → System → Logon → **Do not display network selection UI**
- **Registry**: `HKLM\SOFTWARE\Policies\Microsoft\Windows\System` → `DontDisplayNetworkSelectionUI` (DWORD) = 1

Notes

- Tenable/Nessus typically checks that **`DontDisplayNetworkSelectionUI=1`** under `HKLM\SOFTWARE\Policies\Microsoft\Windows\System`.

- Change takes effect at the **next logon**; a **reboot or logoff** helps ensure the lock/Welcome screen reflects it.

- This is **computer-scope** (HKLM), so you don’t need per-user actions.

---

6. Conclusion

The finding **WN11-CC-000120** 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.